

**YILDIRIM BEYAZIT UNIVERSITY**  
**GRADUATE SCHOOL OF NATURAL AND APPLIED**  
**SCIENCES**



**STYLE-BASED GENERATIVE ADVERSARIAL**  
**NETWORKS FOR ENHANCING DEEP-LEARNING-**  
**BASED PERSON RE-IDENTIFICATION**

**Ph.D. Thesis by**

**Saleh Hussin Salem HUSSIN**

**Department of Computer Engineering**

**June, 2021**

**ANKARA**

**STYLE-BASED GENERATIVE ADVERSARIAL  
NETWORKS FOR ENHANCING DEEP-LEARNING-  
BASED PERSON RE-IDENTIFICATION**

**A Thesis Submitted to**

**The Graduate School of Natural and Applied Sciences of**

**Ankara Yıldırım Beyazıt University**

**In Partial Fulfilment of the Requirements for the Degree of Doctor of  
Philosophy in Computer Engineering, Department of Computer Engineering**

**by**

**Saleh Hussin Salem HUSSIN**

**June, 2021**

**ANKARA**

## Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**STYLE-BASED GENERATIVE ADVERSARIAL NETWORKS FOR ENHANCING DEEP-LEARNING-BASED PERSON RE-IDENTIFICATION**” completed by **SALEH HUSSIN SALEM HUSSIN** under the supervision of **PROF. Dr. Remzi YILDIRIM** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Ph.D.

Prof. Dr. Remzi YILDIRIM

---

Supervisor

Asst. Prof. Dr. Uraz YAVANOGLU

---

Jury Member

Asst. Prof. Dr. Gokhan GULTEKIN

---

Jury Member

Assoc. Prof. Dr. Gazi Erkan BOSTANCI

---

Jury Member

Asst. Prof. Dr. Yilmaz AR

---

Jury Member

Prof. Dr. Ergün ERASLAN

---

Director

Graduate School of Natural and Applied Sciences

## **ETHICAL DECLARATION**

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information, and documents are obtained in the framework of academic and ethical rules,
- All information, documents, and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of the above statements, I accept to renounce all my legal rights.

**Date:14 June, 2021**

**Signature**

**:.....**

**Name & Surname: Saleh Hussin Salem HUSSIN**



## ACKNOWLEDGMENT

First, I want to give praise and thanks to **ALLAH** Almighty for gifting me the strength and ability to complete this thesis.

It is also a great pleasure to acknowledge my deepest thanks and gratitude to my advisor, **Prof. Dr. Remzi YILDIRIM**, of the Graduate School of Natural and Applied Sciences, of Ankara Yıldırım Beyazıt University, for his continuous supervision, and sharing of his experiences, encouragement, and guidance throughout my study.

In addition, I wish to thank the **members of my thesis committee** for all of their support and valuable suggestions, and insightful comments, which greatly contributed to this work.

Moreover, I wish like to extend my very deepest thanks and gratitude to **my parents**, who have always given me their love and unconditional support.

Finally, my greatest thanks go to my beloved **family members** for their infinite support. I find no words to acknowledgment the sacrifice, help, and inspiration rendered by my **loving wife** and my **darling daughters** to take up this research. This thesis is dedicated to them. And my gratitude is extended to all of my **friends and colleagues** who provided me with all of the help that they could.

**June 2021**

**Saleh Hussin Salem HUSSIN**

# **STYLE-BASED GENERATIVE ADVERSARIAL NETWORKS FOR ENHANCING DEEP-LEARNING-BASED PERSON RE- IDENTIFICATION**

## **ABSTRACT**

Person re-identification (Re-ID) has been given a great deal of attention in recent decades. Because it is an essential task in intelligent surveillance systems and has possible widespread applications in many fields. When provided an image of a person captured from an alone camera, person Re-ID involves identifying the person from gallery images of people captured by several other cameras. This is a very challenging task, as the appearance of the person may appear very differently from camera to camera as a result of a variety of challenges, which can include occlusion, variations in illumination, changes in the poses of the individual, the viewpoint, and chaos that occurs in the background. Deep-learning (DL) technologies have greatly boosted the performance of person Re-ID, and this has contributed to reducing the effect of the challenges to some extent. However, they also add a new challenge, in that these deep methods need large amounts of labeled data for training. Hence, person Re-ID remains an open problem, for which no prominent solutions have been found for all of these different challenges. Thus, there is a need to develop a person Re-ID method that uses DL technology with a sufficient dataset for training. This study proposes an improved person Re-ID method based on DL technology, along with a style-based generative adversarial network (StyleGAN) and label-smoothing regularization for outliers (LSRO) algorithm. The proposed method can solve the main issue of DL-based person Re-ID, namely the lack of data needed for training. It begins by constructing a successful baseline model to extract the strong discriminative features necessary for person Re-ID by modifying a general convolutional neural network (CNN) model developed for the general object recognition task. Then, fine-tuning it using the transfer learning approach to make it more suitable for the person Re-ID problem domain. Moreover, random erasing and re-ranking are combined with the baseline model proposed herein to achieve significant performance improvement further and avoid overfitting. Afterward, the proposed method for person Re-ID exploits the StyleGAN to generate synthetic images that are both new and high-quality using the person Re-ID datasets that already exist. These newly generated images are then used to enlarge the training sets by introducing more extensive variations in terms of illumination, background, poses, and color. Then, the LSRO algorithm is used to integrate

the StyleGAN-generated images into the originally labeled training images. This is done by giving each of them uniform label distribution and designating a regularized loss function to them for the training of the baseline model. Generation of the new high-quality synthetic images using the StyleGAN and integrating them with the real images in datasets using the LSRO is the foremost contribution of the present research. The conducted experimental analysis and results using various person Re-ID datasets proved that the proposed person Re-ID approach yielded better overall performance when compared with state-of-the-art person Re-ID approaches

**Keywords:** Person re-identification; Deep learning; Transfer learning; Convolutional neural networks; Generative adversarial networks; Label smoothing regularization for outliers; StyleGAN.

# **DERİN ÖĞRENME TABANLI KİŞİ YENİDEN TANIMLAMAK İÇİN STİL TABANLI ÜRETİCİ ÇEKİŞMELİ (ADVERSARIAL) AĞLAR**

## **ÖZ**

Son yıllarda, kişinin yeniden tanımlanması, akıllı gözetim sistemleri ve birçok alanda yaygın olarak kullanılmakta, büyük ilgi görmekte ve uygulama olanaklarına da sahiptir. Kameradan çekilmiş bir kişinin görüntüsü verildiğinde, bu kişi çok sayıda kamera tarafından insan görüntülerinin bulunduğu veri tabanından taranarak bulunmaktadır. Kapatma, örtme, aydınlatma sorunu, poz değişiklikleri, bakış açısı ve arka plandaki karmaşıklıktan dolayı bir çok sorundan dolayı bir kişinin görünüşü farklı kameralarda büyük değişikliklere maruz kalabileceği için de seçmek de zordur. Derin öğrenme teknolojisi, kişiyi yeniden tanımlama performansını büyük ölçüde artırdı. Öğrenmedeki zorlukların etkisini bir dereceye kadar azaltmaya da katkıda bulundu. Ancak, bu derin öğrenme yöntemlerin eğitimi için büyük miktarda lisanslı veriye ihtiyaç duyulduğu için yeni bir zorluk ile de karşılaşıldı. Dolayısıyla, kişinin yeniden tanımlanması hala ciddi bir sorundur ve tüm değişik zorluklar için de belirgin bir çözümü de yoktur. Bu nedenle, eğitim setleri için yeterli veri seti ile derin öğrenme teknolojisini kullanan bir kişiyi yeniden tanımlama yönteminin geliştirilmesine ihtiyaç vardır.

Bu çalışmada, derin öğrenme teknolojisine dayalı kişi yeniden tanımlama yöntemi ile birlikte geliştirilmiş stil tabanlı üretken çekışmeli ağı (StyleGAN) ve aykırı değerler için etiket düzenlemesi (LSRO) algoritması önermektedir. Önerilen yöntem, derin öğrenmeye dayalı kişinin yeniden tanımlanmasına ilişkin temel sorunu, yani eğitim için gerekli veri eksikliğini çözmektedir. Nesne tanıma için geliştirilen genel evrişimli sinir ağı (CNN) modelini değiştirerek kişiyi yeniden tanımlamaya yönelik güçlü ayırt edici özellikler elde etmek için başarılı bir temel model oluşturularak işe başlar. Ardından, kişinin yeniden tanımlama problemi alanına daha uygun hale gelmek için transfer öğrenme yaklaşımını kullanarak ince ayar yapar. Ayrıca, rastgele silme ve yeniden sıralama, önerilen temel model ile birleştirilerek daha da ileri götürülerek, önemli performans iyileştirmesi elde edilir ve aşırı uydurma da önlenir. Daha sonra, mevcut kişi yeniden tanımlama veri kümelerinden yüksek kaliteli yeni sentetik görüntüler oluşturmak için StyleGAN'ı kullanır. Oluşturulan bu görüntüler, arka plan,

renk, aydınlatma ve pozlar açısından çok daha kapsamlı bir çeşitlilik sunarak eğitim setlerini genişletmek için kullanır. Sonrada, LSRO algoritması, StyleGAN tarafından üretilen görüntüleri, tek tip bir etiket dağılımı yaparak ve temel modeli eğitmek için düzenli bir kayıp işlevi tanımlayarak orijinal etiketli eğitim görüntülerine entegre etmek için kullanılır. StyleGAN kullanılarak yeni yüksek kaliteli sentetik görüntülerin oluşturulması ve bunların LSRO kullanılarak veri kümelerindeki gerçek görüntülerle entegre edilmesi, bu çalışmanın bu alana yaptığı en önemli katkısıdır. Farklı kişileri yeniden tanımlama veri kümeleri kullanılarak gerçekleştirilen deneysel analiz ve sonuçları, önerilen kişiyi yeniden tanımlama yaklaşımımız, diğer teknoloji metotlara göre daha iyi performans elde edilmiştir.

**Anahtar Kelimeler:** Kişinin yeniden tanımlanması; Derin öğrenme; Transfer öğrenimi; Evrişimli sinir ağları; Üretken düşmanlık ağları; Aykırı değerler için etiket yumuşatma düzenlemesi; StyleGAN.

## CONTENTS

<b>Ph.D. THESIS EXAMINATION RESULT FORM.....</b>	<b>ii</b>
<b>ETHICAL DECLARATION .....</b>	<b>iii</b>
<b>ACKNOWLEDGMENT .....</b>	<b>iv</b>
<b>ABSTRACT .....</b>	<b>v</b>
<b>ÖZ.....</b>	<b>vii</b>
<b>NOMENCLATURE.....</b>	<b>xii</b>
<b>LIST OF FIGURES .....</b>	<b>xv</b>
<b>LIST OF TABLES .....</b>	<b>xviii</b>
<b>CHAPTER 1 .....</b>	<b>1</b>
<b>INTRODUCTION.....</b>	<b>1</b>
1.1    Person Re-identification .....	3
1.1.1    Definition .....	3
1.1.2    Applications of Person Re-identification .....	5
1.1.3    Challenges of Person Re-identification.....	6
1.2    Motivation and Research Problem .....	9
1.3    Research Objectives and Contributions .....	11
1.4    Dissertation Organization.....	12
<b>CHAPTER 2.....</b>	<b>14</b>
<b>LITERATURE REVIEW .....</b>	<b>14</b>
2.1    An Overview of Person Re-Identification Current State .....	14
2.1.1    Traditional Hand-Crafted Person Re-Identification Methods.....	15
2.1.2    Deep-Learning-Based Person Re-Identification Methods .....	18
2.2    Deep Convolutional Neural Networks .....	18
2.2.1    Convolutional Neural Network Layers .....	19
2.2.2    Activation Functions .....	24
2.2.3    SoftMax Classifier and Loss Function.....	25
2.2.4    Training of Deep Convolutional Neural Networks.....	26
2.2.5    Handling an Overfitting in Deep Convolutional Neural Networks .....	29
2.2.6    Deep Convolutional Neural Networks Architectures .....	31
2.3    Transfer Learning for Developing Convolutional Neural Network Models.....	38
2.3.1    The Concept of Transfer Learning.....	39
2.3.2    Transfer Learning with Fine-Tuning the Pre-Trained Deep Models ...	40

2.4	Generative Adversarial Networks .....	42
2.4.1	The Derived Generative Adversarial Networks Architectures .....	45
2.5	Taxonomy of Deep Learning-Based Person Re-identification Methods ....	51
2.5.1	Methods Based on the Identification Model .....	51
2.5.2	Methods Based on the Verification Model .....	54
2.5.3	Methods Based on the Distance-Metric-Based Model .....	57
2.5.4	Methods Based on the Part-Based Model .....	59
2.5.5	Methods Based on the Generative Adversarial Network Model .....	61
2.6	Person Re-identification Benchmark Datasets and Evaluation Metrics.....	64
2.6.1	Person Re-identification Benchmark Datasets.....	64
2.6.2	Person Re-identification Evaluation Metrics .....	68
2.7	Concluding Summary.....	70
<b>CHAPTER 3.....</b>		<b>73</b>
<b>METHODOLOGY .....</b>		<b>73</b>
3.1	The Proposed StyleGAN-LSRO Method.....	74
3.2	Baseline Model Building for Person Re-Identification.....	75
3.2.1	Baseline Model Construction.....	76
3.2.2	Baseline Model Training.....	78
3.2.3	Features Extraction and Distance Metric Learning.....	79
3.2.4	Baseline Model Optimization .....	80
3.3	Style Generative Adversarial Network for Image Generation .....	85
3.3.1	StyleGAN Architecture.....	85
3.3.2	StyleGAN Training.....	91
3.4	Label Assignment Method for StyleGAN Generated Images.....	93
3.5	Experimental Study .....	97
3.5.1	Benchmark Datasets.....	97
3.5.2	Data Pre-processing .....	98
3.5.3	Performance Evaluation Metrics.....	101
3.5.4	Experimental Setup .....	104
<b>CHAPTER 4.....</b>		<b>107</b>
<b>EXPERIMENTAL RESULTS AND DISCUSSION.....</b>		<b>107</b>
4.1	Baseline Model Training and Evaluation.....	107
4.1.1	Baseline Model Training Results.....	108
4.1.2	Baseline Model Evaluation Results .....	110

4.2	StyleGAN Training and Evaluation .....	117
4.2.1	StyleGAN Training Results .....	118
4.2.2	StyleGAN Evaluation Results.....	121
4.3	Re-identification Evaluation using Expanded Datasets .....	125
4.3.1	Baseline Evaluation using Different Numbers of StyleGAN Images	125
4.3.2	Baseline Model Results Comparison and Analysis .....	128
4.3.3	Re-Identification Qualitative Analysis.....	129
4.3.4	Comparison with State-Of-The-Art Methods .....	130
<b>CHAPTER 5.....</b>		<b>134</b>
<b>CONCLUSION AND FUTURE WORK .....</b>		<b>134</b>
5.1	Conclusion.....	134
5.2	Future Work .....	137
<b>REFERENCES.....</b>		<b>138</b>
<b>APPENDICES .....</b>		<b>160</b>
<b>Appendix A – The Mathematical Foundation of GANs .....</b>		<b>161</b>
<b>Appendix B – Label Smoothing Regularization (LSR) .....</b>		<b>167</b>
<b>Appendix C – PyTorch Framework .....</b>		<b>170</b>
<b>Appendix D – TensorFlow Platform .....</b>		<b>171</b>
<b>CURRICULUM VITAE.....</b>		<b>172</b>



## NOMENCLATURE

### Acronyms

AdaIN	Adaptive Instance Normalization
Adam	Adaptive Moment Estimation
cGAN	Conditional Generative Adversarial Network
CMC	Cumulative Matching Characteristics Curve
CNN	Convolutional Neural Network
CTGAN	Multi-Camera Transfer Generative Adversarial Network
CycleGAN	Cycle Generative Adversarial Network
DCGAN	Deep Convolutional Generative Adversarial Network
DenseNet	Densely-Connected Convolutional Network
DG-GAN	Discriminative and Generative-Generative Adversarial Network
DPM	Deformable Part Model
FAN	Feature Aggregation Network
FC	Fully-Connected
FD-GAN	Feature Distilling Generative Adversarial Network
FFN	Fusion Feature Network
FID	Fréchet Inception Distance
FLOP	Floating-Point Operations Per Second
FPNN	Filter Pairing Neural Network
GAN	Generative Adversarial Network
GP	Gradient Penalty
HA-CNN	Harmonious Attention Convolutional Neural Network
HSV	Hue, Saturation, Value
IDE	ID Discriminative Embedding

ILSVRC	ImageNet Large Scale Visual Recognition Challenge
KISSME	Keep It Simple and Straightforward Metric
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LFDA	Local Fisher Discriminant Analysis
LOMO	Local Maximal Occurrence
LSGAN	Least Squares Generative Adversarial Network Loss
LSRO	Label Smoothing Regularization for Outliers
mAP	mean Average Precision
MSCDA	Mixed Selective Convolutional Descriptor Aggregation
OIM	Online Instance Matching
PAN	Pedestrian Alignment Network
PCA	Principal Component Analysis
PCB	Part-based Convolutional Baseline
PN- GAN	Pose-Normalization Generative Adversarial Network
ProGAN	Progressive Growing Generative Adversarial Network
PSD	Positive Semi Definite
PTGAN	Person Transfer Generative Adversarial Network
RCN	Recurrent Comparative Network
ReLU	Rectified Linear Unit
ResNet	Residual Convolutional Neural Network
RGB	Read, Green, Blue
RPP	Refined Part Pooling
RRI	Restraint and Relaxation Iteration
SGD	Stochastic Gradient Descent
SIFT	Scale Invariant Feature Transforms

SILTP	Scale Invariant Local Ternary Pattern
SLSR	Sparse Label Smoothing Regularization
SSIM	Structural Similarity
StyleGAN	Style Generative Adversarial Network
SVD	Singular Vector Decomposition
SVM	Support Vector Machine
WGAN	Wasserstein Generative Adversarial Network
XQDA	Cross-view Quadratic Discriminant Analysis

## LIST OF FIGURES

<b>Figure 1.1</b> Block diagram of a standard person Re-ID system. ....	4
<b>Figure 1.2</b> Some examples of person Re-ID challenges. ....	7
<b>Figure 2.1</b> The general CNN structure [61]. ....	19
<b>Figure 2.2</b> An illustration showing the operations of the Conv layer. ....	21
<b>Figure 2.3</b> Examples of max- and average pooling with a $2 \times 2$ filter and stride equal to 2. ....	22
<b>Figure 2.4</b> Simple network with 2 fully-connected layers. ....	23
<b>Figure 2.5</b> Activation functions that are mostly used in NNs. ....	24
<b>Figure 2.6</b> Schematic diagram of Dropout. ....	30
<b>Figure 2.7</b> Typical residual learning building blocks. ResNet-18/34 building block (left) and ResNet-50/101/152 building block (right). ....	32
<b>Figure 2.8</b> Architecture of ResNet-50. Down sampling by a stride of 2 is applied before each residual block. ReLU activation is utilized for all of the layers except SoftMax for the output layer [86]. ....	33
<b>Figure 2.9</b> Illustration of a typical residual block of ResNet-50. ....	33
<b>Figure 2.10</b> Five-layer dense block that has a growth rate of $k = 4$ . Each of the layers uses all of the preceding feature maps as inputs [83]. ....	34
<b>Figure 2.11</b> The architecture of DenseNet-121. Layers that are between 2 adjacent blocks are called transition layers, as they change the sizes of the feature-map as a result of convolution and pooling. ....	36
<b>Figure 2.12</b> GoogLeNet Inception module with dimensionality reductions [84]. ....	37
<b>Figure 2.13</b> Architecture of the Inception-v3 [84]. ....	38
<b>Figure 2.14</b> The difference between traditional machine learning and transfer learning approaches. ....	39
<b>Figure 2.15</b> A schematic diagram of the general workflow of transfer learning. ....	40
<b>Figure 2.16</b> The deep block structure of the CNN model for the fine-tuning procedure. ....	42
<b>Figure 2.17</b> General architecture of the original GAN. ....	43
<b>Figure 2.18</b> DCGAN generator architecture [103]. ....	47
<b>Figure 2.19</b> Overview of the ProGAN architecture. ....	50
<b>Figure 2.20</b> Progressive training of the ProGAN, from the low-resolution to high-resolution layers for synthetic faces generation. All existing layers remain trainable throughout the process. The 6 face images shown on the right are sample images generated using progressive growing at $1024 \times 1024$ pixels [110]. ....	50
<b>Figure 2.21</b> Basic architecture of the standard identification deep model [1]. ....	51
<b>Figure 2.22</b> Basic architecture of standard verification deep model [4]. ....	54

<b>Figure 2.23</b> Triplet deep model architecture. ....	57
<b>Figure 2.24</b> Structure of the PCB deep model [57].....	60
<b>Figure 2.25</b> Examples of the Market-1501 dataset images. Each of the columns represents one identity. ....	66
<b>Figure 2.26</b> Examples of the DukeMTMC-reID dataset images. Each of the columns represents one identity. ....	67
<b>Figure 2.27</b> Examples of the images from the MSMT17 dataset. Each row represents a single identity. ....	68
<b>Figure 2.28</b> CMC curve to Rank 1, 5, 10 conversions.....	69
<b>Figure 3.1</b> Overall block diagram of the StyleGAN-LSRO method.....	75
<b>Figure 3.2</b> The proposed baseline model architecture. ....	77
<b>Figure 3.3</b> Features extraction and distance metric learning.....	80
<b>Figure 3.4</b> RE results on the Market-1501 dataset.....	83
<b>Figure 3.5</b> RR procedure for person Re-ID.....	85
<b>Figure 3.6</b> StyleGAN generator architecture. ....	87
<b>Figure 3.7</b> AdaIN layer of the generator. ....	87
<b>Figure 3.8</b> Example of style mixing. The fine features in the images within the grid are those that were taken from the images at the top, whereas the coarse features are those that were taken from the images that are on the left.....	88
<b>Figure 3.9</b> StyleGAN discriminator architecture. ....	89
<b>Figure 3.10</b> Overall architecture of the StyleGAN model for person images generation.....	91
<b>Figure 3.11</b> Progress of the training process of the StyleGAN.....	93
<b>Figure 3.12</b> Label distributions of a real image and a StyleGAN generated image..	94
<b>Figure 3.13</b> Flowchart of training the baseline deep model using real and generated person images.....	96
<b>Figure 3.14</b> Examples of the normalized images from Market-1501. ....	99
<b>Figure 3.15</b> Examples of the horizontally random flipped images from Market-1501. ....	99
<b>Figure 3.16</b> Sample images from the Market1501 dataset after resizing to 256 x 256. ....	100
<b>Figure 3.17</b> Effect of truncation trick as a function of style scale $\psi$ . ....	106
<b>Figure 4.1</b> Training process of the baseline model on the Market-1501 dataset. ...	108
<b>Figure 4.2</b> Training process of the baseline model on the DukeMTMC-reID dataset. ....	109
<b>Figure 4.3</b> Training process of the baseline model on the MSMT17 dataset. ....	109
<b>Figure 4.4</b> CMC curves on the Market-1501 dataset. ....	110

<b>Figure 4.5</b> CMC curves on the DukeMTMC-reID dataset. ....	111
<b>Figure 4.6</b> CMC curves on the MSMT17 dataset. ....	112
<b>Figure 4.7</b> CMC curves on the Market-1501 dataset with RE and RR. ....	113
<b>Figure 4.8</b> CMC curves on the DukeMTMC-reID dataset with RE and RR. ....	114
<b>Figure 4.9</b> CMC curves on the MSMT17 dataset with RE and RR. ....	115
<b>Figure 4.10</b> Example of rank-10 results before and after applying RR. ....	116
<b>Figure 4.11</b> Samples of initial noise images that were generated using the StyleGAN generator during training on the Market-1501 dataset. ....	118
<b>Figure 4.12</b> Samples of images that were generated using the StyleGAN generator at a resolution of $8 \times 8$ during training on the Market-1501 dataset. ....	119
<b>Figure 4.13</b> Samples of images that were generated using the StyleGAN generator at a resolution of $16 \times 16$ during training on the Market-1501 dataset. ....	119
<b>Figure 4.14</b> Samples of images that were generated using the StyleGAN generator at a resolution of $32 \times 32$ during training on the Market-1501 dataset. ....	119
<b>Figure 4.15</b> Samples of images that were generated using the StyleGAN generator at a resolution of $64 \times 64$ during training on the Market-1501 dataset. ....	120
<b>Figure 4.16</b> Samples of images that were generated using the StyleGAN generator at a resolution of $128 \times 128$ during training on the Market-1501 dataset. ....	120
<b>Figure 4.17</b> Samples of images that were generated using the StyleGAN generator at a resolution of $256 \times 256$ during training on the Market-1501 dataset. ....	120
<b>Figure 4.18</b> Sample images generated by the StyleGAN generator trained on the Market-1501 dataset. ....	121
<b>Figure 4.19</b> Sample images generated by the StyleGAN generator trained on the DukeMTMC-reID dataset. ....	122
<b>Figure 4.20</b> Visual comparison of the images generated with the StyleGAN model with those generated with various state-of-the-art person generation methods using the Market-1501 dataset and the real ones, concentrating on the image backgrounds and the foregrounds. ....	123
<b>Figure 4.21</b> FID score that was taken while training the StyleGAN model using the Market-1501 dataset. ....	124
<b>Figure 4.22</b> Top 20 retrieved results for specific queries belonging to the Market 1501 dataset via the application of the proposed method. Queries are shown in the column furthest to the left, while the images retrieved from the gallery are given in order, from left to right, based on the similarity scores. The letters T (in green) and F (in red) at the top of each image indicate the true and false equivalents. ....	130
<b>Figure A.1</b> KL Divergence. ....	165
<b>Figure A.2</b> JS Divergence. ....	165
<b>Figure B.1</b> Effect of label smoothing on the accuracy of the classification deep models. ....	167

## LIST OF TABLES

<b>Table 2.1</b> Architecture of ResNet. Building blocks (in brackets) with the number of stacked blocks [82].	33
<b>Table 2.2</b> The architecture of DenseNets. Each of the Conv layers in the table corresponds to the sequence BN-ReLU-Conv [83].	35
<b>Table 2.3</b> Summary of commonly used datasets for person Re-ID.	66
<b>Table 3.1</b> Structure of the discriminator network.	90
<b>Table 3.2</b> Details of the selected person Re-ID datasets.	97
<b>Table 4.1</b> Validation accuracy.	108
<b>Table 4.2</b> Market-1501 dataset evaluation results.	111
<b>Table 4.3</b> DukeMTMC-reID dataset evaluation results.	111
<b>Table 4.4</b> MSMT17 dataset evaluation results.	112
<b>Table 4.5</b> Market-1501 dataset evaluation results with RE and RR.	113
<b>Table 4.6</b> DukeMTMC-reID dataset evaluation results with RE and RR.	113
<b>Table 4.7</b> MSMT17 dataset evaluation results with RE and RR.	114
<b>Table 4.8</b> Comparison of the baseline performance with the state-of-the-art DL approaches.	117
<b>Table 4.9</b> Comparison of the FID and SSIM for the Market-1501 dataset images, both real and the generated. The lowest FID score indicated better quality, whereas the highest SSIM meant that there was more variety within the generated images.	124
<b>Table 4.10</b> Performance improvement contributions (%) via adding the different numbers of StyleGAN images to the Market-1501 dataset under a single query mode.	126
<b>Table 4.11</b> Performance improvement contributions (%) via the addition of the different StyleGAN image numbers to the Market-1501 dataset under a multi-query mode.	126
<b>Table 4.12</b> Performance improvement contributions (%) via the addition of the different StyleGAN image numbers to the DukeMTMC-reID dataset under a single query mode.	126
<b>Table 4.13</b> Performance improvement contributions (%) via the addition of the different StyleGAN image numbers to the MSMT17 dataset under a single query mode.	127
<b>Table 4.14</b> Comparison of the baseline model that was proposed herein with the baseline models in [50] and [148] on the Market-1501 dataset. Rank-1 accuracy and mAP under single query mode are listed without RR.	129
<b>Table 4.15</b> Comparison of the baseline model that was proposed herein with the baseline models in [50] and [148] on the DukeMTMC-reID dataset. Rank-1 accuracy and mAP under single query mode are listed without RR.	129

<b>Table 4. 16</b> Comparison of the StyleGAN-LSRO method proposed herein with the existing state-of-the-art non-generative person Re-ID methods. The Rank-1 accuracy and mAP in single-query mode are given. ....	131
<b>Table 4.17</b> Comparison of the StyleGAN-LSRO method proposed herein with the existing state-of-the-art person Re-ID methods that use the data generated. Rank-1 accuracy and mAP in single-query mode are given.....	132
<b>Table A.1</b> GAN loss variants .....	166



# CHAPTER 1

## INTRODUCTION

Currently, intelligent video surveillance systems are active and demanding fields of research in computer science and computer engineering. In this area, computer vision as well as machine learning techniques are in great demand for the automation of monitoring and analyzing camera network videos to handle and understand camera network-acquired videos. This area of research incorporates a range of different tools to help surveillance operators and forensic investigators. These tools include online applications for detecting and monitoring persons, recognizing unusual actions/behavior from the camera network, and offline applications that are used to gather person-of-interest images from the video frames that were acquired from a range of cameras views [1]. Searching for a target person in videos captured by many non-overlapping cameras is an important yet, challenging problem in the fields of intelligent video surveillance. Hence, person re-identification (Re-ID) is a key technique in the person searching task.

Person Re-ID has been proposed as one of the tools of intelligent video surveillance systems. Person Re-ID comprises the recognition of an individual via a video surveillance camera network that possibly has fields of view that are non-overlapping [2, 3]. In a general sense, applications consisting of person Re-ID provide support for surveillance operators as well as forensic investigators when they need to retrieve videos that show a person of interest. In the person Re-ID, when an image of a person captured from a single camera is given, the task entails identifying this person from a gallery set that was captured by various other cameras. This is a very challenging task, as the appearance of the person may appear very differently from camera to camera as a result of a variety of challenges, which can include occlusion, variations in illumination, changes in the poses of the individual, the viewpoint, and chaos that occurs in the background [4].

The recent progress in person Re-ID has mainly been based on advances in the architecture of deep convolutional neural networks (CNNs) in building an efficient

algorithm for person Re-ID. Algorithms such as this often necessitate a great deal of labeled data to be used in training. Despite the fact that some of the large-scale datasets that are used for person Re-ID have been released, a problem lies in the fact that they remain insufficient for use in deep model training, in which the number of identities in each dataset, as well as the number of images of each person, remain limited. Technology for data augmentation, to some extent, can be used to overcome these limitations in the existing person Re-ID datasets [4]. Generative adversarial network (GAN) [5] models have been employed as data augmentation methods to produce additional images using the original dataset in order to solve overfitting problems and to address a number of the limitations in person Re-ID, and also to boost the learning ability of the deep models that used for the person Re-ID task. However, there is a problem that continues to limit their optimal use, which comprises the fact that the synthesized images generated are not of the best quality; hence, the noise is also present in the training dataset. Moreover, an effective method must be found for labeling these newly generated un-labeled person images. For future research, a promising trend for person Re-ID comprises using GAN models to create new, high-quality, and highly diverse images. The use of newly generated data such as this would allow the accuracy of the person Re-ID models to be improved in real scenes.

To overcome the aforementioned problems, this research proposes a new method for producing new and large-scale datasets, which will be a great deal larger and much more comprehensive than those that exist now, allowing deep models that are more powerful to be constructed for the person Re-ID task. To this end, rather than labeling the images taken from the cameras, which are placed at different locations in certain scenarios manually, a novel synthetic method for data generation is investigated in this research. A new data augmentation method is proposed for person Re-ID based on the style-based GAN (StyleGAN) [6]. The StyleGAN is used to generate high-quality person images using the person Re-ID dataset that already exists. These newly generated images are then used to enlarge the training sets via the introduction of more extensive variations with regards to the background, illumination, color, and poses so that the robustness and accuracy of the person Re-ID models can be regularized and improved. The label-smoothing regularization for outliers (LSRO) algorithm was adopted. The un-labeled images that are generated were assigned a uniform label

distribution to define the regularized loss function to be used in training. Also developed was a baseline model based on a CNN, which was used to learn the discriminative features to be used in recognizing the person's identity. The experiments conducted in this research showed that the method proposed could significantly improve the performance of the person Re-ID task.

## **1.1 Person Re-identification**

### **1.1.1 Definition**

Person Re-ID is seen as an important task within the field of computer vision, which has been defined as: When given an image that was taken of a person from a single surveillance camera, Re-ID is a process in which the identification of that same person can be performed from the gallery set that was captured by various other surveillance cameras, with possibly non-overlapping fields of views and at different time intervals [2]. It is a challenging problem since the appearance of pedestrians can change significantly between different cameras. The necessity for this task was mainly the result of the increasing demand in the public security sector and the need to have an extensive surveillance camera network in public places, including those at airports, universities, shopping malls, etc.

The person Re-ID task is different from the classic identification and detection tasks. The identification task consists of determining a person's identity in an image to determine who it is. The detection task consists of discriminating people from the background without knowing the identity to indicate whether it is a person. In comparison, Re-ID answers whether a given image belongs to the same person as a query image and tells when and where this person appeared concerning a given camera, using several cameras, potentially allowing for the estimation of the person's trajectory certain period of time.

Typically, a person Re-ID system comprises 3 phases: person detection, tracking, and retrieval. While phases 1 and 2 are independent computer vision tasks, the majority of the person Re-ID studies that have been conducted have focused on phase 3. Therefore,

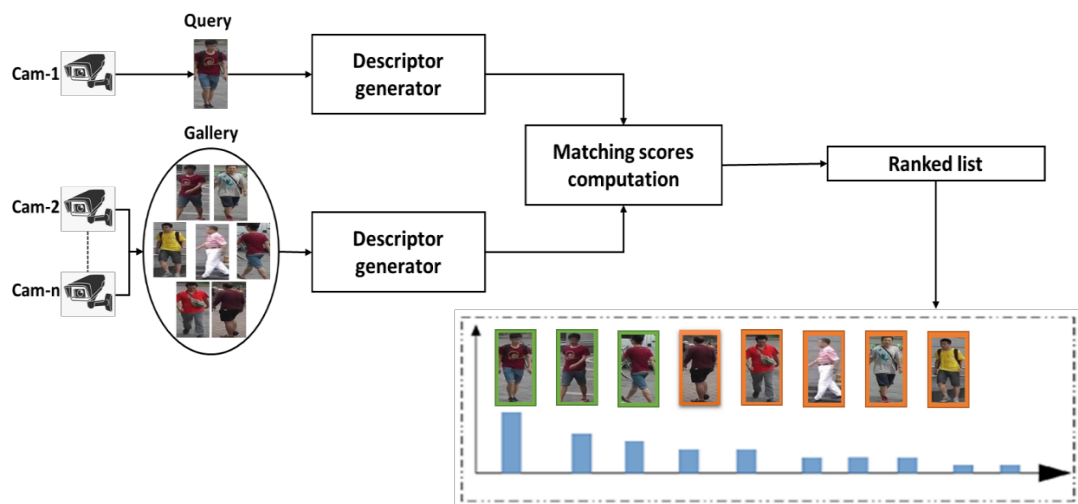
the current research conducted herein will focus on phase 3, which is person retrieval. The person retrieval process can be formulated as follows:

$$i^* = \arg \max_{i \in \{1, 2, \dots, N\}} M(Q, G_i) \quad (1.1)$$

Here  $i^*$  is the identity of query image  $Q$ , and  $M$  is a distance metric function to measure the similarity between query image  $Q$  and gallery image  $G$ . Hence, the person re-identification task generally consists of the following two main steps:

1. Constructing a descriptor or representation for the query image as well as each of the images in the gallery set that is capable of describing and discriminating the appearance of the person and is insensitive to illumination, occlusion, pose change, and other variances.
2. Performing the similarity matching between the images using an appropriate distance metric, and finally, showing the best-matched images according to the measured similarity score.

Figure 1.1 shows a depiction of the standard person Re-ID system. In step 1, a description of the image is generated for the query as well as each of the images in the gallery set. Then, the similarity, or the matching scores, between all of the gallery images and the query image is computed. Finally, after sorting all the matching scores in decreasing order, the ranked list of the best-matched images is then generated.



**Figure 1.1** Block diagram of a standard person Re-ID system.

Great efforts in research have concentrated on distance metric learning and feature extraction to facilitate an improvement in the performance of person Re-ID systems. However, in recent years, deep-learning (DL)-based approaches have become more and more popular in the person Re-ID tasks as a result of their success in image classification.

### 1.1.2 Applications of Person Re-identification

Person Re-ID is useful in various intelligent video surveillance applications. The most typical scenario where it is used is when multiple cameras are located around a specific location to ensure security, such as a shopping mall, airport, parking lot, university, or any other location. An image of a particular person is given (the query), and the search for that person is made in a set of images extracted from different cameras (the gallery). Thus, it is possible to track that person across these cameras. Person Re-ID methods can be applied in many practical applications, including:

- **Cross-multi-camera person tracking:** When a person moves out of the scene of one camera and into that of another, person Re-ID is used to determine the correlation between these disconnected tracks that are retrieved from the different cameras of the surveillance system. This allows for the reconstruction of the trajectory of a person and tracks their movement path from one point to another. For example, this helps to track lost children or elders and the escape routes of criminals to aid the police.
- **Person retrieval:** Here, Re-ID is associated with a recognition task. The specific query that comprises a target person is first provided, and then a search is conducted to find all of the related instances within a large database. Thus, the Re-ID task is used for image retrieval and typically provides ranked lists and related items, for example, person image retrieval used in forensics databases.
- **Human behavior identification and analysis:** The person Re-ID could help to identify and analyze different human behaviors. For example, to identify criminal or other suspicious behaviors for the consideration of public safety. Another example would include analyzing customer shopping trends by observing them,

touching, surveying, and trying products in stores under different surveillance cameras.

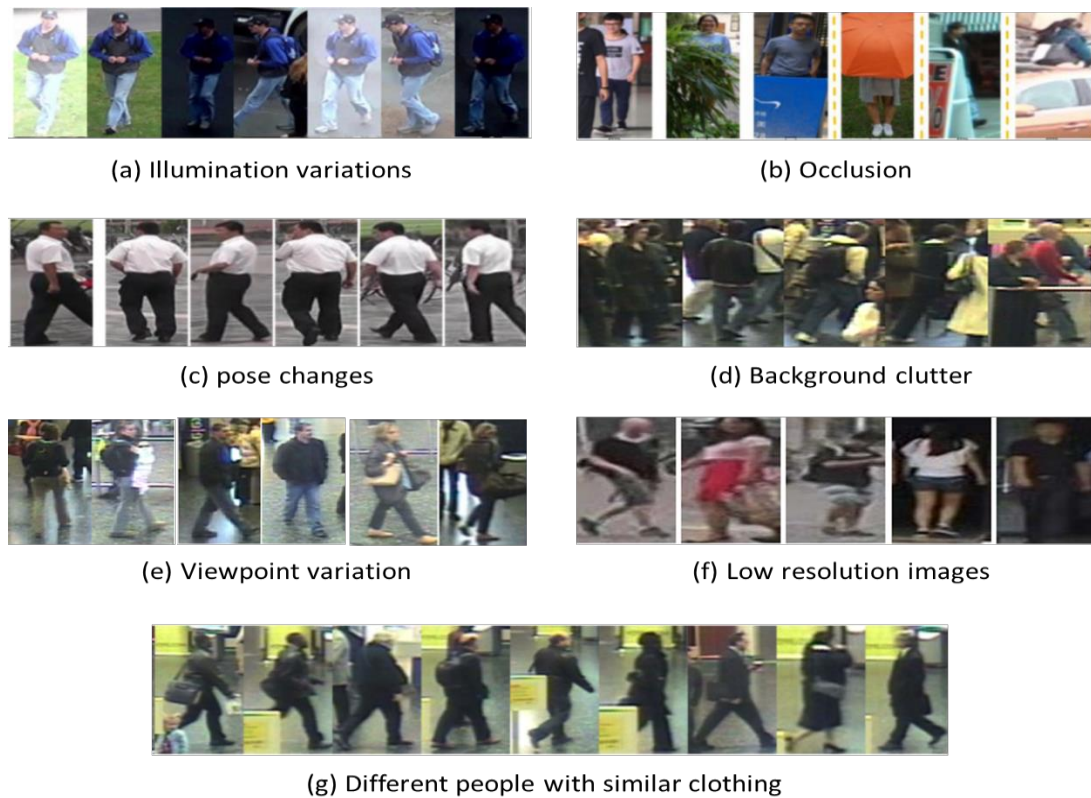
- **Human-machine interaction:** Recently, person Re-ID applications in human-machine interaction have been sharply increasing, which signifies the critical role of such a problem. In a scenario such as robotics, solving the Re-ID problem would be considered non-cooperative target recognition. The identity of the interlocutor is maintained, thus allowing the robot to have continuous awareness of the people surrounding it.
- **Sports analysis:** For large-scale sports, including basketball and soccer, the person Re-ID could automatically track and record the behaviors of the players to provide a better analysis.

### 1.1.3 Challenges of Person Re-identification

Similar to a situation that comprises several computer vision tasks, the person Re-ID task is significantly challenging and has several inherent difficulties since a person's appearance can change a great deal between different cameras. Therefore, the visual features of the image representation of a person can be various. Examples of some major challenges currently being addressed in the research community regarding the person Re-ID task are shown in Figure 1.2 and explained in the following.

- **Illumination variations:** Since illumination conditions can vary across multiple camera scenes for multiple reasons, including daylight intensity differences, changes in illumination color between indoor and outdoor environments, the effect of shadow, etc., changes may occur in the appearance and color of the person. Hence, pictures of the same person may sometimes be different, which increases the ambiguity and uncertainty in the matching process.
- **Occlusion:** Sometimes, in large and complex public areas, it is not easy to obtain an image of the complete body of a person without occlusion. In such places, it cannot be guaranteed that the cameras will face the pedestrians directly, as partial or complete overlaps will occlude people with other people or structures in the environment. If some important or discriminative parts of the person are not visible, the matching will likely be a failure.

- **Human body pose changes:** Pose variations imply that the body part localization of the person and visibility changes across multiple camera scenes. Therefore, due to the large impact of pose variations on a person's visual appearance, it is extremely challenging to predict and match the resulting images, which are most often of relatively low-resolution and low-quality. A learned model on a standing pose will probably fail to detect a running, crouching, or sitting person.



**Figure 1.2** Some examples of person Re-ID challenges.

- **Clothing or accessory variations:** The Re-ID algorithms rely mainly on the person's overall appearance, and the assumption that is generally made is that person wears the same clothing in the different scenes. However, in a realistic setting, there is a good chance that a person may be wearing different clothes or carrying different objects in the different camera scenes. For example, when a person takes their backpack off their back and carries it in their hand, or they take off their hat or coat, or put one on. Moreover, there is a high probability that different people have similar clothing. This leads to the similarity of the visual

appearance of different people, which increases the ambiguity and uncertainty in the matching process. In this case, it is more difficult to find the discriminative signature from the different visual appearances of the person.

- **Camera viewpoint variation:** The difference in the height of the cameras, the distance between the person and the cameras, and the direction in which the people face the cameras can cause the shape and gestures of the person to differ greatly under different viewing angles. Person images from different people from the same viewpoint might appear to have more similarity than 2 images taken of the same person from different views. The viewpoint variation is one of the most challenging problems, which increases, at the same time, the intra-class variation and the inter-class confusion.
- **Background clutter:** The use of different cameras results in unique background information. In an effort to eliminate this background information, person Re-ID methods usually operate on cropped person images that person detection algorithms have returned. However, the performance of existing person detection algorithms is not that accurate for the Re-ID purpose, i.e., detections include too much background or contain only part of the person. Therefore, human body regions are not well aligned across images, which seriously affects the performance of Re-ID in most of the existing methods.
- **Image resolution:** In practical wide area surveillance applications, cameras are usually installed in high places on walls, and pedestrians are usually far from the cameras. Even for high-resolution cameras, the image could still be of relatively low resolution for a given person. Low image resolution can also make a person's appearance different, which increases the ambiguity and uncertainty in the matching process.
- **Lack of labeled data:** Currently, most person Re-ID schemes are based on DL models. Training a good model robust to all variations in a supervised way could not be done without a sufficient amount of annotated data. Given that collecting an amount of data from the camera manually and annotating it is very expensive, usually, there is a limit to the number of labeled images of each person in public person Re-ID datasets. This is insufficient to train a good deep model. Therefore,



to achieves good generalization capabilities; complex algorithms must be taught using a huge amount of labeled data.

## **1.2 Motivation and Research Problem**

Recently, there has been an increasing demand for stable and reliable video surveillance systems in public places, such as shopping centers, airports, train stations, public transportation, sports centers, and so on, for monitoring the behavior, activities, or other changing information of people. It is also an efficient way to prevent crimes and to enhance the capabilities of forensic investigators. Most of the video surveillance systems that exist today are only able to capture, store, and distribute video, and as such, the task of threat detection is left exclusively to human operators. Hence, in a large-scale surveillance network, such human-based monitoring is very time-consuming and also, most often, provides limited accuracy. With the advent of powerful computing resources, automatic and intelligent video analysis has become possible, as well as more common in video surveillance applications, resulting in faster processing of the data, considerably enhanced accuracy, and a significantly improved ability to preempt incidents.

Person Re-ID has become a critical tool that is used in intelligent video surveillance systems. It consists of searching for a target person in videos that many non-overlapping cameras have captured by exploiting the full-body visual appearance of a pedestrian. Hence, it is both appealing and complementary when compared to the other Re-ID types, which are based on traditional biometric modalities, including fingerprint, palm-print, face, or gait recognition, which usually require precise measurements that low-resolution cameras cannot render in surveillance systems. It has many advantages, as personal body data is able to be captured from a significant distance, although with a low resolution. Additionally, the full body of a person is hard to emulate. In attempting to do so, it is most likely that the person will appear to be more suspicious than when other biometric techniques are used, like face recognition, because a person can easily hide their face. Moreover, in situations where face recognition is not possible, the person becomes very useful as a biometric parameter. Person Re-ID becomes a more challenging and difficult problem since the appearance

of a pedestrian can change significantly between different cameras. Although much research has been done in this field and improvements in the performance of person Re-ID have been gained using the DL methods, several problems are still unresolved/unsolved. Therefore, it is emerging as an exciting research field.

Person Re-ID has many problems and challenges. The relevant research problems can be listed as follows:

- As a result of developments in both computer- and GPU-accelerated systems, there is a greater possibility to use more sophisticated and powerful methods when solving person Re-ID problems. DL-based methods have become more and more popular because they have been successfully applied in image classification tasks, specifically CNNs. The success of DL-based solutions is correlated with the volume of data that is available for training. Several datasets are proposed for person Re-ID; however, none have been sufficiently large to train CNNs from scratch. These setbacks pose the need to find an alternative learning approach to train such deep models and produce better results in person Re-ID task.
- The training, evaluation, and performance characterization of deep models for the person Re-ID task means that it is necessary to create labeled data in so much that all the images collected of the same person will be bundled together. Creating a dataset with images that contains the diversity of situations that may arise in a practical scenario is an issue in itself. Still, they are of the utmost importance to correctly benchmark algorithms and drive state-of-the-art to higher levels. A dataset needs to be challenging to test the algorithms and highlight their flaws and bottlenecks correctly. Moreover, collecting large quantities of high-quality images and then manually labeling the appearances of a person is quite a tedious and time-consuming task. Additionally, concern regarding privacy issues is becoming an increasing issue. As a result of the General Data Protection Regulation [7], a strict process became necessary for collecting and handling the images of people. However, with the use of artificially-generated data, there are no issues with gathering, labeling, or privacy.

- Recently, some relatively large-scale person Re-ID datasets have been released; however, they still fall short for use in deep model training. The existing datasets have low diversities and small scales, where the number of identities included in each of the datasets and the number of images of each person remains limited. As an example, for large-scale Re-ID datasets, like Market-1501 or DukeMTMC-reID, the average number of images of each person is, respectively, 17.2 and 23.5, respectively [4]. If scale datasets like these are used to train the deep models, it will result in an overfitting problem and poor generalization performance. Therefore, there is a need to find some way to generate more images so that the volume of the datasets can be increased for use in the training process and to propose a new method to use in assigning them labels.
- Few researchers have recently proposed new methods based on unsupervised domain adaptation using GANs. GAN deep models have been employed as methods for data augmentation in the generation of person images, which provide the possibility to make person Re-ID datasets larger. However, the quality of the synthesized images that are generated is still quite poor because of the noise that is brought into the training dataset, which limits their use. Therefore, a promising trend for person Re-ID comprises using GAN models to create new, high-quality, and highly diverse images from the datasets that already exist. Using this generated data would thus address the dataset limitation problem and boost the generalization ability of person Re-ID deep models.

### **1.3 Research Objectives and Contributions**

The main objective of this dissertation was to investigate and propose a framework for person Re-ID utilizing DL-based solutions. The proposed framework will improve the performance and address several challenging issues that are perceived to adversely affect the person Re-ID task.

The essential contributions that were provided by this research were summarized, as are given below:

1. Proposing a successful baseline model to extract strong and discriminative features for use in the person Re-ID by modifying a basic CNN model that was developed for use in object recognition, and then fine-tune it using the transfer learning approach to become better suited for the person Re-ID problem domain. Moreover, combining random erasing (RE) and re-ranking (RR) with this newly proposed baseline model to achieve a significant performance improvement and avoid overfitting.
2. As far as is known, being the first to apply a StyleGAN model as a generative model in the generation of highly diverse and high-quality person images from the person Re-ID datasets that already exist. These newly-generated images can then be used to expand the training sets through the introduction of substantially more extensive variation with regards to the color, background, poses, and illumination.
3. Proposing the employment of the LSRO method in the integration of StyleGAN-generated images into original labeled training images by defining a regularized loss function for the training and assigning them a uniform label distribution.
4. Developing an evaluation protocol to measure the effectiveness of the proposed framework. This evaluation protocol includes qualitative as well as quantitative evaluations for as assessment of the suitability of the StyleGAN to generate new person images that are of appropriate diversity and quality. Moreover, it follows the commonly used standard evaluation protocol, which adopts the cumulative matching characteristics (CMC) curve, in addition to the mean average precision (mAP), as metrics that are used to evaluate the person Re-ID baseline model performance, as well as guarantee that there is an acceptable comparison between the approach proposed herein and the latest approaches in the literature.

## **1.4 Dissertation Organization**

The organization of this dissertation comprised five chapters, which are outlined, in brief, below:

Chapter 1: This introduction to the thesis presents the overview of the person Re-ID task, the research main motivations, the research problem statement, and the research objectives, as well as its contributions.

Chapter 2: The literature review presents the existing related works on person Re-ID and emphasizes DL techniques. The chapter also provides background on the different person Re-ID approaches, datasets, evaluation metrics, and other related works.

Chapter 3: Addressing the methodology, this chapter describes the proposed person Re-ID framework and technical details of the implementation. It also presents the evaluation protocol to evaluate different experimental analyses.

Chapter 4: Presenting the results and a discussion, this chapter discusses the results of the experiments that were conducted to evaluate this newly proposed method.

Chapter 5: The final chapter summarizes the presented work and concludes with the findings and contributions of the dissertation and possible future works.

# CHAPTER 2

## LITERATURE REVIEW

This chapter will present a comprehensive literature review of the relevant works that underpin the person Re-ID task. An overview of the DL techniques is given, particularly the CNN and the GAN, which are often applied for person Re-ID. An explanation of using the transfer learning approach on deep models to enhance the performance is also provided. Next, many of the state-of-the-art methods used for person Re-ID have been critically analyzed. A brief categorization of these person Re-ID methods is presented based on the type of DL technique used to develop them and evaluations of their pros and cons. Finally, some well-known benchmark datasets and evaluation methods are discussed at the end of this chapter.

### 2.1 An Overview of Person Re-Identification Current State

With the increasing popularity of surveillance systems, many research studies have emerged regarding the problems associated with person detection and tracking, and most recently, person Re-ID. Although person detection and multiple persons tracking within a single camera are challenging problems with their own difficulties, significant improvement has been achieved in recent years regarding their accuracy, efficiency, and robustness [8, 9]. Therefore, in this research, person detection and person tracking have been ignored. The primary work focuses on the person Re-ID process, which aims to automatically identify and match different images that have been taken of people taken from surveillance cameras that are separate and non-overlapping at different times. The person Re-ID process generally consists of 2 main stages: extracting robust feature descriptors, or representations, that can both describe and discriminate the persons and develop a similarity matching procedure between the extracted feature descriptors using an appropriate distance metric to tell which representations belong to the same person. As far as is known, the person Re-ID research began in 2005. The term ‘person re-identification’ was proposed with the work on multi-camera tracking by W. Zaidel, Z. Zivkovic, and B.J.A. Krose from the

University of Amsterdam. Their research on multi-camera tracking aimed to re-identify a person who left the field of view and then later re-entered [10]. A year after that, Gheissari et al. used just the visual cues of people after applying a spatial-temporal segmentation algorithm in foreground detection [11]. This work distinguished multi-camera tracking from person Re-ID, thus marking its beginning as an independent computer vision task [12]. Numerous person Re-ID methods have since been proposed in recent years, and their performance has been improved substantially. These methods were developed by exploiting computer vision techniques to address feature representation and similarity matching problems. DL approaches have recently been applied for person Re-ID, and state-of-the-art results have been achieved. According to some surveys and comprehensive reviews [4, 12-15] reported in the literature, the development, as well as the progress of person Re-ID methods, can be classified into 2 categories, comprising traditional hand-crafted person Re-ID methods and DL-based person Re-ID methods.

### 2.1.1 Traditional Hand-Crafted Person Re-Identification Methods

Traditional approaches proposed for person Re-ID were based on designing hand-crafted descriptors that relied on low-level features, such as color [16], texture [17], spatial structure [18], or any other descriptors combinations to describe the appearance of the person, and finding a reasonable distance metric function to compute the similarity scores between samples. Therefore, the traditional person Re-ID methods can be classified into 2 categories: feature-based and metric-based methods.

- **Feature-based methods:** Feature-based methods focus on designing hand-crafted features that can describe images and reliably distinguish between different people. These features should be discriminatively powerful and invariant to the common issues faced in the problems of person Re-ID, including, but not limited to, poses, illumination, viewpoints change, occlusion. Many different types of hand-crafted features have been proposed by using or combining low-level features, such as color [16, 19-22], texture [17, 21, 23] or interest point detectors information [17, 21]. These low-level features can be extracted globally from the whole person image or locally from the body parts by dividing the bounding box and extract the features separately. Hand-crafted features that are commonly used for person Re-

ID include color histograms [17, 19, 21, 24], local binary patterns [24, 25], color names [20, 22], scale-invariant feature transforms [17, 21, 26], and scale-invariant local ternary patterns [19, 27]. A combination of 3 different color histograms, within the red, green, blue (RGB); hue, saturation, value (HSV); and luminance, chrominance (YCbCr) color spaces, as well as 2 texture features, using the Gabor [28], and Schmid [29] texture filters, have also been used for building feature descriptors [30]. This feature representation combines lots of information. Moreover, histogram features, known as the local maximal occurrence features, can achieve some extent of invariance to viewpoint changes and demonstrate quite impressive performance on many benchmark datasets [19, 27]. Recently, a new trend has emerged, that instead of using low-level features, some handcrafted feature learning works use mid-level features, such as prominent human body areas. The salient regions, or discriminative areas, cause a person to stand out from those around them. These salient regions can provide valuable information that can boost the performance of person Re-ID models. A forward-looking approach was developed and presented in this direction by Zhao et al. [21]. This research mainly focuses on the DL-based model, which generally uses images as input but does not extract their low-level features. It has been proven that deep features are more discriminative and robust than low-level features. Hence, they will not be extensively explored in the research conducted herein.

- **Metric-based methods:** Metric-based methods aim to learn an effective distance metric that is able to measure the similarity or relation between two samples. Finding a good metric has a significant role in the success rate of the classification process of the hand-crafted features. More difficulty exists in learning a suitable distance metric than finding a discriminative descriptor [12]. In the person Re-ID, distance metric learning focuses on learning a distance metric that will minimize the distance between the images of the same person and maximize the distance between the images of different people. With a metric such as this, the matching and identification will be a lot easier to accomplish, and much improvement can be obtained in the performance. Many prominent metric learning algorithms [31-35] have been proposed to find a distance metric function to address the person Re-ID problems, the most popular of which is the KISSME (keep it simple and



straightforward metric) [31]. The KISSME, which is a method of Mahalanobis metric learning, attempts to learn a metric from equivalence constraints based on the perspective of statistical inference. Generally, a Mahalanobis distance metric provides the squared distance that exists between 2 images. The KISSME is able to handle large-scale data and does not require complicated optimization problems that necessitate expensive computations. Local Fisher discriminant analysis was proposed for person Re-ID in an effort to minimize within-class variances while maximizing inter-class separability [34]. To deal with the non-linearities in the feature space, it was proposed to use kernel-based dimensionality reduction techniques [35]. Support vector machine (SVM) learning was also proposed for learning the decision boundaries that can adapt to the data samples [33]. Some of the research that has been conducted recently has placed interest in learning a discriminative subspace for use in the matching procedure instead of learning a certain distance metric. Simply put, this means that a projection will be applied from the features of the original feature space to a low-dimensional subspace. In reality, principal component analysis (PCA) has been extensively applied for dimension reduction of the feature descriptors of images, followed by metric learning in the PCA subspace. Regarding person Re-ID as a ranking problem, RankSVM is applied to learn the subspace, which is actually a variation of SVM. In SVM, a set of RankSVMs that are weak are learned and then combined into a much stronger rank through ensemble learning, which makes RankSVM tractable for large-scale problems [36]. Another method that was proposed is known as cross-view quadratic discriminant analysis (XQDA). It is used for learning a discriminant low-dimensional subspace, and at the same time, in the derived subspace, a QDA metric is learned. This method is an extension of the KISSME approach. XQDA is suitable for dealing with the person Re-ID task because most Re-ID situations are directly under non-overlapping cameras; hence, the data comes from different views [19]. Positive semi definite (PSD) logistic metric learning is another well-known metric learning algorithm. It uses an efficient asymmetric sample weighting strategy [27]. Although these metric learning methods have achieved real competitive results with the hand-extracted features, they become less significant when the DL-based person Re-ID plays an essential

role. Because the recent DL person Re-ID approaches can generate more robust feature representations, the direct application of the preliminary distance metrics, such as Euclidean distance or Cosine distance, on the deep features can achieve excellent performance. Thus, these metric learning methods will not be widely explored in this research.

### **2.1.2 Deep-Learning-Based Person Re-Identification Methods**

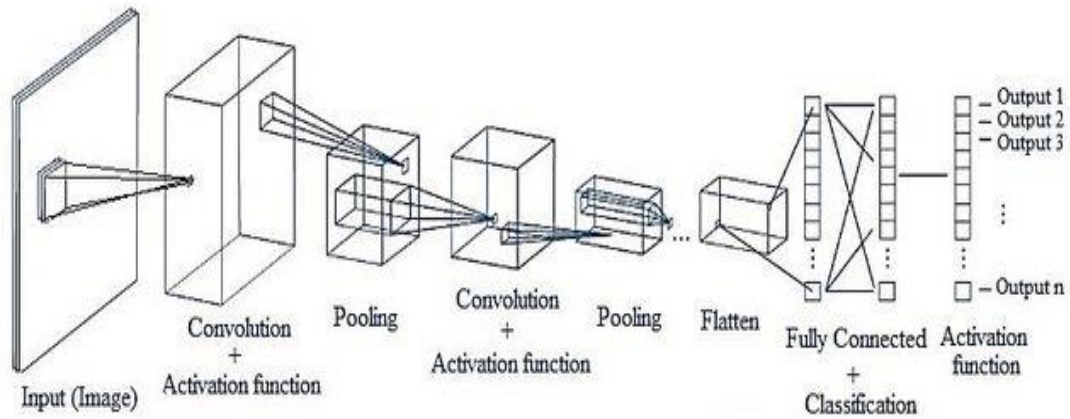
In addition to hand-crafted-based person Re-ID methods, DL-based methods have also become widely used in the person Re-ID task, especially the CNNs. This results from the discriminative and generalization power that these deep models have, which can provide much better performance and achievements in person Re-ID when compared to traditional hand-crafted feature methods. Different from traditional methods, the DL method can adaptively learn the deep features of people in images and learn a similarity metric with good effects. Therefore, most recent works have shifted from hand-crafted features to utilizing DL features [37-49]. The DL methods in person Re-ID still suffer from a lack of training datasets. Due to the gradual maturity of GANs in recent years, some person Re-ID methods that are based on GANs have also gradually arisen, and the GANs have generated satisfactory results in the expansion of person Re-ID datasets [50-56]. At present, most person Re-ID methods still belong to the category of supervised learning. Still, researchers have also carried out extensive research on transfer learning, semi-supervised, and unsupervised learning method [57]. This research mainly concentrates on the DL-based person Re-ID approaches. Therefore, in the following sections, an overview of the DL techniques used to address the person Re-ID problem, such as the CNNs and GANs, is given. A brief overview of the use of the transfer learning approach with deep models to gain good results is also given. Moreover, the taxonomy of some recent works that concern DL techniques for the person Re-ID tasks will be critically analyzed, with evaluations of their pros and cons.

## **2.2 Deep Convolutional Neural Networks**

DL, one of the branches of machine learning, aims to create higher-level representations of data while using multiple-layer non-linear operations [58, 59].

These high-level representations are a great deal more useful for recognition and classification than raw or basic features. Nowadays, CNNs have become the most popular DL architecture. They have shown to be able to outperform state-of-the-art methods in many tasks in fields such as natural language processing, computer vision, and robotics, just to name a few. Therefore, they have become widely used in recent years for learning optimal deep features for Re-ID [38, 39, 41, 44, 45]. In this section, the main concepts of CNNs are introduced.

A CNN [60] is a feed-forward NN that has multiple layers. It is composed of one or more Conv layers, which often include a pooling step. This is followed by one or more fully-connected (FC) layers, such as in a basic multi-layered NN. The architecture of the CNN has been designed to make use of the 2-dimensional structure of an image given as input. It constructs hierarchical connected translation-invariant features that are able to learn directly from the training dataset. Moreover, CNN models can be trained easier, and they have considerably fewer parameters compared to FC networks with the same number of hidden layers and neurons. Basic CNN modules comprise 5 layers: the input layer, the Conv layers, the pooling layers, the FC layer, and the output, or classification, layer. The general deep CNN structure is shown in Figure 2.1.



**Figure 2.1** The general CNN structure [61].

### 2.2.1 Convolutional Neural Network Layers

As shown in Figure 2.1, the CNN is composed of a certain set of layers customized based on the problem to be solved. In this section, a summary of the most important

layers will be given, and in addition, the role that they play in CNNs, as well as their advantages and disadvantages, will be discussed.

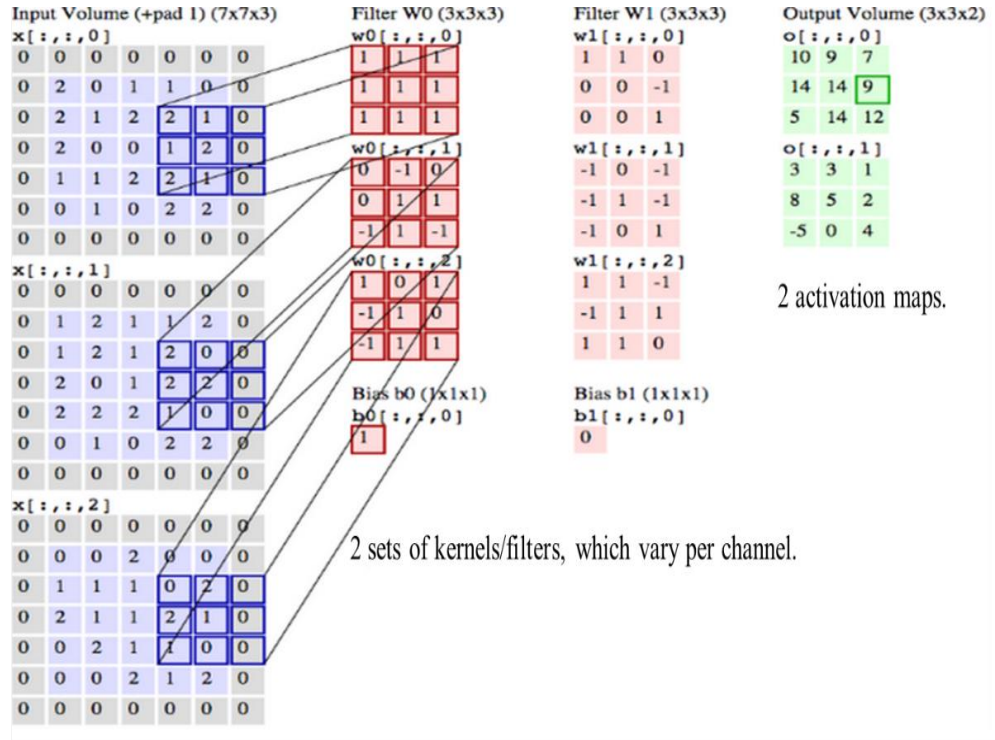
- Input layer:** In the CNN, this is the first layer, and it is given at a specific size. In this layer, the size of the image has significant importance in the success of CNN. If the size of the input image data is increased, this might result in both the system and the training process having better success under different circumstances [62]. If the size of the input image data is small, although a reduction in the training process will occur, this might also decrease the success of the system. If the size of the input image data is big, although this might increase the success of the system, this might also cause an increase in the training process. Therefore, it is necessary that the image data selected is of a size that is appropriate for the system that is being designed. The volume of the input layer comprises an image with the dimensions of [width  $\times$  height  $\times$  depth]. It is a matrix of pixel values.
- Convolutional layers (Conv):** This comprises the main layer in a CNN, which is used often in image feature extraction. The process of feature extraction is performed by filtering the images at a series of filters, known as Conv kernels. These filters perform these Conv operations as they scan the dimensions of the input image. There are different Conv kernels, which each extract different characteristics from the input data. The more Conv kernels that there are in the Conv layer, the more features can be extracted from the input data. The key design hyperparameters of the Conv layer usually involve the kernel size, the number of kernels, and the kernel stride. Theoretically, the 'size' is the filter size. These filters might be of different sizes; as an example:  $2 \times 2$ ,  $3 \times 3$ ,  $5 \times 5$ , or  $7 \times 7$ . The 'number' is the number of feature maps that can be obtained from the upper layer using a Conv filter, and the 'stride' is the number of pixels, that is, the displacement, through which the filter matrix can be slid over the input matrix. If a stride of 1 is used, then the filters are moved one pixel at a time. If a stride of 2 is used, then the filters move 2 pixels at a time as they are slid around. Using a stride that is larger will result in smaller feature maps. Hence, sometimes, it is best to pad zeros along the border of the input matrix so that the filter can be applied to the elements on the border of the input image matrix. A useful feature of padding it with zeros is that it provides control of the feature map size. The resulting output of this layer is

referred to as an activation map or feature map. It is produced by applying the activation functions, such as a Sigmoid, Tanh, or rectified linear unit (ReLU). A depiction of a Conv layer is presented in Figure 2.2.

Extraction of the image Conv feature is performed on an  $(n_h \times n_w)$  image by setting the size of the Conv kernel filter to  $(w \times w)$  with a stride of  $k$ . Then, it is possible to generate a feature map that has a size of  $\left(\frac{n_h - w + k}{k} \times \frac{n_w - w + k}{k}\right)$ .

Basically, the smaller the size of the Conv kernel, the higher the quality of the feature will be. However, the determination of the size should be based on the input image size.

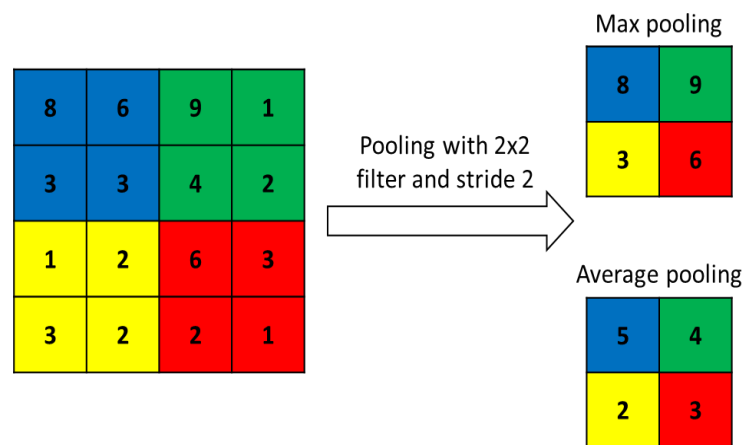
Thus, the goal of training a deep CNN is to learn a set of kernels that can detect the presence of certain visual characteristics within an image. Kernels at the input of the network will learn to detect the presence of certain lower-level features, such as lines and edges. In contrast, kernels at the output of a network will learn to detect the presence of more higher-level, field-specific visual characteristics of an image.



The input image's 3 color channels.

**Figure 2.2** An illustration showing the operations of the Conv layer.

- Pooling layers (pool):** In CNNs, the main purpose of the pooling layer, which is most often positioned after the Conv layer, is to reduce the input matrix size by removing any unnecessary information and preserving only the most important information. As with the Conv layer, selected filters in the pooling layer are defined, which can be moved using a specific step on the image. Calculating the pixel values can be performed in 2 ways: average and maximum pooling (max-pooling) [61]. In max-pooling, the maximum image pixel values are chosen, while in average pooling, the average image pixel values are selected. There are 2 significant parameters in the pooling layer, which comprise the filter size and the stride. Usually, pooling layers consist of small filters, such as  $2 \times 2$ . Some disadvantages exist when using the layers in a network. When using average pooling, the activation value can be minimized due to negative and low values. When implementing max-pooling, useful and discriminative information can be lost in that layer due to the fact that it will ignore all of the low values, which will probably cause the dataset to be over-fit quite quickly [63]. Examples of max- and average pooling are given in Figure 2.3.

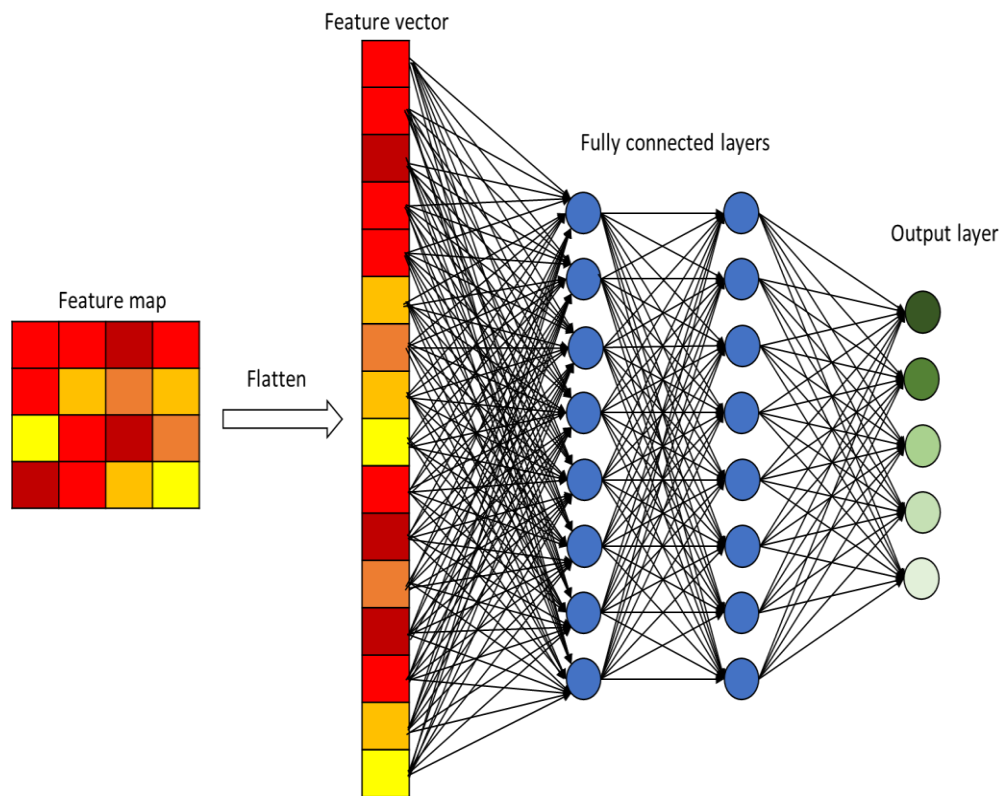


**Figure 2.3** Examples of max- and average pooling with a  $2 \times 2$  filter and stride equal to 2.

- Fully-connected layers (FC):** In the architecture of CNNs, once the dimensionality of the input has sufficiently decreased in size, the output, which represents high-level features of the input images, is often passed on to one or more of the FC layers, which are positioned after the Conv layer and the pooling layer.



The layers are called FC because the units in each layer are connected to all of the nodes within the previous Conv or pooling layers. And how many of these layers there are based on the CNN structure used. The number of weights will be equal to that of the nodes in the Conv layer, multiplied by the number of nodes that are in the connected layers [64]. The purpose of the FC layers is to use the extracted high-level features in the classification of the input image into different classes. Aside from this, adding an FC layer is also a method that is easy to use for learning the non-linear combinations of the features [65]. Figure 2.4 demonstrates the layout of a simple network with 2 FC layers.

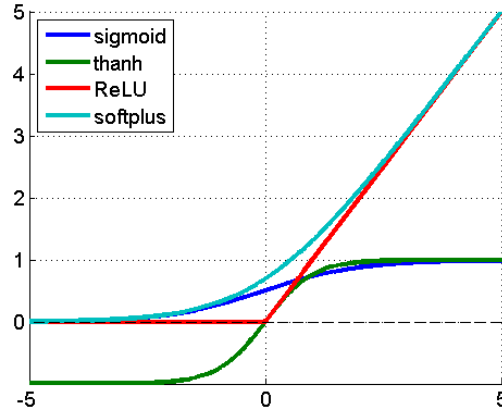


**Figure 2.4** Simple network with 2 FC layers.

- **Output (classification) layer:** The classification layer in a CNN, which is positioned after the FC layer, is where the learning process takes place. It provides probability distribution over the output classes [62]. Here, multiple loss functions can be used, such as SoftMax, which is the most common, but also Sigmoid or Euclidean. SoftMax is a linear classifier that uses log probability distribution.

### 2.2.2 Activation Functions

Activation functions in NNs provide an abstract depiction of the action potential of a cell [66]. They aim to map the output of a layer into the desired range, such as between 0 and 1. This introduces non-linearities into a CNN, increasing the capability of a network to learn more complex mappings. Generally, the activation functions that are used most often in the NNs are the Sigmoid, Tanh, SoftPlus, and ReLU functions. Figure 2.5 shows the activation functions that are mostly used in NNs.



**Figure 2.5** Activation functions that are mostly used in NNs.

The first 2 activation functions are mostly used in traditional backpropagation NNs. Because the ReLU function is in the form of an unsaturated, linear, and unilateral suppression that has sparse activation is used more in CNNs [67], where all of the Conv layers and the first 2 of the fully-connected layers are then activated via the ReLU function [68] which is responsible for a great deal of the recent success of deep NNs. Assuming that the neural node activation function is  $h^{(i)}$ , the ReLU function can be expressed as:

$$h^{(i)} = \max\left(\left(w^{(i)}\right)^T x, 0\right) = \begin{cases} \left(w^{(i)}\right)^T x, & \left(w^{(i)}\right)^T x > 0 \\ 0, & \text{Otherwise} \end{cases} \quad (2.1)$$

Here,  $i$  is the number of hidden layer nodes and  $w^{(i)}$  is their weight. The advantages of the ReLU are that it is more computationally efficient to compute while also minimizing the negative impacts of the vanishing and exploding gradient problems



[69]. Furthermore, deep CNNs that have ReLU activation functions are able to train several times faster than their equivalents using other activation functions [67]. However, the ReLU experiences a problem that is known as ‘*dying ReLU*’, where weights are updated in such a way that the neuron never activates. Hence, the gradient flowing through the neuron will remain at zero. Attempts have been made to resolve this by methods such as the ‘*Leaky ReLU*’ [70], where instead of the output of a ReLU being set to be zero if  $x < 0$ , it is instead set to a very small value.

### 2.2.3 SoftMax Classifier and Loss Function

When a CNN is used to conduct a task such as image classification, there is a need for a function to be appended to the output layer that will take whatever the values are and change them into a probability distribution. SoftMax [71] is a function that can transform a vector with  $K$  real values into a vector with  $K$  real values that has a sum that is  $= 1$ . It does not matter if the input values are negative, positive, 0, or  $>1$ , because SoftMax can transform them into values that are between 0 and 1, so they can then be interpreted as probabilities. SoftMax activation is used when performing multiclass classification because it is able to ensure that all of the activations in a single layer will have a sum that is  $= 1$ . The SoftMax activation function is shown below, in Equation (2.2), in which  $K$  is the number of activations and  $z$  is the vector of the activations,

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{For } j = 1, \dots, K \quad (2.2)$$

The goal of SoftMax regression is solving multiple classification problems. As label  $y$  may have  $k$  different values, rather than only 2, for the training dataset  $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ , we have  $y^{(i)} \in \{1, 2, \dots, k\}$ . For a given test input  $x$ , the target is to estimate the probability value  $p(y = j|x)$  for each category  $j$  by the hypothesis function. That is, to estimate the probability of each category of  $x$ . As a consequence, the hypothetical function will give a vector that has a  $k$  dimension, where the sum of the vector element is 1, as output, which represents the probability value of the estimate. Specifically, the hypothetical function  $h_\theta(x)$  determined herein is as given below:

$$h_{\theta}(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; \theta) \\ p(y^{(i)} = 2 | x^{(i)}; \theta) \\ \dots \\ p(y^{(i)} = k | x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \dots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \quad (2.3)$$

The loss function is an important function in CNNs. It measures the error between the value that the model predicted and what the value actually is. To say it in a more simple way, calculation of the gradients is performed by using the loss, and the weights of the CNN are then updated using the gradients. The use of the loss function is based on the task to be solved, and it directly affects the speed at which the training converges. In most instances, CNNs make use of a cross-entropy loss function, which is an indication of the accuracy of the network. When initializing the network using random values, there will be a high loss function. However, the goal when training the network is to decrease the loss function to as low as is possible. A CNN that has a low loss function will classify the training set with higher accuracy. Formulation of the formula used for the cross-entropy loss for the problem of multi-class classification is as follows:

$$L_{CrossEnt.} = - \sum_{n=1}^N \log(p(n))q(n) \quad (2.4)$$

Here,  $p(n) \in [0, 1]$  is the probability that the CNN model predicts that the input image belongs to the  $n$ -th class, and  $q(n)$  represents the distribution of the image labels.

#### 2.2.4 Training of Deep Convolutional Neural Networks

CNNs are trained depending on the training data that is used, as is most every machine learning problem. Learning algorithms, which are often called optimizing algorithms, are used to train the CNNs by the means of minimizing the loss function dependent on various learnable parameters, such as weight, bias, etc. Training is performed iteratively. Hence, in each of the iterations, the network parameters are changed to improve performance. When training a NN, the user has to give it an example of the input to be used. When the learning begins, the network will have no idea of how to behave toward the input, because the parameters will be initialized to small random

numbers. The forward pass will obtain a result that will probably be wrong. To correct the NN, it is necessary to define the measure of error, which is known as the loss or cost. Computation of the loss is performed through an evaluation of the loss function on the desired output, as well as a prediction of the training example. Then, the learning algorithm has to ‘learn’ how to transform all of the network parameters, so as to decrease the loss. As a consequence, calculation of the negative gradient of the loss with regard to the parameters is performed by applying the chain rule recursively, layer-by-layer toward the input. This process is known as backpropagation, and it is repeated for each of the examples. Then what is known as a fraction, which is the learning rate of the average of all of the negative gradients that have been obtained, is added to the weights, which updates them. This results in the optimization of all of the trainable network parameters, as well as convergence to local minima [71]. Many different algorithms are used when training NNs. The most commonly used algorithms include Stochastic gradient descent (SGD) [71] and adaptive moment estimation (Adam) [72]. SGD is likely the optimization algorithm that is used the most in machine learning, and, in particular, in the DL. SGD is expressed as shown in Equation (2.5).

$$\theta_{i+1} = \theta_i - \alpha \nabla_{\theta_i} L(\theta_i) \quad (2.5)$$

Here,  $\theta_{i+1}$  is the newly-learned parameters,  $\theta_i$  is the current parameters,  $\alpha$  is learning rate,  $\nabla_{\theta_i}$  is a gradient that is relative to  $x_i$ , and  $L(\theta_i)$  is the loss function.

Adam is also a commonly used adaptive-learning rate optimization algorithm. The name ‘Adam’ derives from the phrase ‘adaptive moments estimation’, and it was named this because it makes use of the 1<sup>st</sup> and 2<sup>nd</sup> moment estimations of the gradient to transform the learning rate for each of the weights in the NN. The 1st moment is the mean of the gradient, while the 2nd moment is the un-centered variance of the gradient, respectively. To perform the estimation of the moments, Adam makes use of averages that move exponentially, which are computed on the gradient as follows:

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) g_t \quad (2.6)$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2) g_t^2 \quad (2.7)$$

Here,  $m$  and  $v$  are the moving averages of the gradient and squared gradient,  $g$  is a gradient on the present mini-batch, and the  $\beta$ s are the initial decay rates used when estimating the 1<sup>st</sup> and 2<sup>nd</sup> moments of the gradient, which have default values, respectively, of 0.9 and 0.999. The moving average vectors are initialized using zeros at the first iteration, leading to moment estimates that are biased towards zero. This bias in the initialization can easily be counteracted, which then results in bias-corrected estimates for the 1<sup>st</sup> and 2<sup>nd</sup> moments,  $\widehat{m}_t$  and  $\widehat{v}_t$ .

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.8)$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.9)$$

Then these bias-corrected estimates are used to update the parameters by using the Adam update rule:

$$\theta_{t+1} = \theta_t - \alpha \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} \quad (2.10)$$

Here,  $\theta_{t+1}$  is the newly-learned parameters,  $\theta_t$  is the current parameters,  $\alpha$  is the learning rate, and  $\epsilon$  is the step size, which has a default value of  $10^{-8}$ .

In addition to the weight and bias parameters, which are optimized during training to produce the most accurate prediction results, CNNs also contain a set of parameters that are known as hyperparameters, which are predefined during the design of the network architecture, rather than learned during training. There is a wide variety of different types of hyperparameters within a CNN. Some relate to the size and architecture of a network, like the number of hidden layers, or the number of units that are in each of the layers. Moreover, other hyperparameters define how a network trains, such as the learning rate or batch size. Clearly, some hyper-parameters are dependent on others, such as the number of units within a particular hidden layer being

dependent on the number of hidden layers [73]. Hyperparameters are generally estimated via repeated training on a given data set with a variety of different values, observing the performance of the network following each trial [74].

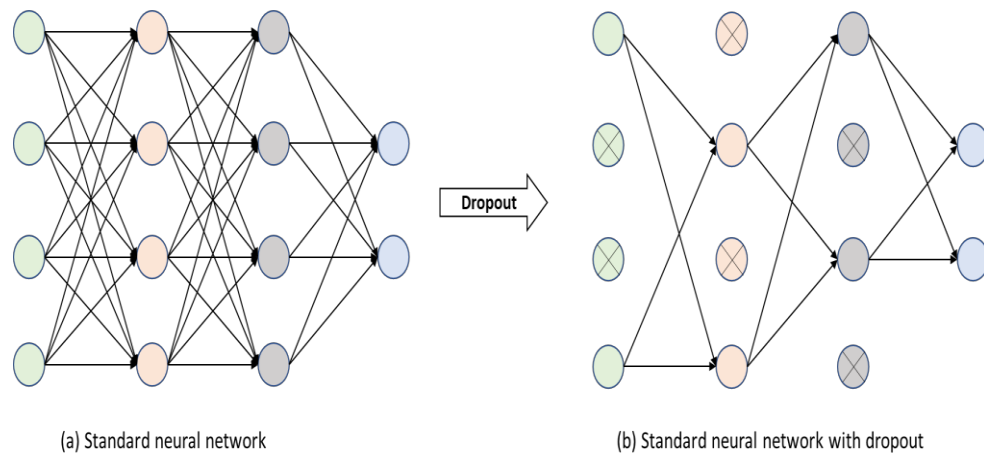
### 2.2.5 Handling an Overfitting in Deep Convolutional Neural Networks

Overfitting occurs when the CNN model achieves a good fit on the training data. However, it will not generalize well on new or unseen data. This means that the model will learn patterns that are specific to the training data, but they will be useless in other data. To reduce the occurrence of overfitting in DL models, it is best to attain a higher quantity of training data. However, unfortunately, in real-life situations, this is often not possible due to budget, time, or technical constraints.

Several approaches have been proposed to prevent the occurrence of overfitting in deep CNN models trained on fairly small datasets. In a specification, random cropping [75], random flipping [76], and random erasing [77] operations are commonly used as methods for data augmentation when training deep CNN models. Moreover, regularization methods are also a well-known approach for preventing overfitting. One of the most common regularization techniques is to apply  $L_1$  or  $L_2$  regularization [78], which penalizes complex models by adding an additional term to the loss function. In recent years, batch normalization (BatchNorm) [79] and dropout [80] have been 2 commonly used methods when training CNNs, and they have exhibited benefits when trying to prevent overfitting. Dropout aims to randomly, using a probability, remove the output of each of the hidden neurons throughout the training process. BatchNorm aims to reduce the internal covariate shift (ICS) via normalization of each hidden neuron's output using the mean and variance of the minibatch. Because person Re-ID datasets are quite small, effective means to prevent overfitting are required to build a high-accuracy person Re-ID model. In this work, to prevent overfitting in the proposed deep CNN model Dropout and BatchNorm are used.

- **Dropout:** Dropout is a method that is used for optimization in the training of NNs. It prevents overfitting by randomly excluding individual units within a layer at a prespecified rate [80]. Dropout introduces noise into the training process by randomly ignoring a percentage of the units within a layer. These non-working units are not calculated; however, their weights are temporarily kept, although not

updated, because they might work at another time with the next sample input. As the decision for which units to drop is random and carried out on an epoch-by-epoch basis, the units which are dropped differ every epoch and, therefore, make it harder to overfit, given the variability of the data that has passed through the network [80]. The structures of the NN, without dropout as well as with, were compared and are given in Figure 2.6.



**Figure 2.6** Schematic diagram of Dropout.

An advantage that Dropout has is its computational cheapness. When using dropout for training, it only requires  $O(n)$  computations for each example for each update, for it to generate  $n$  random binary numbers and then multiply them by the state. It also does not have major limits with regards to the model type or training procedure to be used. It works very well with almost all models that use a distributed representation, and it can be trained with SGD.

- **Batch Normalization:** In the NN deep training process, the ICS phenomenon is often observed, which can be defined as the change in the network activation distribution as a result of the change in the network parameters that occur during training. This slows the training because it requires lower learning rates and very careful initialization of the parameters, and as is well-known, it is extremely difficult to train a model that has saturating nonlinearities. BatchNorm attempts to reduce the ICS, and in doing so, causes a dramatic acceleration in the training of deep neural nets. It is able to do this through its normalization step, i.e., when

computation of SGD is performed, the activation output corresponding to this is normalized via mini-batch; hence, the mean of the results will be 0, and the variance will be 1. This means that the output, which was going to decrease, instead increases. BatchNorm also has a beneficial effect on the gradient flow through the network, by reducing the dependence of gradients on the scale of the parameters or of their initial values. This allows using much higher learning rates without the risk of divergence. Furthermore, BatchNorm regularizes the model. Finally, BatchNorm makes it possible to use saturating nonlinearities by preventing the network from getting stuck in the saturated modes [79].

Furthermore, overfitting can be caused by training the network for too long on the training data [81]. To prevent this, early stopping is used to ensure that the network stops training once the network's performance stops increasing. Therefore, the weights which give the best prediction performance are preserved.

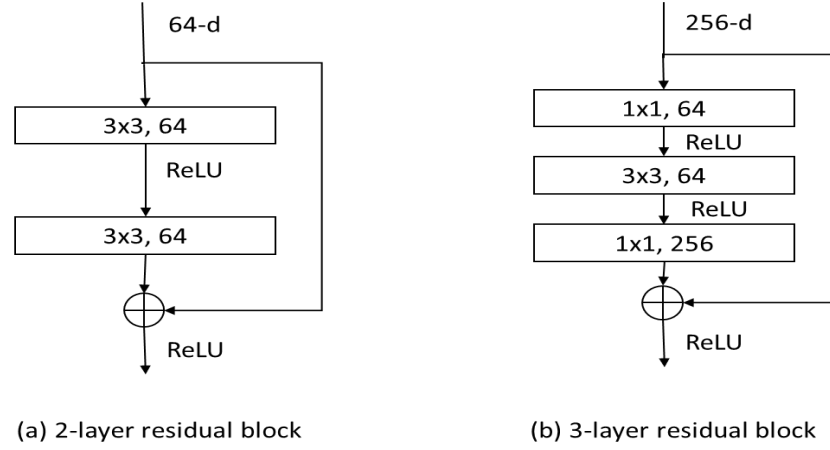
### 2.2.6 Deep Convolutional Neural Networks Architectures

In the literature, several CNNs architectures power modern computer vision applications and are extensively used in the person Re-ID task. This section explains the well-known CNN architectures that have been used in this work, namely ResNet [82], DenseNet [83], and GoogLeNet/Inception Network [84]. These are all very different networks, and each is specialized for specific problems.

- **ResNet:** ResNet is the residual CNN architecture, which was originally developed for object recognition by He et al. [82]. This deep NN was trained on 1.28 million images from ImageNet [85]. ResNet achieved overwhelming success in the large-scale visual recognition challenge 2015 (ILSVRC2015) and claimed 1<sup>st</sup> place for the detection, classification, and localization tasks on the ImageNet dataset and the detection and segmentation tasks on the COCO dataset [82].

The essential motivation behind ResNet is resolving the degradation problem, in which, when the network depth increases, the accuracy becomes saturated and then rapidly degrades. However, with the increase in the depth of the model, the network learning ability strengthens, but the deep model should not attain a high error rate. The reason for the degradation is that it is difficult to optimize the network. This is why a residual structure was proposed. ResNet block is either 2

layers deep, which are used in small networks, such as ResNet 18, 34, or 3 layers deep, such as with ResNet 50, 101, or 152. Figure 2.7 gives a depiction of the structure of the residual learning building blocks.



**Figure 2.7** Typical residual learning building blocks. ResNet-18/34 building block (left) and ResNet-50/101/152 building block (right).

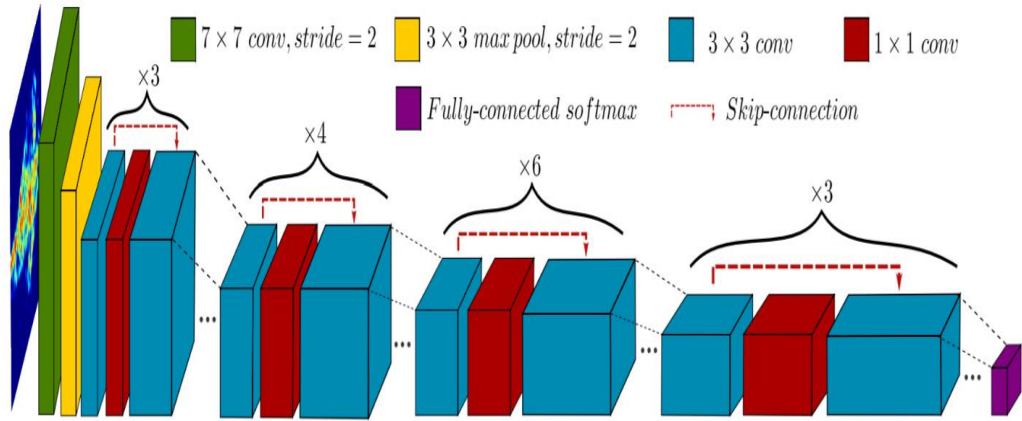
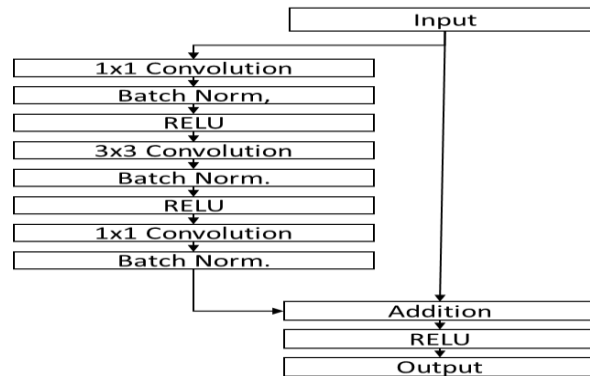
The ResNet network architecture has many different configurations, which comprise 18, 34, 50, 101, and 152 layers [82]. The first 2 are constructed by using a 2-layer residual building block, while the last 3 are constructed by using a 3-layer residual building block. Table 2.1 exhibits the detailed description of the different ResNet network architectures. In this thesis, the ResNet-50 model is used, considering its competitive performance as well as its comparatively brief building style. The ResNet-50 network architecture is given in Figure 2.8.

A detailed description of ResNet-50 is given as follows. This network begins with the input layer, which makes use of a  $224 \times 224 \times 3$  RGB image. The ResNet-50 model has 5 stages, each of which has a Conv block and identity block. Each Conv block, as shown in Figure 2.9, has 3 Conv layers. Each Conv layer has Conv, ReLU, and BatchNorm layers. The shortcut connections are used to perform the identity mapping, and then their outputs are added to those of the stacked layers. The 3 Conv layers consist of  $1 \times 1$ ,  $3 \times 3$ , and  $1 \times 1$ . The  $1 \times 1$  layer must reduce the dimensions and then increase, i.e., restore, them. The final layer in the network is a global average pooling layer, followed by a 1000-way FC layer with SoftMax. The ResNet-50 has over 25 million trainable parameters [82].



**Table 2.1** Architecture of ResNet. Building blocks (in brackets) with the number of stacked blocks [82].

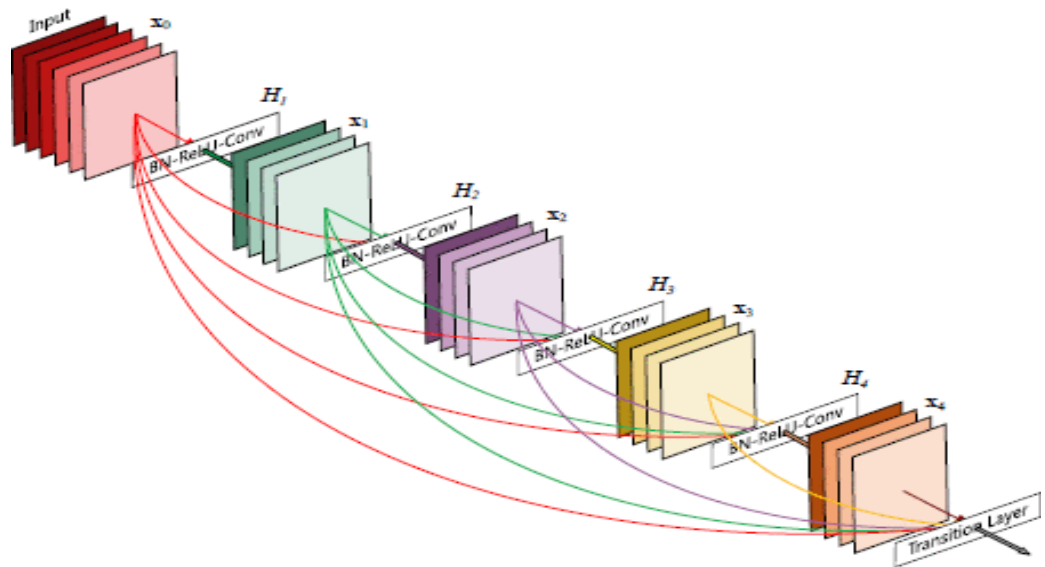
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

**Figure 2.8** Architecture of ResNet-50. Down sampling by a stride of 2 is applied before each residual block. ReLU activation is utilized for all of the layers except SoftMax for the output layer [86].**Figure 2.9** Illustration of a typical residual block of ResNet-50.

- **DenseNet:** Densely connected convolutional networks or DenseNets [83] are a recently proposed CNN architecture, that have an interesting connectivity pattern, called dense connectivity. DenseNet introduces connections from each of the layers to all of the other subsequent layers within the dense block, where one layer could receive all of the feature maps from the previous layers. Figure 2.10 shows the structure of the dense block. Each layer operates on all the input feature maps in the following transformations: BatchNorm, a ReLU, and Conv with a  $3 \times 3$  kernel. Symbolically, it would look like the following: the  $l^{\text{th}}$  layer is given the feature-maps of all the preceding layers,  $x_0, \dots, x_{l-1}$ , as input:

$$X_l = H_l(x_0, x_1, x_2, \dots, x_{l-1}) \quad (2.11)$$

Here,  $[x_0, x_1, \dots, x_{l-1}]$  is the concatenation of the feature-maps that are produced in layers  $0, \dots, l-1$ .



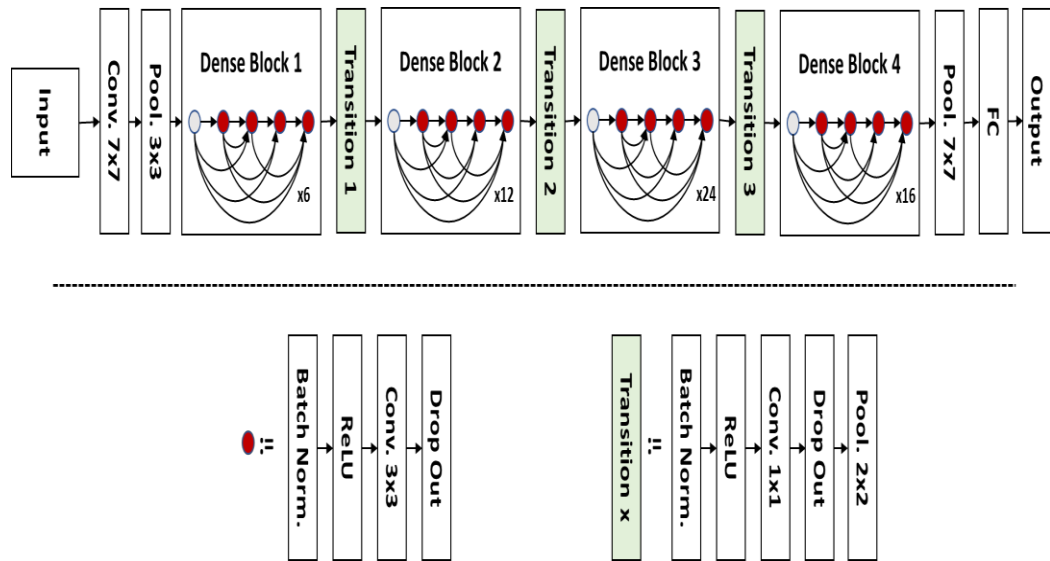
**Figure 2.10** Five-layer dense block that has a growth rate of  $k = 4$ . Each of the layers uses all of the preceding feature maps as inputs [83].

DenseNets have many interesting advantages, such as solving the vanishing gradient problem, strengthening propagation of the features, encouraging reuse of the features, and substantially decreasing the number of parameters. The DenseNet

network architecture has many different configurations, which comprise DenseNet-121, -169, -201, and -161. The growth rate for DenseNet-121, -169, and -201 is  $k = 32$ , while it is  $k = 48$  for DenseNet-161, where the growth rate is the number of feature maps produced by each DenseNet unit. Table 2.2 exhibits the detailed description of the DenseNet network architectures. In this work, the DenseNet-121 model was selected for the person Re-ID task process. DenseNet-121 yielded state-of-the-art accuracies on the ImageNet dataset. Also, it achieved accuracy results similar to ResNet via the use of less than half of the number of parameters and about half of the number of floating-point operations per second [87]. Figure 2.11 depicts the DenseNet-121 network architecture. The number 121 corresponds to the number of layers with trainable weights (excluding the BatchNorm layer). This network begins with the input layer, which makes use of a  $224 \times 224 \times 3$  RGB image. The final layer in the network is a global average pooling layer, which produces a  $1 \times 1024$  representation, and is followed by a 1000-way FC layer with SoftMax.

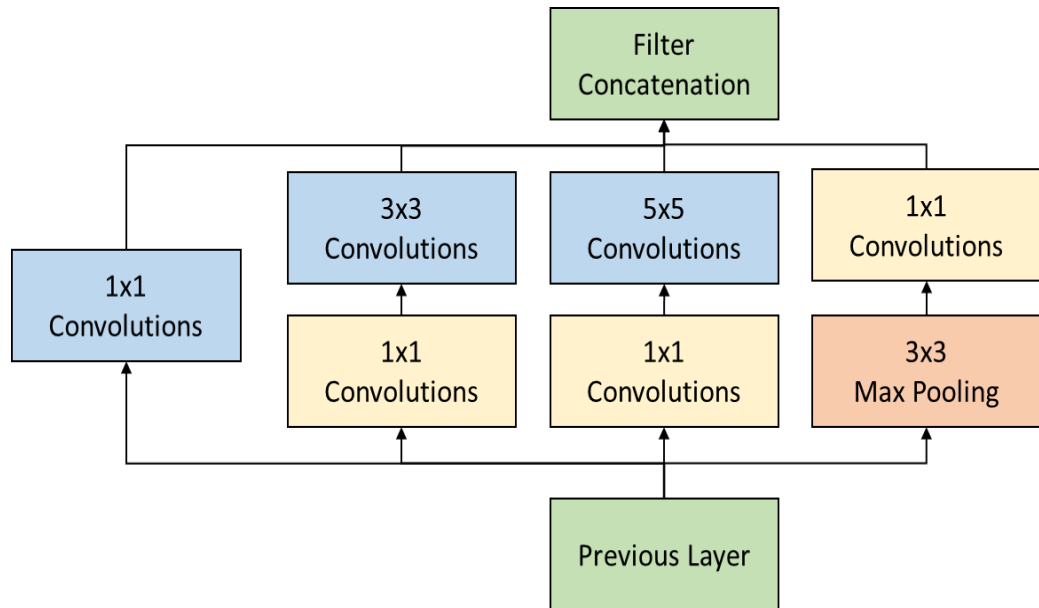
**Table 2.2** The architecture of DenseNets. Each of the Conv layers in the table corresponds to the sequence BN-ReLU-Conv [83].

Layers	Output Size	DenseNet-121( $k = 32$ )	DenseNet-169( $k = 32$ )	DenseNet-201( $k = 32$ )	DenseNet-161( $k = 48$ )
Convolution	$112 \times 112$	$7 \times 7$ conv, stride 2			
Pooling	$56 \times 56$	$3 \times 3$ max pool, stride 2			
Dense Block (1)	$56 \times 56$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	$56 \times 56$	$1 \times 1$ conv			
	$28 \times 28$	$2 \times 2$ average pool, stride 2			
Dense Block (2)	$28 \times 28$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	$28 \times 28$	$1 \times 1$ conv			
	$14 \times 14$	$2 \times 2$ average pool, stride 2			
Dense Block (3)	$14 \times 14$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition Layer (3)	$14 \times 14$	$1 \times 1$ conv			
	$7 \times 7$	$2 \times 2$ average pool, stride 2			
Dense Block (4)	$7 \times 7$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification Layer	$1 \times 1$	$7 \times 7$ global average pool			
		1000D fully-connected, softmax			



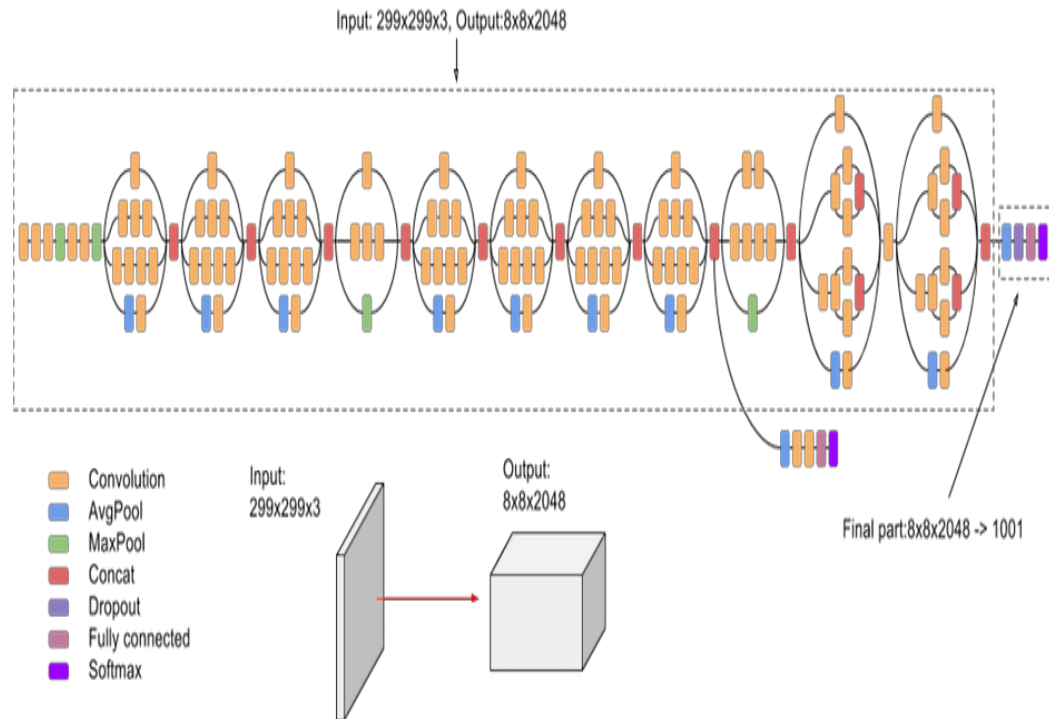
**Figure 2.11** The architecture of DenseNet-121. Layers that are between 2 adjacent blocks are called transition layers, as they change the sizes of the feature-map as a result of convolution and pooling.

- GoogLeNet/ Inception:** The GoogLeNet [88] architecture was introduced by Google and named Inception-v1. It won the ILSVRC competition in 2014. It represented a breakthrough in the effort for reducing the computation complexity of CNNs. GoogLeNet's main attractive feature is that it can run very fast due to a new concept known as the inception module that allows the network to choose between multiple Conv filter sizes in each of the blocks. The inception layer comprises the combination of a  $1 \times 1$  Conv layer, a  $3 \times 3$  Conv layer, and a  $5 \times 5$  Conv layer, and their output filter banks are integrated into one single output vector, which forms the input for the next stage. For dimensionality reduction, the  $1 \times 1$  Conv layer and the parallel max-pooling layer are used before applying another layer [88]. The inception module structure is presented in Figure 2.12. Since its development, the inception's architecture has been refined in several ways. First, there was the introduction of BatchNorm [79], which was known as Inception-v2 [84]. After that, additional techniques included regularization, dimension reduction, factorized convolutions, and parallelized computations in the 3rd iteration, known as Inception-v3 [84]. The aim of factorizing convolutions is to reduce the number of connections or parameters, without causing a decrease in the network efficiency [84].



**Figure 2.12** GoogLeNet Inception module with dimensionality reductions [84].

The Inception-v3 model possesses a fairly simple architecture when compared to the other high-performance NNs, a more complete architecture, and a modest computational cost. Thus, the Inception-v3 module is appropriate to use when the aim is to process large amounts of data at a reasonable cost, or when memory and computing power are limited. The architecture of the Inception-v3 model also further improves the classification effect on ImageNet [89]. Taking a very high-level view, the architecture of the Inception-v3, shown in Figure 2.9, can be summarized as follows: Inception-v3 is a 48-layer CNN with a  $299 \times 299 \times 3$  input layer. Convolutions and max-pooling convert the input image into an  $8 \times 8 \times 2048$  representation, and average pooling is used to bring this representation to  $1 \times 1 \times 2048$ . The network ends with an FC layer that maps the output of the average pooling layer to 1000 output predictions. One auxiliary classifier output branch is inserted between the layers to act as a regularizer during training and is then discarded during inference.



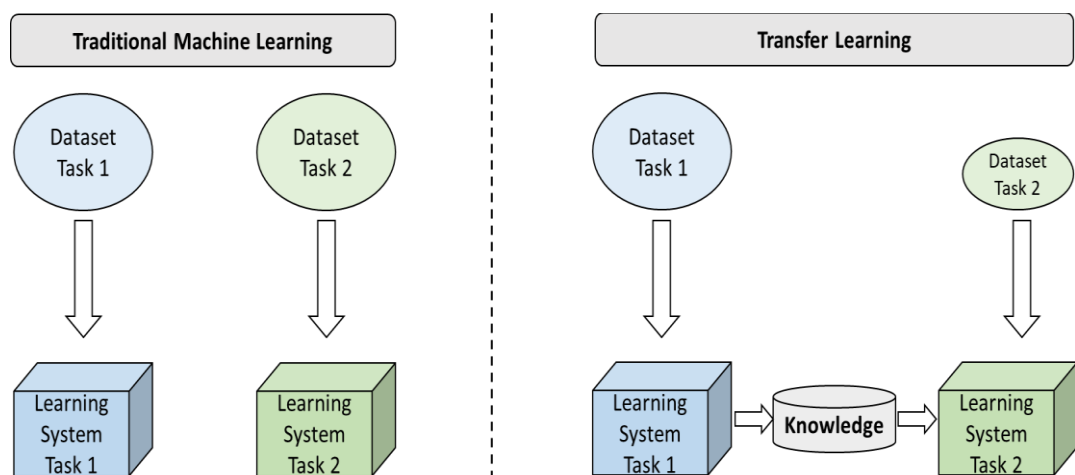
**Figure 2.13** Architecture of the Inception-v3 [84].

## 2.3 Transfer Learning for Developing Convolutional Neural Network Models

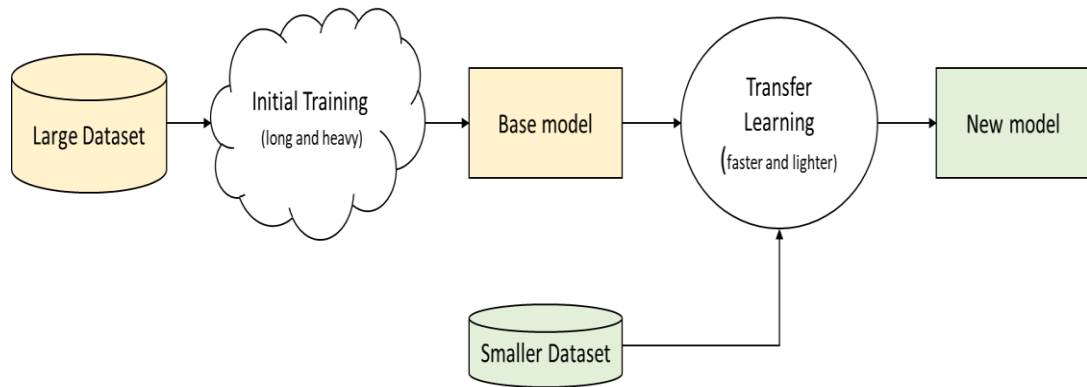
Training a deep CNN model from scratch can take significant time, as well as computational resources. Furthermore, CNN models require a large amount of labeled data to improve their performance. A lack of sufficient data is among the most difficult problems to overcome in the training of a deep CNN. Therefore, to decrease the resources necessary for training a network from scratch and tackle the lack of sufficient data, it is common for CNNs not to be initialized with random weights but with weights that have been pre-trained on a similar problem. This is analogous to how humans can use knowledge from one task to learn another [90]. This approach of learning is called transfer learning. Transfer learning, one of the branches of machine learning, focuses on using knowledge gained from solving different yet related problems while solving a particular problem [91]. This section explains the concept of transfer learning and how to use it when developing CNNs for computer vision applications.

### 2.3.1 The Concept of Transfer Learning

The concept of transfer learning generally refers to the idea of applying a model that trained on one problem in some way, on a second related problem. Generally, DL models are designed for a specific problem using training and testing datasets that are from the same domain space. Transfer learning makes an effort to change this via the development of algorithms that are able to use both training and testing data that were received from different domains or distributions. In actuality, this is quite important, as, in a lot of real-life applications, it is difficult to collect a sufficient amount of the training data that is required and then rebuild the models. If the transfer of knowledge is performed successfully, it is possible to remove the expense of data labeling and improve the learning capacity's performance [90]. Figure 2.14 shows the difference that exists between the traditional and transfer learning approaches as well as their learning processes [91]. The traditional learning approach aims to learn each of the tasks from scratch. However, the transfer learning approach works differently, as it aims to transfer some of the knowledge from previously performed tasks to the current target task. In the 2nd one, the available training data are fewer than in the first. Hence, the knowledge learned from performing the previous tasks is then used in place of the training data. The schematic diagram showing the general workflow of transfer learning is exhibited in Figure 2.15.



**Figure 2.14** The difference between traditional machine learning and transfer learning approaches.



**Figure 2.15** A schematic diagram of the general workflow of transfer learning.

Depending on the difference that exists between the task space and the domain and whether or not the training data is labeled, several settings can be used in transfer learning, which comprises inductive, transductive, and unsupervised transfer learning. In inductive transfer learning, the source and target tasks are different, and there is no importance placed on whether they were from the same or from different domains. In transductive transfer learning, the target and source tasks are the same; however, their domains are not. Finally, in unsupervised transfer learning, the source and target tasks are different; however, but they can be related [91]. In this research work, inductive transfer learning was used because the CNN models used had been trained previously for image recognition tasks. For re-use of these models in the person Re-ID problem, it was necessary to transfer the knowledge from image recognition to the person Re-ID domain. Now, the target task, which comprises person Re-ID, and the source task are different, as are their domains. With regard to CNNs, the use of transfer learning requires not just the re-training of the CNN using the new training data but also necessitates the fine-tuning of the pre-trained network weights while the backpropagation process is underway.

### 2.3.2 Transfer Learning with Fine-Tuning the Pre-Trained Deep Models

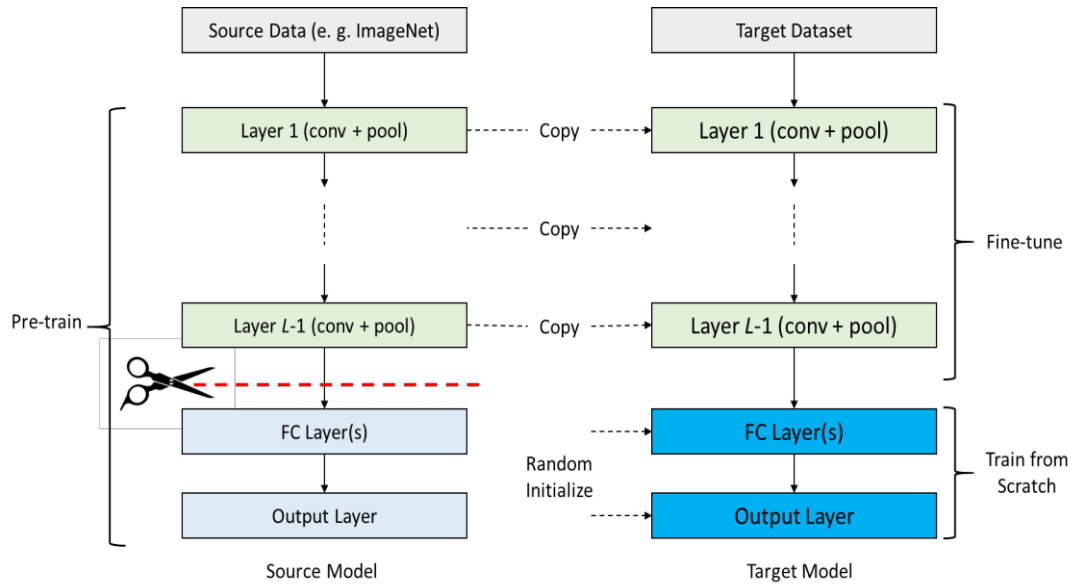
Fine-tuning is a transfer learning strategy that allows for successfully applying pre-trained CNN models for problems with small training datasets, such as person Re-ID. It can not only re-train the CNN model using the new dataset, but it can also fine-tune the weights of the pre-trained CNN model while the training process continues. A pre-trained CNN model is actually a saved network, which was trained previously on a



large dataset, usually on the ImageNet dataset for object recognition tasks [85]. The ImageNet dataset contains over one million images with bounding boxes, representing over 1000 different classes [92]. Thus, it contains an amount of knowledge appropriate for a vast number of related domains, one of which is person Re-ID. Basically, 2 important parameters make the fine-tuning procedure a good choice for the person Re-ID problem. First, the training dataset size is small, and second, it is very different than the original training dataset [91]. The pre-trained networks are used as a weight initialization for training. Initializing the weights of an attributed prediction network with ImageNet's pre-trained weights gives a head-start on training and increases performance while decreasing the time required to train the network. Furthermore, it also potentially reduces the chance of the network overfitting. Following the initialization of a network with pre-trained weights from another domain, the next step is to alter the architecture of the network to fit the target domain. This typically requires replacing the final FC layer to contain a number of units that are also equal to the number of classes within the target domain [93].

CNN models are layered architectures that hierarchically learn different features. These layers are then connected to the last layer, which is usually an FC layer, in supervised learning so as to attain a final output. This layered architecture allows for the utilization of the pre-trained CNN models, such as ResNet-50, DenseNet-121, etc., without the use of its final layers, as a fixed feature extractor that can be used for the other tasks. To perform the fine-tuning procedure, the FC classifier layers of the pre-trained model are removed, and new FC layers with random values are added to train on the new data and perform the new task. It is possible to fine-tune each of the layers or keep some and then fine-tune some others. It has been generally accepted that, to avoid overfitting, the lower layers should be kept the same, and only some of the higher layers should be fine-tuned. This is because the lower layers represent the more generic features, such as the edge and blob features, while the higher layers tend to contain more specific features, such as those regarding the problem space. These higher layers are jointly trained using the newly added classifier layers, which allows for fine-tuning of the higher-order feature representations in the pre-trained CNN model, making them more relevant for the task being performed. Figure 2.16 provides an illustration of the deep block structure of the CNN model for the fine-tuning procedure. It has been

demonstrated that utilizing transfer learning to transfer knowledge from one domain to another increases prediction results [53, 94, 95].



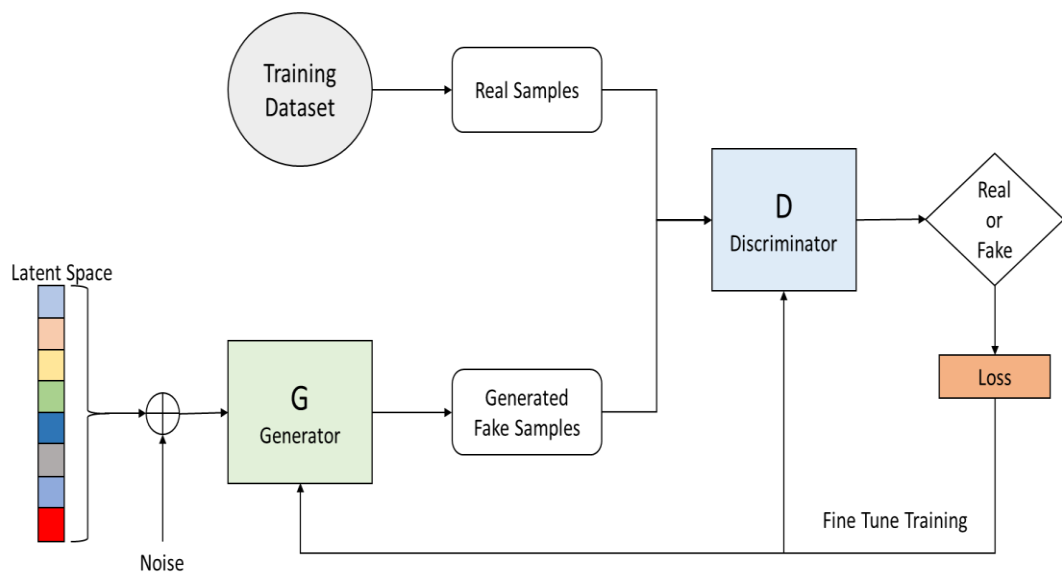
**Figure 2.16** The deep block structure of the CNN model for the fine-tuning procedure.

## 2.4 Generative Adversarial Networks

Despite the fact that deep CNNs have gained significant popularity in many computer vision tasks in the last few years, they still possess pervasive problems, such as limited data sets, class imbalance, and missing data. These issues occur because of the nature of data collection costs, data space, new markets, limitations, and absolute rarity [96, 97]. These issues create problems for building models, leading to inadequate data, inaccurate or misleading models, biased accuracy measures, and subjective uncertainty margins, which create further model risk [98]. For many computer vision problems, it is difficult to locate sufficient training data, such as in the person Re-ID. This research focuses on how to handle these problems with regards to the task of person Re-ID through generative modeling. Generative models are flexible models, which can learn the data distribution and then perform the sampling from that, thus resulting in the creation of new synthetic cases. Many learning algorithms for generative models have been proposed, including variational autoencoders [99], GANs [5], autoregressive models [100], and normalizing flow models [101]. GANs are believed to be of better quality than the other generative models [5, 71, 102-104], and they have had a

significant impact on person Re-ID because they allow the creation of new synthetic people images that can be seen from any desired viewpoint [53, 105, 106]. This section reviews the general principles of GANs and explains their common architectures.

GANs were launched originally in 2014, by Goodfellow et al. [5] as an approach of generative modeling using an adversarial process. GANs have gained a great deal of attention in the last few years, as they have been successfully applied in many tasks of computer vision, for example, but not limited to, image-to-image translation [107, 108], synthetic image generation [109-111], super-resolution imaging [112], photo inpainting [113], And others. A typical GAN contains 2 models, the generator model, and the discriminator model. In terms of images, the generator (G) learns to take random noise vector ( $z$ ) as input and generates synthetic images in the problem domain. The noise vector is pulled randomly from the latent space (Gaussian distribution) and used to seed the generative process. The discriminator (D) is a classification model that learns to distinguish between the real images that were taken from the training dataset and the synthetic images that the G model generated. The original GAN architecture is presented in Figure 2.17, which is the baseline model that all of the other GAN variants were based on.



**Figure 2.17** General architecture of the original GAN.

The set-up for the GAN model makes use of an adversarial process that can estimate the parameters of the 2 CNN models via iterative and concomitant training. The GAN training process can be considered as a zero-sum or minimax game. Through multiple generation and discrimination cycles, the models can train each other, and at the same time, each tries to outwit the other [5, 114, 115]. The G model tries to cheat the D model, while the D model tries not to be deceived by the G model. Through backpropagation, the D model updates its weights and provides a signal that the G model uses to update its weights. Once the G model has been optimally trained, it is able to create new samples and then alter the training data set.

Typically, the training process of the GAN model can be summarized as follows:

- G randomly chooses a sample,  $z$ , from the latent space that is defined by  $p(z)$ , and then it generates samples using this distribution. G has to learn the parameters  $\theta_g$ , given an input,  $z \sim p_z(z)$ , which will give the output,  $G_{\theta_g}(z)$ . G is trained to fool D, that is, to make the output by D for fake/generated sample  $D(G(z))$ , closer to 1. The parameters for G are learned by minimizing the GAN loss over  $\theta_g$ .
- D receives the generated samples from G, and also receives the real data samples from the real data distribution,  $p_{data}(x)$ , and then D has to distinguish the 2 samples of data for authenticity. The output that results,  $D_{\theta_d}(x)$ , for an input,  $x$ , is the probability of  $x$  being sampled from  $p_{data}(x)$  rather than  $p_g$ , where  $p_g$  is the implicit distribution defined by G. The vector,  $\theta_d$ , represents the parameters that were learned from D. Now, the goal for D is to attain a yield for  $D(x)$  that is near 1 for  $x \sim p_{data}$  and that for  $D(G(z))$  to be closer to 0 for  $p_z(z)$ . This can be achieved by maximizing the GAN loss over  $\theta_d$ .

The loss (objective) function of the GAN that the G model aims to minimize, while, at the same time, the D model aims to maximize is called the MiniMax GAN (MM-GAN), and it is formulated in Equation (2.12).

$$\text{Min}_{\theta_g} \text{Max}_{\theta_d} V(D, G) = E_{x \sim p_{data}(x)} [\log(D_{\theta_d}(x))] + E_{z \sim p_z(z)} [\log(1 - D_{\theta_d}(G_{\theta_g}(z)))] \quad (2.12)$$

Here,  $E_x$  is the expected value over that of the real data distribution, and  $E_z$  is the expected value over that of all of the fake data distribution. The  $\log(D(x))$  term in the

above function cannot be directly affected by the G model, thus, for the G model, being able to minimize the loss is equal to being able to minimize the  $\log (1 - D(G(z)))$ . training of the GANs alternate between the gradient descent on G and the gradient ascent on D [71]. Generally, for every training performed by G, D is then trained  $k$  number of times. This is shown in Algorithm 2.1. Gradient-based updates can be performed via any of the optimizers that were reviewed earlier in section 2.2.4. Most often, this is done using SGD with Momentum for D, and Adam for G, as these tend to do well in practice [5, 103].

**Algorithm 2.1** Mini-batch SGD training of GANs with the original objective for MM-GAN.,  $k$  is a hyperparameter. For every training performed by G, D is then trained  $k$  number of times [5].

---

```

1: for number of epochs do
2:   update the discriminator
3:   for  $k$  steps do
4:     • Sample mini-batch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from the noise prior  $p_g(z)$ .
     • Sample mini-batch of  $m$  true examples  $\{x^{(1)}, \dots, x^{(m)}\}$  from the training data distribution  $p_{data}(x)$ .
     • Update the discriminator D by ascending its stochastic gradient on these mini-batches:
       
$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log D(x^{(i)}) + \log (1 - D(G(z^{(i)})))] .$$

5:   end for
6:   update the generator
7:     • Sample mini-batch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from the noise prior  $p_g(z)$ .
     • Update the generator by descending its stochastic gradient computed on this mini-batch:
       
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(z^{(i)}))) .$$

8:   end for

```

---

### 2.4.1 The Derived Generative Adversarial Networks Architectures

Although the original architecture of the GAN (MiniMax GAN) promises to generate real-like data in the problem domain, it has many notable issues, including the fact that it is known to be difficult to train and evaluate properly, it is not easy to compute the likelihood, and it often experiences the vanishing gradient problem, boundary distortion, mode collapse, and overfitting [71, 102, 104, 116, 117]. Therefore, to avoid these problems, stabilize the training of GANs, and have them produce a sensible output, the original GAN architecture has been modified via several developments, such as changing the architecture and the objective function, and finding better ways

of optimizing GANs. These developments have led to the emergence of many recently developed architectures for GANs. Given the enormous number of them that cannot be all covered in this research, this section briefly describes some of the most common and popular GAN architectures, from which most, if not all, of the GANs that are used for the person Re-ID task, are built upon.

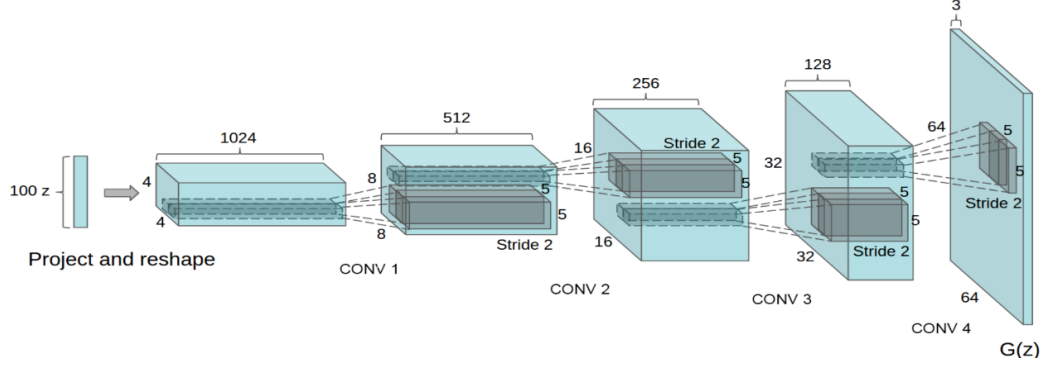
- **Conditional GAN (cGAN):** cGAN was the first extension of the original GAN, which introduced the class label  $y$  as a conditional variable in both the G and D models, making them class-conditional [118]. Feeding of the labels for the G and D models is performed using an extra input layer. In the G model, the initial input noise  $p_z(z)$  and  $y$  are combined in a joint and hidden representation. For the D model,  $x$  and  $y$  are presented as input for the discriminative function. While adapting these changes, the objective function of the minimax game, as proposed in the initial GAN work, would be formulated as in Equation 2.13.

$$\text{Min}_{\theta_g} \text{Max}_{\theta_d} V(D, G) = E_{x \sim p_{data}(x)} [\log(D_{\theta_d}(x | y))] + E_{z \sim p_z(z)} [\log(1 - D_{\theta_d}(G_{\theta_g}(z | y)))] \quad (2.13)$$

cGAN allows the creation of diversified samples and forces the G model to create certain samples, thus resolving the mode collapse problem. There are other cGANs, such as the auxiliary classifier GAN (AC-GAN) [114] and InfoGAN [119] for various tasks. These versions have been instrumental in many domains, such as image synthesis and face aging.

- **Deep Convolutional GAN (DCGAN):** DCGAN is an improved GAN architecture in which supervised learning with a CNN and unsupervised learning with a GAN are combined to give stable training. In the DCGAN architecture, standard CNNs are used to build G and D models [103]. Three core changes were adopted in the standard CNN architecture, namely replacing any pooling layers with strided convolutions, using BatchNorm in most of the hidden layers of both G and D, and eliminating the FC layers. DCGAN provides some further tricks using the transposed convolution for up-sampling in the G model. Figure 2.18 shows the DCGAN generator architecture. DCGANs are known for their remarkable speed,

stable training, and having better performance than GANs [103]. As a result, most GAN variants have the structure of a DCGAN.



**Figure 2.18** DCGAN generator architecture [103].

- **Wasserstein GAN (WGAN):** WGAN brought forth the proposal of a different loss function [104], and has become the most widely used and widely studies GAN architecture since its inception [117]. WGAN is able to provide better quality to the generated synthetic data than is possible with the original GAN, and it can also alleviate most of the issues with GANs [104]. WGAN has been amended through the addition of a gradient penalty (GP) in the cost function, which resulted in WGAN-GP [120], which can be defined using the loss function given below:

$$\begin{aligned} \text{Min}_{\theta_g} \text{Max}_{\theta_d} V(D, G) = E_{X \sim p_{data}(x)} [D(x)] + E_{Z \sim p_z(z)} [1 - D(G(z))] + \\ \lambda E_{x \sim p_{data}} \left[ \left( \left\| \Delta D(x) \right\|_2 - 1 \right)^2 \right] \end{aligned} \quad (2.14)$$

Here,  $\tilde{x}$  samples the uniformly along the straight line that is between the points sampled from  $p_{data}$  and  $p_g$  and  $\lambda$  is the GP term. WGAN-GP exhibits a better distribution of the learned parameters when it is compared with WGAN [120]; thus, it has been the default method that has been used in most of the GAN loss variants. Other commonly-used GAN loss variants include the least-squares GAN

(LSGAN) [121], Boundary Equilibrium GAN (BEGAN) [122], and Loss Sensitive GAN (LS-GAN) [123].

- **Cycle-Consistent GAN (CycleGAN):** CycleGAN is a type of GAN that is used for image-to-image translation with unpaired collections of images from 2 different domains [115]. CycleGAN has 2 generators,  $G_A$  and  $G_B$ .  $G_A$  gathers images from domain A and then translates them, and puts them into domain B, and then conversely for the  $G_B$ . For the 2 domains, A and B, CycleGAN learns the mappings  $a \rightarrow G_A(a) \rightarrow G_B(G_A(a)) \approx a$  and  $b \rightarrow G_B(b) \rightarrow G_A(G_B(b)) \approx b$ . These mappings should be the reverse of each other, and both mappings should be bijections. This is achieved through a cycle consistency loss. Combining this loss and the adversarial losses on both A and B fulfills the full objective for unpaired image-to-image translation. For the mapping  $a \rightarrow G_A(a) \rightarrow G_B(G_A(a)) \approx a$  and its discriminator  $D_B$  the objective function formulated as in Equation (2.15).

$$L_{GAN}(G_A, D_B, A, B) = E_{b \sim p_{data}(b)} (\log D_B(b)) + E_{a \sim p_{data}(a)} \left[ \log(1 - D_B(G_A(a))) \right] \quad (2.15)$$

Here,  $G_A$  aims to generate images,  $G_A(a)$ , that appear to be similar to the images from domain B, whereas  $D_B$  aims to differentiate between the translated samples  $G_A(a)$  and real samples  $b$ . A similar loss is postulated for the mapping  $b \rightarrow G_B(b) \rightarrow G_A(G_B(b)) \approx b$  and its discriminator  $D_A$ . The cycle consistency loss function will reduce the space for the possible mapping functions by enforcing forward and backward consistency. This is formulated as in Equation (2.16).

$$L_{cyc}(G_A, G_B) = E_{a \sim p_{data}(a)} \left[ \|G_B(G_A(a)) - a\|_1 \right] + E_{b \sim p_{data}(b)} \left[ \|G_A(G_B(b)) - b\|_1 \right] \quad (2.16)$$



The full objective is formulated as in Equation (2.17).

$$L(G_A, G_B, D_A, D_B) = L_{GAN}(G_A, D_B, A, B) + L_{GAN}(G_B, D_A, A, B) + \lambda L_{cyc}(G_A, G_B) \quad (2.17)$$

Here,  $\lambda$  is a hyperparameter that allows a user to determine the relative impact of the adversarial and cycle consistency losses. The optimal generators can be found by solving the equation below:

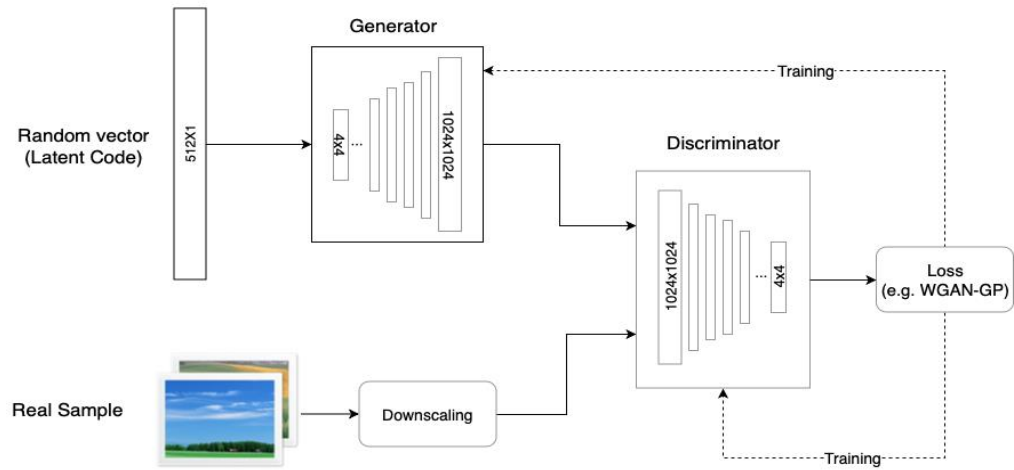
$$G_A^* G_B^* = \arg \min_{G_A, G_B} \max_{G_A, G_B} L(G_A, G_B, D_A, D_B) \quad (2.18)$$

CycleGAN can be used to convert different images to another, and to generate even data sets that have no alignment, for example, to convert zebras to horses, winter to summer, and so on.

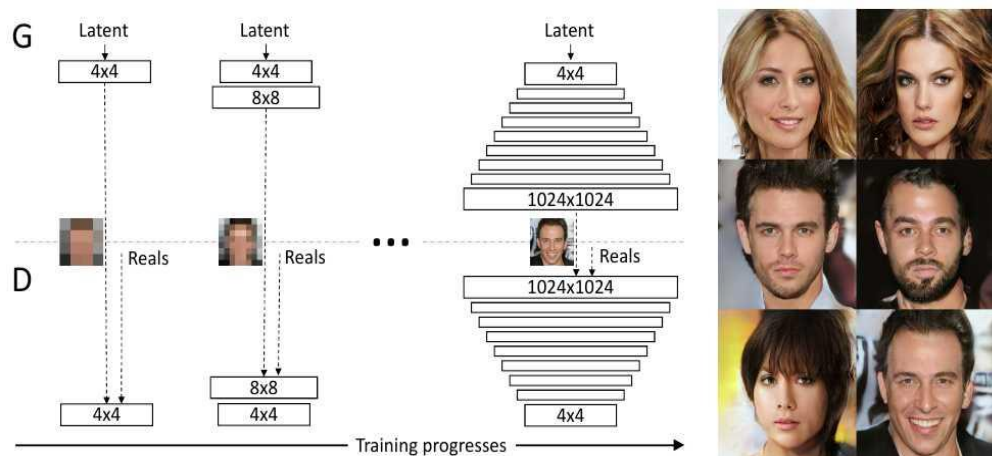
- **Progressive Growing GAN (ProGAN):** ProGAN is a recent advanced GAN architecture. It has a similar structure to DCGAN and proposes a new training methodology. Figure 2.19 shows an overview of the ProGAN architecture. The key idea of training ProGAN is to increase the size of both the generator (G) and the discriminator (D) progressively[110]. Figure 2.20 shows the training process that is used for ProGAN, which begins by using a low-resolution image and then, later, adding a high-resolution layer, each time, until reaching the desired high resolution. First, this technique creates a foundation for the image via learning about the base-line low-level features, and then it learns more details over time, as its resolution increases. It is not just easier and faster to training the low-resolution images; it also helps in training the higher levels. As a result, total training is also faster and leads to the generation of high-quality, high-resolution, fake images.

ProGAN can generate high-quality images but provides very limited ability to control the specific features of the generated images [124]. Even a small change in the input affects multiple features. Recent advancement in the GAN architecture,

called style-based GAN (StyleGAN), has further improved the progressive training of ProGAN, by redesigning the generator to adjust the style of each Conv layer and avoid feature entanglement [6]. StyleGAN has the same progressive training approach and same discriminator architecture as ProGAN. It proposes many changes in the generator, allowing it to generate high-quality photo-realistic images and have superior control over the visual features of the input images. StyleGAN was selected in this research to generate high-quality and highly diverse person images to improve the person Re-ID task effectively. More details will be given in Chapter (4).



**Figure 2.19** Overview of the ProGAN architecture.



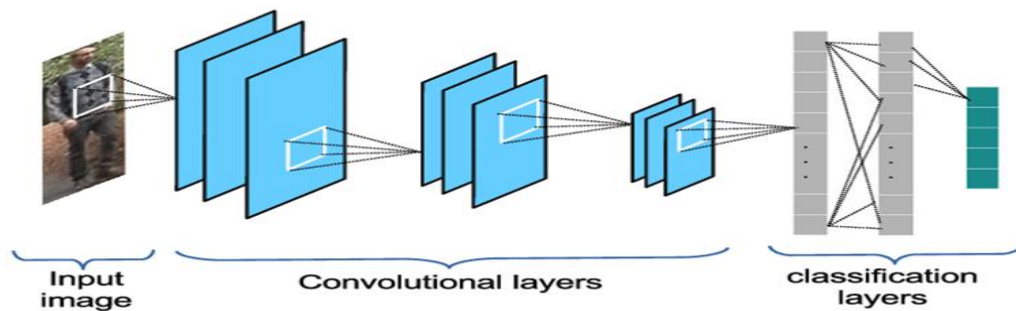
**Figure 2.20** Progressive training of the ProGAN, from the low-resolution to high-resolution layers for synthetic faces generation. All existing layers remain trainable throughout the process. The 6 face images shown on the right are sample images generated using progressive growing at  $1024 \times 1024$  pixels [110].

## 2.5 Taxonomy of Deep Learning-Based Person Re-identification Methods

DL-based methods have gained increasing popularity for person Re-ID tasks due to their successful application in numerous computer vision tasks. These methods can adaptively learn the features of pedestrians in images and learn the similarity metric with good results [125]. Several interesting DL-based models have been proposed for improving the performance of person Re-ID with DL development, either through modification of the existing DL architectures or the design of new deep models. Based on the type of deep model, there are basically 5 kinds of DL-based methods that have been designed for person Re-ID, which include those based on the identification model, the verification model, the distance metric-based model, the body part-based model, and the GAN model. These methods are prevalent and can be an indication of future trends in the person Re-ID community. The following sub-sections primarily give a summary of these types and focus on the state-of-the-art DL-based methods for person Re-ID in more detail.

### 2.5.1 Methods Based on the Identification Model

Here, the DL-based methods that regard the person Re-ID task as a classification issue [4] are discussed. In such methods, the deep model takes an image of a person, it then calculates the probability for the class corresponding to the person, and then it outputs the labels corresponding to the input person images [1]. Figure 2.21 illustrates the general deep architecture of the identification model.



**Figure 2.21** Basic architecture of the standard identification deep model [1].

Many works in the literature have employed identification deep models in an attempt to address the person Re-ID task. Wu et al. [44] proposed a fusion feature network (FFN) for feature extraction in person Re-ID. This model jointly utilizes both the CNN feature and the hand-crafted features, which include the color histogram and texture features. After this, a buffer layer follows both of these extracted features, and then an FC layer is used, which acts as the fusion layer. The buffer layer, which is essential, is used for the fusion action, bridging the gap that is between the 2 features, which are very different. Then, in the backpropagation phase, the CNN features are limited as a result of the variety in hand-crafted features. The training of the overall network is done by SoftMax loss. To produce linearly separable hidden representations for person samples with extensive appearance variations, in [126], a hybrid deep architecture was presented that was proposed for person Re-ID. This architecture comprised a deep CNN model that was combined with a Fisher vector to produce final non-linear features that could represent the person images. The network was trained in an end-to-end manner by optimizing a linear discriminant analysis (LDA) Eigen-valued based objective function, where the LDA-based gradients can be backpropagated to provide updated parameters in the Fisher vector encoding. Inspired by transfer learning, Zheng et al. [12], and [127], took direct advantage of the deep CNN models that had been pre-trained on ImageNet [85]. They fine-tuned them via an approach that is known as ID discriminative embedding (IDE). IDE views the person Re-ID training process as if it were a multi-class classification mission [128]. According to the IDE results reported in [127], it will be of great value to utilize such an approach to improve person Re-ID performance. Therefore, the most currently used DL-based methods also make use of the pre-trained models as backbones, with searches that focus on finding new technology that can be used to enhance the performance of person Re-ID systems [129]. In [45], Xiao et al. proposed a CNN model that could learn generic deep feature representations from many different data sets and discover effective neurons for each training data set. Specifically, they were able to produce a baseline model that was strong and able to work on many different data sets at the same time by combining both the data and the labels from more than one person Re-ID datasets, and then train the CNN using SoftMax loss. After this, for each of the data sets, they performed a forward pass on all of the samples and computed the average impact of each neuron

on the objective function. As a final step, they removed the standard dropout operation and replaced it with the deterministic domain guided Dropout to be able to remove the useless neurons in each of the datasets, and then train the CNN model for several more epochs. The newly-learned generic embedding after the domain-guided dropout yielded competitive performance for person Re-ID. It was proposed by Su et al. [130] that the deep representation learning process could be optimized with singular-vector decomposition (SVD). Specifically, using the restraint and relaxation iteration (RRI) training scheme, they were able to iteratively integrate the orthogonality constraint in the training of the CNN, which yielded what is known as SVDNet. Experimental results of SVDNet on person Re-ID datasets have demonstrated that significant performance improvement was achieved with competitive Re-ID accuracy. Afterward, Zheng et al. [131] proposed an identification task that could learn pedestrian alignment. They designed a pedestrian alignment network (PAN) that was able to learn the person descriptors, and at the same time, align them with a bounding box..

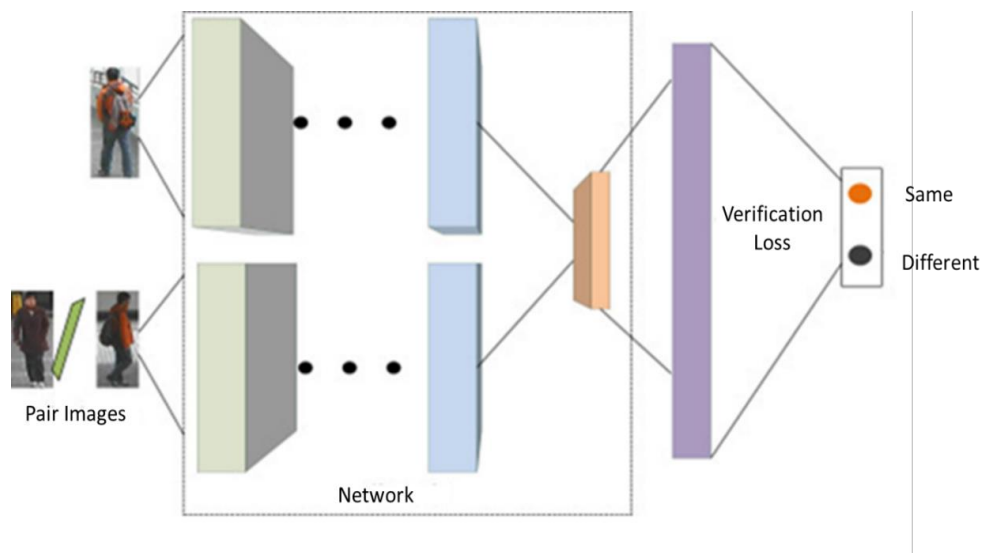
To fill the gap between the real-life scenarios and those of the person Re-ID datasets, Xiao et al. [132] designed a deep network architecture that could handle person detection and Re-ID tasks in an end-to-end manner. In their study, they introduced a random sampling of SoftMax loss for use in training the deep architecture using unbalance and spare labels. Not long after that, Xiao et al. [133] conducted further research on the joint learning system and thus designed an online instance matching (OIM) loss function that could be used in place of the commonly used SoftMax loss in the training of the network. When compared with SoftMax loss, the OIM is non-parametric. Hence, it is able to exploit an unlabeled pedestrian using a circular queue. However, the OIM loss has one drawback, which comprises the tendency toward easily over-fit because it is non-parametric.

Overall, the input of the identification model not difficult to perform, and it is able to fully use the label information that is within the datasets [12], which aids in it being able to improve the training efficiency. However, the training object of the identification model is not consistent with its test method; thus, this affects the accuracy of the results. Additionally, in the current person Re-ID datasets, the scales are very small, which means that the identification model cannot entirely be trained.

In this research, an identification baseline model was developed based on a pre-trained CNN as a backbone. The proposed model was fine-tuned utilizing the IDE approach.

### 2.5.2 Methods Based on the Verification Model

Here, the DL-based methods that regard the person Re-ID task as binary-class classification problems will be discussed [37, 42, 43]. In such methods, the deep model uses a pair of person images as the input and then outputs a similarity value that is used to determine if the paired images are of the same person [4]. The basic verification deep model architecture is shown in Figure 2.22. Several studies in the literature have employed the verification deep models to address the person Re-ID task.



**Figure 2.22** Basic architecture of standard verification deep model [4].

Li et al. [42] introduced the first study to use the verification model for dealing with the person Re-ID problem. They proposed a network architecture named filter pairing NN (FPNN). FPNN includes a patch matching layer, which compares 2 sets of features learned from 2 different input images, and from this, it calculates a set of displacement matrices. This layer is followed by a max-out grouping layer, where only the displacement matrices with the highest activations are passed onto the following layers. The subsequent layers are further Conv and max-pooling layers, followed by an FC layer, which results in the final feature descriptor for each of the input images. The final layer uses a SoftMax function for determining if the 2 input images depict

the same identity. It has a large learning capacity that allows the modeling of a mixture of complex geometric and photometric transforms. This network was extended by Ahmed et al. [37]. They proposed the learning of cross-input neighborhood difference features, which consisted of comparing the features from a particular region in one of the input images with the features that were obtained from neighboring locations in the second input image. The justification for this was to add robustness to the positional variation present between the 2 input images. The output of this layer produces an approximate relationship between the features extracted from the 2 input images. The authors also proposed an additional novel layer that summarizes these neighborhood difference features into a smaller feature descriptor that is passed through a further Conv and max-pooling layer. Finally, 2 FC layers and a SoftMax function are used to determine if the 2 input images depict the same identity. Then, based on these 2 studies above, in [43], Wu et al. proposed a PersonNet model for person Re-ID. The model made use of the patch matching layer that was proposed by Köstinger et al. [31] to capture the local relationship that was between the patches. Additionally, a deep architecture with smaller Conv filters helped to increase the architecture's depth. These verification models above have a drawback, in that their network depths are quite shallow, which provides little benefit when digging for the deep features that can discriminate the images. Moreover, the verification model must construct pairs of images as the input, and it can only use weak dataset labels [12], which reduces the training efficiency.

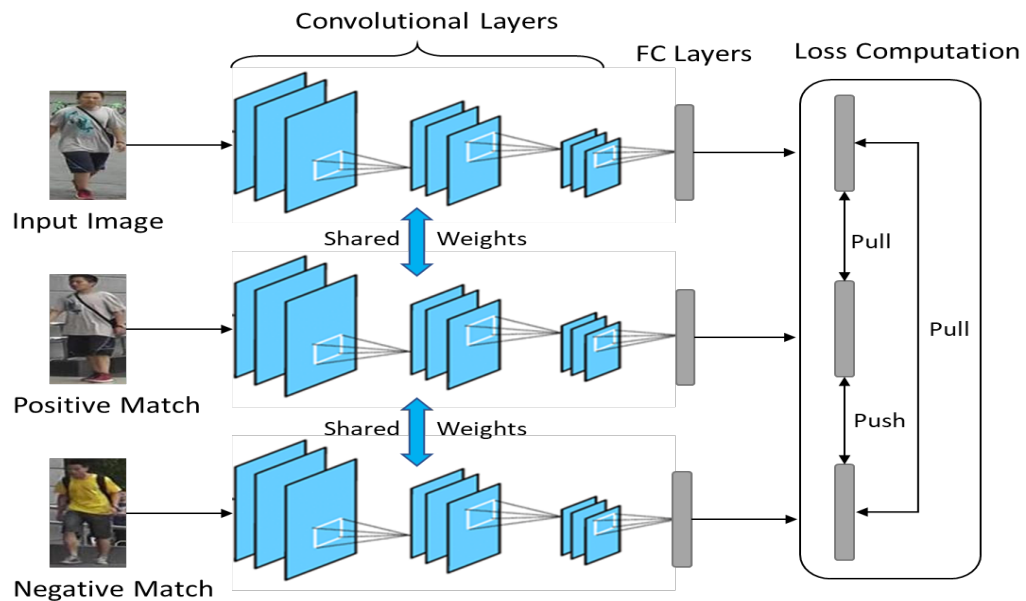
It is also possible to incorporate both the identification loss and verification loss within a single network architecture. These networks simultaneously learn features that are suitable for both the multi-class recognition of a given person identity while also being able to determine if a pair of input images belong to the same person. It should be mentioned that combining the identification and verification in a single model has resulted in promising results for person Re-ID. Chen et al. [94] proposed a network architecture that utilizes a combination of 2 losses, which include identity classification loss, which was preferred because the model requires pre-trained on the auxiliary ImageNet dataset to perform the object classification task. The other is verification loss, which tries to learn feature representation for a matching person. The authors proposed a loss-specific dropout unit, which ensures that the same dropout

map is applied to both feature maps extracted from a given image pair. Following the loss-specific dropout unit, the feature map pair that resulted was passed to a verification loss layer, and each of the feature maps of the pair is individually passed to an identification loss layer. Soon after, Zheng et al. [134] proposed a verification-identification network that combines the verification and identification losses to learn more discriminative pedestrian descriptors for person Re-ID tasks. Given an input pair of images, each image is passed through a separate branch of a network. Both branches are pre-trained CNN models that share weights and predict the input 2 image pair identity labels simultaneously. Each branch has an ID classification loss layer, which uses a SoftMax function to predict the identity of the input image. The final feature maps of each branch are then taken and compared using their proposed square layer, which takes 2 feature maps, subtracts one feature map from the other, and performs a square operation element-wisely. Finally, the output feature map is used as input to the verification loss layer, which uses a SoftMax loss function to predict whether the two input images represent the same identity or not. Afterward, Zhong et al. [47] proposed an architecture that utilizes both verification loss and identification loss. First, each image of an image pair is passed through a feature extraction network named the feature aggregation network (FAN), which was built based on a ResNet-50 [82] network architecture. FAN extracts feature maps from different layers in the network and provide an element-wise sum to produce the fused feature map. Next, for the verification branch, the feature maps are passed to a recurrent comparative network (RCN), which compares the appearance of images that form an image pair using a combination of attention and recurrent networks. The attention component weights discriminatively significant regions of the feature map, while the recurrent networks aggregate the discriminatively significant regions of the image pair. The RCN output is then used as input to the verification loss layer, computing whether the 2 input images are of the same person. For the identification branch, the feature maps extracted from the FAN are pooled using a global average pooling operation and then passed to an FC layer with SoftMax loss to predict the probability that the image belongs to each class.



### 2.5.3 Methods Based on the Distance-Metric-Based Model

Distance metric-based deep model attempts to decrease the distances that are between the same person images to a size that is as small as possible, while at the same time increasing the distances that are between different person images to a size that is as large as possible. The metric approach that is most commonly used is the triplet-based deep architecture. Methods that utilize triplet loss take 3 images as the input, namely a probe image, one that has the same identity as the probe image, and one that has a different identity than that of the probe image. The network then pulls the output of the input image so that it is closer to the positive match image by minimizing the distance that is between them in the output feature space. Next, the network then pushes the input image and the positive match image away from the negative match image by maximizing the distance that is between the different identities in the output feature space. The basic deep architecture of the triplet model is shown in Figure 2.23.



**Figure 2.23** Triplet deep model architecture.

Training a network using triplet loss is beneficial within the field of Re-ID. Therefore, multiple methods have employed this approach. Ding et al. [135] were the first to adopt the triplet model to address the person Re-ID problem. They proposed a scalable deep feature learning model through the use of relative distance comparison. In their model,

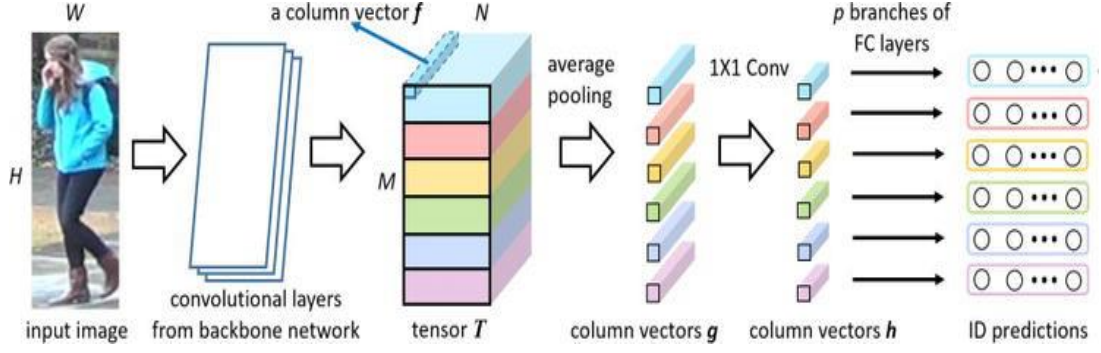
the CNN network had to be trained by a set of triplets to be able to produce the features that could satisfy the relative distance constraints that were organized by that same triplet set. To cope with the fact that the number of triplets grew cubically, these researchers presented a very effective triplet generation scheme as well as an extended network propagation algorithm that was able to train the network iteratively and efficiently. This learning algorithm can ensure that the general computation load will mostly depend on the number of training images instead of the number of triplets. Soon afterward, Cheng et al. [39] proposed an improved triplet method for person Re-ID, where both the global image features and the local body part image features are integrated and then passed to a triplet loss function. Furthermore, they believed that the typical triplet loss training strategy was not able to specify how close the deep features that were generated from the same person images should be, which led to a situation wherein the image that was ascribed to the same person might have an intra-class distance that is very large. Consequently, the authors proposed a further constraint to the triplet loss function, ensuring that the distance between the probe image and positive input must be below a certain threshold. This improved triplet loss function is capable of pushing the different person images further apart from each other and then pull the same person images closer, simultaneously, using the newly-learned deep feature space. It can improve the matching rates by up to 4%. Hermans et al. [136] proposed a batch hard triplet loss that uses a hard mining strategy in a mini-batch. They randomly selected  $P$  person identities and then sampling  $K$  images from each  $P$  class. Hence, the mini-batch contains  $PK$  images. As a final step, they select the hardest negative and positive samples from one of the mini-batches and use that to form the triplet units. Zhao et al. [137] proposed constructing a feature descriptor using a concatenation of feature maps that were taken from many different layers within a network. This combines shallow layer information with deeper layer information, allowing different visual characteristics to be captured by each component, which makes up the final feature descriptor. These multilevel features are then used as part of a triplet loss training strategy. The drawback of the triplet-based deep model is that it only uses label information from the Re-ID datasets, which are weak [12]. In addition, it is necessary for the model that the triplet units are constructed as input, because this will reduce the efficiency during the training phase. Some of the

approaches have been used to optimize deep architecture via the combination of the triplet and the identification losses [138-140] or by the combination of the triplet and the verification losses [38, 141].

#### **2.5.4 Methods Based on the Part-Based Model**

The part-based deep model tries to learn the local feature representations from human body parts instead of learning the global full-body representations. Local information is given as a discriminative clue for use in pedestrian recognition. Hence, the exploitation of part-level features when performing the person image description is able to provide fine-grained information, and It can be considered useful for person Re-ID [142]. Local feature representations usually learn part- or region-aggregated features, which then makes them very robust against misalignment variations [57, 143]. Body parts can either be generated through the use of human pose estimation or roughly through horizontal division [144].

Several works in the literature have adopted the part-based strategy in an attempt to learn discriminative deep features for the person Re-ID task. Cheng et al. [39] made use of a global Conv layer to attain the global Conv features. Then, they divided the features into 4 equal and individual branches, so as to obtain the part-based deep features. As a final step, they integrated the global and the part-based feature vectors so that they could produce the final deep features. Like-wise, in [145], Li et al. also used the division strategy. However, they made use of multi-classification losses for optimizing their designed network. Sun et al. [57] developed a deep model that was named part-based Conv baseline (PCB), as shown in Figure 2.24, which can perform uniform partitions on the Conv layer to learn the part-level features. While using a single image as the input, this model is able to output a descriptor that consists of 6 part-level features. The refined part-pooling (RPP) approach was also proposed to re-assign the outliers of each part from the uniform partition to the parts that they are the closest to, which resulted in refined region partition as well as within-part consistency that was much improved. This model was able to achieve very competitive performance on some of the most popular public datasets.



**Figure 2.24** Structure of the PCB deep model [57].

Li et al. [41], in their study, designed a context-aware multi-scale network for exploiting the features of local and global body parts. In practical terms, rather than using the predefined division parts, they proposed the use of spatial transform networks for localizing the latent person parts. As a final step, they then integrated the full body as well as the body parts to construct the final representation. The attention mechanism, which was used as a part-feature learning module, was used by Liu et al. [138], and Wang et al. [146] for enhancement of the discriminative ability of the learned deep feature. Li et al. [147] proposed a novel harmonious attention CNN (HACNN) for joint learning of person Re-ID attention selection and feature representations in an end-to-end fashion. Extensive comparative evaluations validated the superiority of this new HACNN model for person Re-ID over a wide range of state-of-the-art methods. Generally, the local visual cues were close to the visual habits of human beings and were also quite complementary to global information. Using this combination of the global and local features was a good choice for use in the person Re-ID. However, the part-based deep method does have various limitations, such as the fact that adding the part-based branches to the deep model might possibly increase the complexity of the model, which would then cause a reduction in the efficiency of the training. In addition, the attention-based deep models only take into consideration the region-level attention, which means that they ignore the pixel-level saliency [147]. These methods are ineffective when dealing with the small, labeled training datasets or noisy pedestrian images that have background clutter and are misaligned. Moreover, although the spatial contextual data are important elements for discriminative representations, very few of the methods that were listed above take into consideration the spatial context information that exists between the different part-based features.

### 2.5.5 Methods Based on the Generative Adversarial Network Model

The GAN-based person Re-ID methods aim to extend the existing training sets by generating more samples from the original dataset using GANs, rather than proposing new deep models. Generally, it is common knowledge that the performance of DL models relies on the amount of training data. In the current person Re-ID community, the number of training samples was still insufficient and lacked diversity. As an example, the average number of samples per person for large-scale Re-ID datasets, such as CUHK03, Market-1501, and DukeMTMC-reID, is, respectively, 9.6, 17.2, and 23.5 [4]. Using datasets of such large scale when training the deep model may lead to an overfitting issue. GANs have been highly successful in generating realistic images after being trained on sample images [128]. Therefore, the use of GANs for solving the person Re-ID problem has gradually become a research hotspot in recent years.

Zheng et al. [50] proposed the use of a DCGAN for the generation of unlabeled person image samples, as well as the adoption of a CNN sub-model to be used in feature representation learning. Provided that the person images that are generated do not have labels, by using the LSRO method, the model can conduct the training by combining the unlabeled DCGAN images and the real labeled images. The big advantage that can be gained from using LSRO is that it allows for dealing with many more training images, i.e., outliers, close to the real training images, within the sample space, and it also introduces a greater degree of variances, such as the color, lighting, and pose, which can regularize the model. They made use of the DCGAN in the generation of new pedestrian images that had new labels via the use of semi-supervised learning. They indicated which were the generated images using the LSRO label distribution. The new idea of their proposed approach was the improvement of the generalization capability of the model via the production of new samples that were given LSRO labels. Even still, the new images were much too blurry to meet the artificial benchmarks and were not able to directly add to the size of the data. Ainam et al. [148] proposed the exploitation of k-means clustering to create groups of similar persons in the training sets and the training of a DCGAN to generate unlabeled samples for each cluster. The authors introduced sparse label smoothing regularization (SLSR) for use in assigning a non-uniform label distribution to the unlabeled DCGAN-generated

images and defining a regularized loss function for the training. As a next step, the generated images were then assigned smooth labels based on the distribution of their cluster labels, which were used for the DCGAN stream. Thus, SLSR was able to deal with the problem of over-smoothness that is commonly found in the current regularization methods. However, the new images are still too blurry, and it is not possible to add to the data size directly.

For the person Re-ID dataset, the image styles of different cameras may differ; thus, resulting in the pedestrian images that are captured by cameras that do not overlap and experience the significant changes in their background and appearance. To deal with this problem, in [51], Zhong et al. introduced a camera-style adaption model that can be used in the adjustment of the CNN training. In particular, they utilized CycleGAN [115] for transferring the style of the images that were captured by one of the cameras to another. When given a training sample from a specific camera, the model was able to produce new images by using the styles of the other cameras. In addition, for alleviating the noise in the generated images that were caused by CycleGAN, they introduced label smoothing regularization (LSR) to these new images. Because there exist many data sets for the person Re-ID, this raises a problem called the domain gap problem. This problem commonly exists between different datasets and is caused by the variations in the camera settings and capture conditions. The domain gap would cause there to be a serious performance drop when the model was trained on one specific dataset, yet it tested on another. To counter the effects of the significant domain gap between the different Re-ID datasets, Wei et al. [52] proposed a person transfer GAN (PTGAN) model, which performs image-to-image translation from one dataset domain to another to bridge the domain gap. PTGAN fills the gap in the domain by transferring some of the people in the C dataset into the D dataset. After this transaction, the transferred will images maintain their IDs and styles that are similar, such as backgrounds, lighting, and so on, with dataset D. This can theoretically allow a network to more effectively use the training set of data set C to train a network that will be used for evaluating on data set D, leading to a significantly larger training set. In the same direction, Zhou et al. [55] proposed the multi-camera transfer GAN (CTGAN). CTGAN was able to convert the source dataset images to the multi-camera styles of the target dataset. These converted images were able to retain the source

dataset labels, and they were then used as the data for training of the target dataset model. For improving the accuracy of the training, the mixed selective Conv descriptor aggregation (MSCDA) method was used for the reduction of the effect that the noise signal that was caused by the generation of the new image.

The pose of the human body has a significant role in the changes that appear in the appearance of a person. Here, the pose can be viewed as a combination of the body and viewpoint configurations. In addition, it also causes self-occlusion. Qian et al. [53] focused on alleviating the impact of pose variation by proposing a pose-normalization GAN (PN-GAN) model. When provided with a person image, the model then chooses a desirable pose for the generation of a composited image of that same ID, in which the initial pose has been replaced with the more desirable one. These pose-normalized images, as well as the original images, are then used for training of the Re-ID model, which can now produce 2 sets of features, respectively. As a final step, the 2 kinds of features are fused together to form the final descriptor. This process increases the size of the training set and allows the network to extract features more representative of the individual, which are unaffected by pose variation. However, these produced images will be of comparatively poor quality, which will result in the presence of fetching noise in the Re-ID model. In [54], Yixiao et al. proposed yet another GAN model that was named feature-distilling GAN (FD-GAN), which made use of a pose encoder, as well as its generator, to be able to produce new person images that had different target poses but were of the same identity. Recently, Zheng et al. [56] proposed a joint discriminative and generative learning framework called DG-GAN, in which image generation and Re-ID learning were integrated, end-to-end, into a unified model that was able to generate high-quality images through switching of the structures and appearances between the 2 images and the online feed-generated images for the Re-ID learning.

Overall, GAN deep models were used to perform data augmentation in an attempt to deal with some of the person Re-ID limitations and also boost the generalization ability of the person Re-ID model. Despite this, the problem that continued to limit their use was that the generated synthesized images still had relatively poor quality, as a result of the noise that was carried through into the training dataset. Therefore, there is still

a need to develop a GAN model that will be effective in creating new, high-quality, and highly diverse images. Generated data such as this would greatly improve the accuracy obtained when using person Re-ID models in real life. Most recently, quite impressive results were obtained using StyleGAN [6], which was originally an open-source project conducted by NVidia for the generation of high-quality photo-realistic human faces. This was accomplished via gaining control of the style of the created image at different levels of detail by making use of the different degrees of noise and vector styles. StyleGAN [6], after its results with face generation, was used in the current research for the generation of high-quality person images using the person Re-ID dataset that already exists. Then, these newly-generated images were used to increase the size of the training sets to improve the learning capacity of the CNN model.

## **2.6 Person Re-identification Benchmark Datasets and Evaluation Metrics**

To conduct an evaluation of the performance provided by the person Re-ID methods, it was necessary to construct consistent datasets that would aid the researchers in evaluating and comparing their results, and also identify any possible, yet unforeseen limitations, so as to contribute to the enhancement of the performances. It is very important to have an available Re-ID datasets with characteristics such as the illumination variation, occlusions, pose variations, cluttered background, and overlapped bodies to reach a reliable Re-ID rate on this task [4]. It is also important to have appropriate performance analysis metrics to evaluate the Re-ID performance of the proposed person Re-ID methods. The following sub-sections summarize both the datasets and the evaluation metrics that are commonly used to evaluate the person Re-ID task.

### **2.6.1 Person Re-identification Benchmark Datasets**

For the proposed techniques and algorithms, validation via the use of various datasets is vital to be able to guarantee that the parameters or approaches do not contain bias. Therefore, several public datasets have been prepared specifically for the evaluation of the person Re-ID task. These datasets differ from each other in the number of



images, identities, cameras, and image types. And given that the process of collection of the data to be used when training deep models are work that takes a lot of time, and to make it more convenient, the current research will conduct an evaluation of public datasets. Sadly, not all comprise enough data for training DL models, as it is normally necessary to use enormous datasets for training, that have characteristics such as inter- and intra-class variations. Table 2.3 gives a summary of popular benchmark person Re-ID datasets.

These datasets were collected from many different public areas. It can be seen in Table 2.3 that the dataset size was significantly increased. Years ago, datasets were actually quite small. Recently, however, datasets have begun to comprise a larger array of images. Moreover, person images in most of the datasets have usually been marked using hand-crafted boundary boxes. Presently, as a result of improvements in person detection technology, automatic detection and tracking algorithms, including NNs and the deformable part model (DPM) [149] are often used to draw the bounding boxes automatically. In this research,, Market-1501 [22], DukeMTMC-reID [50], and MSMT17 [52] were chosen to be used because they possess a fairly large volume and they are popular choices among researchers who are studying person Re-ID tasks to evaluate the DL models [4]. These 3 datasets were used in this thesis for consistent comparative studies of the proposed algorithm. The sections below present the characteristics and details of these datasets.

- Market-1501 Dataset:** Market-1501 [22] is among the largest person Re-ID datasets available for use today. It comprises 32,668 images that reflect 1501 identities. Capturing of the person images was performed using 6 cameras, of which 5 had very high-resolution, while 1 had low-resolution, and they were positioned in front of a supermarket at the Tsinghua University campus. The dataset contains a lot of instances of occlusion, particularly by bicycles and bags. Many of the images are high quality; however, many of the images also suffer from high levels of blur. The DPM [149] was used to recognize and crop the images. The images were scaled to  $128 \times 64$  pixels and suffered from significant variations in pose, illumination, and background. Moreover, often, the cropping process can remove parts of the body of a person from the image. The Market-1501 dataset

comprises 3 parts, consisting of the training, gallery, and query. The training consists of 12,936 images of 751 identities, the gallery consists of 19,732 images of 750 identities, and the query consists of 3368 images of the same 750 gallery identities. Examples of the images from the Market-1501 dataset are shown in Figure 2.25.

**Table 2.3** Summary of commonly used datasets for person Re-ID.

Dataset	Year released	Number of cameras	Number of identities	Number of images	Detector	Image size	Evaluation
VIPeR [30]	2007	2	632	1264	Hand	128×48	CMC
ETHZ [150]	2007	1	148	8580	Hand	Vary	CMC
PRID2011 [151]	2011	2	934	24,541	Hand	128×64	CMC
CUHK03 [42]	2014	10	1467	13,164	Hand/DPM	Vary	CMC
i-LIDS [152]	2014	2	300	42,495	Hand	Vary	CMC
Market1501 [22]	2015	6	1,501	32,668	Hand/DPM	128×64	CMC/mAP
DukeMTMC-reID [50]	2017	8	1,812	36,411	Hand	Vary	CMC/mAP
Airport [153]	2017	6	9,651	39,902	ACF	128×64	CMC/mAP
MSMT17 [52]	2018	15	4101	126,441	Faster RCNN	Vary	CMC/mAP



**Figure 2.25** Examples of the Market-1501 dataset images. Each of the columns represents one identity.

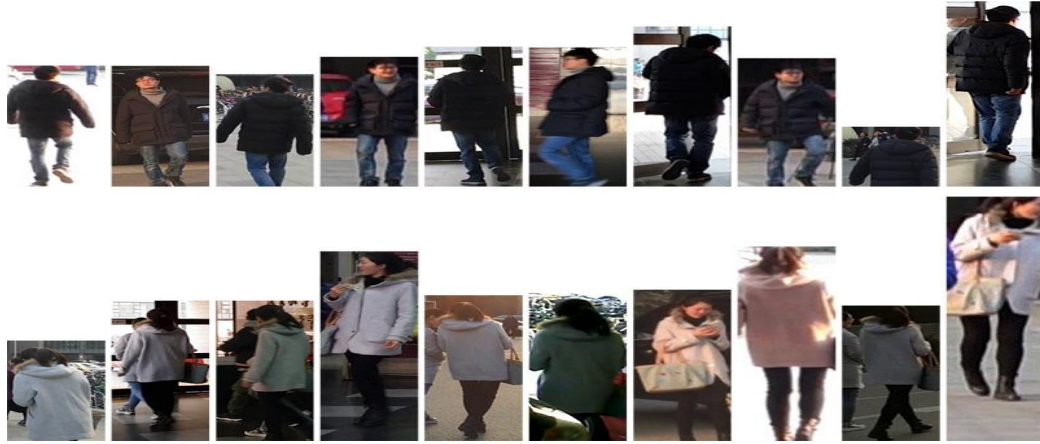
- DukeMTMC-reID Dataset:** DukeMTMC-reID [50] is a subset of the DukeMTMC dataset. This dataset was manually annotated and recorded in an outside environment on the campus of Duke University by using 8 non-overlapping and synchronized cameras. It comprises a total of 36,411 annotated bounding boxes of 1404 identities. Similar to the Market-1501 dataset, this also has three parts, consisting of the training, gallery, and query. The training consists of 16,522 images of 702 identities, the gallery consists of 17,661 images of 1110 identities, and the query consists of 2,228 images of 702 identities. All of the DukeMTMC-reID dataset images vary in size and possess strong variations in pose, illumination, and background. Examples of the images from the DukeMTMC-reID dataset are exhibited in Figure 2.26.



**Figure 2.26** Examples of the DukeMTMC-reID dataset images. Each of the columns represents one identity.

- MSMT17 Dataset:** MSMT17 [52] is a new large-scale person Re-ID dataset. It was collected on a university campus by using 12 cameras that were located outside and 3 cameras that were located inside. The collection procedure for the images took place over a period of 4 days in the same month under different weather conditions. For each of the 4 days, 3 one-hour videos were chosen from the morning, noon, and the afternoon. Faster RCNN was used in the pedestrian detection and bounding box annotation. MSMT17 dataset contains 126,441 images of 4101 identities. This also has three parts, consisting of the training, gallery, and query. The training consists of 32,621 images of 1041 identities, the gallery consists of 82,161 images of 3060 identities, and the query consists of 11,659 images of 3060 identities. All of the MSMT17 dataset images vary in size.

MSMT17 is the largest Re-ID dataset that has been created thus far. Its design is similar to that of Market-1501; however, it contains scenarios that are a lot more complicated. Examples of the images from the MSMT17 dataset are exhibited in Figure 2.27.



**Figure 2.27** Examples of the images from the MSMT17 dataset. Each row represents a single identity.

### 2.6.2 Person Re-identification Evaluation Metrics

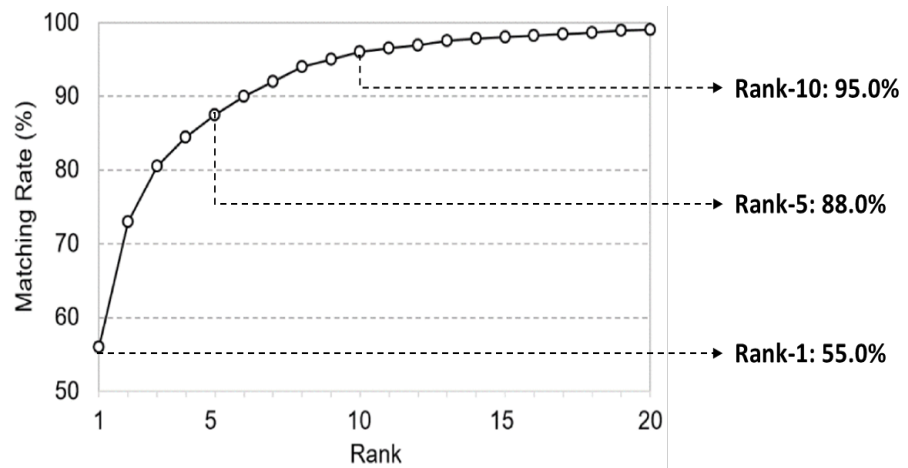
Currently, the 2 most commonly used standard metrics for person Re-ID evaluation are the cumulative match characteristic (CMC) curve [154] and mean average precision (mAP) [155]. Therefore, these 2 metrics were adopted in this research to evaluate the performance of the model that was proposed herein for person Re-ID to be able to provide a good comparison with the previous approaches in the literature.

- **Cumulative Match Characteristic (CMC):** This is used quite extensively for the person Re-ID task to conduct the measurement of the performance of various algorithms. Its use is common because the person Re-ID task is posed intuitively as a ranking problem, in which the ranking of the images in the gallery is performed based on their similarities to or distances from the query image, and then the matching accuracies for each of the ranks are computed. This is able to provide a rank for each of the images in the gallery with regard to the query. The probability that a correct match would be ranked as equal to or less than a specific value is plotted against the size of the set of gallery photos [2]. With a CMC curve, the accumulative number of queries that are matched correctly is given on the basis of

the ranking list from which they were re-identified. In a situation where the number of correctly reidentified queries in rank  $i$  is  $q(i)$ , the CMC value for that rank is then defined as given below:

$$CMC(i) = \sum_{r=1}^i q(r) \quad (2.19)$$

Here,  $r$  is the rank index. On the CMC curve, the first point is known as the first Re-ID rate, which is Rank-1. The matching rate for Rank-1 is the matching accuracy of only the first rank, which is a priority concern for Re-ID operators. A high rate of matching for Rank-1 will significantly increase the efficiency of the person Re-ID model in real-world applications. The CMC evaluation metric has an advantage in that it is able to not only compute Rank-1, but it is also able to determine the correctly matched queries in the other top ranks. Comparisons of the CMC curves for the most recent deep person Re-ID methods were simplified to be able to compare the retrieval rates for Rank-1, -5, -10. Figure 2.28 illustrates the conversion between the CMC curve to Rank-1, -5, and -10. Rank-1, -5, and -10 can be thought of as a more simple version of the CMC curve. To state it simply, CMC is only an accurate evaluation method if 1 ground truth exists for each of the queries. However, if the gallery consists of multiple ground truths, the mAP can be used in the evaluation of the overall performance.



**Figure 2.28** CMC curve to Rank 1, 5, 10 conversions.

- **mean Average Precision (mAP):** The mAP is the standard single-number measure for comparing search algorithms. It was introduced to allow for an evaluation of the performance of person Re-ID algorithms to be used to finding all of the matched samples in the case where there are multiple ground truths in the gallery [22]. For each of the query images, the AP is calculated as the area under its precision-recall curve. Given a query image, the AP can be defined as given in the equation below:

$$AP = \frac{\sum_{k=1}^n P(k) \times G(k)}{N_{gt}} \quad (2.20)$$

Here,  $n$  is the number of test tracks,  $P(k)$  is the precision at cut-off  $k$  in the list of ranks,  $G(k)$  is equal to 1 when the  $k^{th}$  match is true; otherwise, it will be equal to 0, and  $N_{gt}$  is the number of ground truths. Hence, the mAP is the mean value of the APs of all of the queries [22]. The mAP can be calculated over all of the queries as given in the equation below:

$$mAP = \frac{\sum_{q=1}^Q AP(q)}{Q} \quad (2.21)$$

Here,  $Q$  denotes the number of queries. For datasets with multiple images that are matched for each of the query images, the mAPs are evaluated all together with the CMC curve to make a better performance evaluation. The performances of the current person Re-ID methods are usually examined by combining the CMC curve for retrieval precision evaluation and mAP for recall evaluation.

## 2.7 Concluding Summary

In this chapter, a comprehensive review was conducted of the relevant literature, which provides the basis of the person Re-ID task, is presented, which includes a summary of the research that has been conducted with regards to the person Re-ID task, beginning with the early hand-crafted feature engineering up to the recent DL methods.

Also, it introduces the DL techniques utilized for person Re-ID, including CNNs, transfer learning, and GANs. Furthermore, it provides a taxonomy of the DL-based person Re-ID methods based on the type of deep model. Finally, it covers the main public benchmark datasets on which person Re-ID methods are evaluated, and evaluation metrics used to evaluate these methods.

Overall, the use of person Re-ID that is based on DL approaches remains within the stages of development. Currently, the DL methods that have been introduced have mostly been based on improving the basic network models that they currently comprise. As the characteristics of per-person images, establishing a new deep network model has become a main focus of research. Undoubtedly, better results have been gained by these methods, as it is a waste of time to constantly make adjustments in the parameters during the training process, and over-fitting can easily occur when the structure of the network becomes deep. Besides, an improvement in the computation costs and saving time as much time as possible are the keys to practicality. Because the number of training samples that are used has an effect on the performance of the DL model, the establishment of only a standard large-scale dataset will ensure that the trained models maintain good generalization ability in such a complex environment. Simultaneously, as a recognition application that has multiple cameras and crossed views, person Re-ID can be applied in real-life situations. However, the methods that exist today are unable to ensure both rapidity and recognition accuracy at the same time. The GAN-based deep model was used as a technique that could be used for data augmentation, as it is able to handle, to a certain extent, the limitations in the person Re-ID problems that already exist. However, most of the person images that are synthesized by the currently known GAN-based models have a quality that is quite low. Hence, this results in the existence of noise in the original datasets. Therefore, it is necessary for the GAN-based data augmentation of the future to consider producing new, highly varied, and high-quality persons images from the original dataset and utilizing these newly generated training data to improve the accuracy of the person Re-ID deep models.

StyleGAN was used for the first time in the current research, at least, as best as is known, in the generation of human images with very high quality that was similar to

the existing person Re-ID datasets. These newly-generated images were then used for the enlargement of the training sets via the introduction of a considerably more vast variation, with regards to background, color, illumination, and poses, to provide regularized and improved accuracy and robustness of person Re-ID models. The LSRO method was used to integrate the StyleGAN-generated images into the original labeled training images, which it did after it assigned them each a uniform label distribution and defined a regularized loss function to train the classification CNN model, which was utilized to conduct the feature extraction task. Cosine distance was then used in the measurement of the distances of the features. Three popular and relatively large volume person Re-ID datasets were selected, namely Market-1501, DukeMTMC-reID, and MSMT17, as the datasets used in the evaluation. The evaluation was based on the CMC curve and mAP.



# CHAPTER 3

## METHODOLOGY

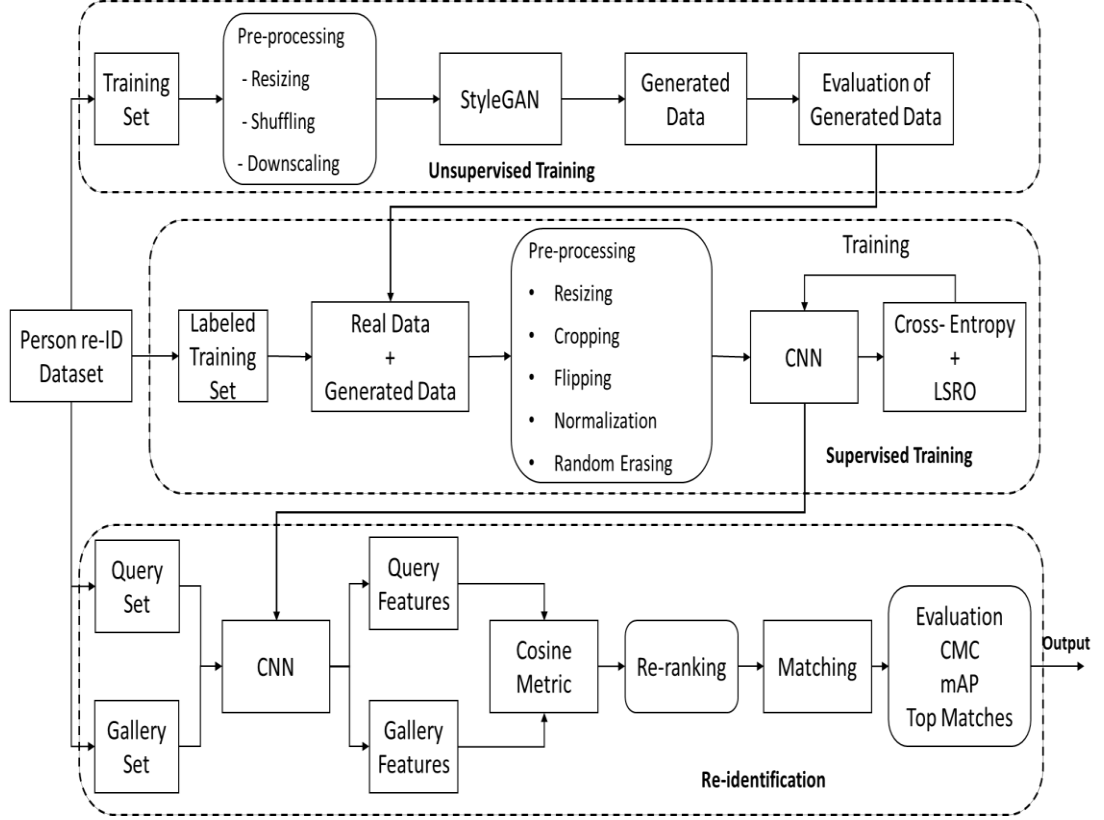
The biggest problem in the DL-based person Re-ID field is that the annotated data is insufficient to build an efficient deep model with good discrimination and generalization ability. Despite the fact that, in recent years, researchers have proposed many methods to solve this problem, by generating additional images by using GANs, the quality and diversity of these generated images have remained quite low as a result of the noise that is brought to the training dataset. Moreover, there is no effective way to label these newly-generated images and add them to the original training data. Therefore, there is still a need to find a GAN model that can generate new, highly-diverse, and high-quality persons images, and moreover, to find an effective way to add these images to person Re-ID datasets. This chapter presents a new method for person Re-ID to overcome the aforementioned problems and the rechallenges presented in the first chapter. This proposed method is called the StyleGAN-LSRO method. In this method, the StyleGAN, a recent state-of-the-art GAN architecture for high-resolution synthetic image generation, is made use of in the generation of person images that are of high quality from the person Re-ID dataset that is already in existence. Then these newly-generated images are utilized in the enlargement of the training sets via the introduction of a considerably more vast variation, with regards to background, color, illumination, and poses, to provide regularized and improved accuracy and robustness of person Re-ID models. The LSRO method was used, which assigned a uniform label distribution to the generated unlabeled images and defined a regularized loss function for the training. In addition, a baseline model that was based on pre-trained CNNs was developed for feature learning, and the Cosine distance metric was used for the similarity matching between features. As far as is presently known, this attempt was the first to make use of StyleGAN as a generative model for the person Re-ID task. In addition, it will be aimed to prove that this new method is much better than the methods that already exist. In the following sections of this chapter, the proposed method will be discussed in detail.

### 3.1 The Proposed StyleGAN-LSRO Method

Figure 3.1 shows the basics of the framework for the StyleGAN-LSRO method that is proposed in this research for person Re-ID. The person Re-ID procedure using the proposed StyleGAN-LSRO method was defined in 3 phases as follows:

1. Unsupervised training of the StyleGAN model using the pre-processed original person Re-ID datasets to generate images of high quality in the domain of these datasets. Then, the trained generator of the StyleGAN model was used in the generation of the unlabeled synthetic samples, which were for the enlargement of the training sets of the person Re-ID datasets.
2. Supervised training of the baseline CNN model to learn the discriminative features that are required for it to recognize the identity of a person using the pre-processed new training set, which contained the generated data in conjunction with the original dataset. The LSRO algorithm was then used to integrate the generated images into original labeled training images during this training process via their assignment of a uniform label distribution in addition to the definition of the loss function. This process for the training was treated as a multi-class classification process in accordance with the number of categories in the real training set.
3. In the final phase, the trained baseline CNN model was exploited to extract the discriminative features of the query and gallery images. The Cosine distance metric was used to calculate the similarity score between them, to rank all of the gallery images with respect to each query image. RR was adopted to optimize the ranking results. Moreover, in this phase, the Re-ID performance was qualitatively and quantitatively evaluated.

In the subsequent sections, the design and implementation process for each phase will be explained in detail. The process of developing the proposed method for person Re-ID was begun by building a baseline model based on pre-trained CNNs and defining its training strategy. Then, it will move onto developing the StyleGAN model and defining its training strategy to generate high-quality images. Finally, it will go back to training the same baseline model on the new dataset that containing generated and real images, using the LSRO loss function.



**Figure 3.1** Overall block diagram of the StyleGAN-LSRO method.

### 3.2 Baseline Model Building for Person Re-Identification

In this section, the details for building the DL-based baseline model for person Re-ID will be given. The approach that was used herein for building such a baseline model followed the IDE approach by Zheng et al. [12, 127], which takes advantage of the deep CNN models that have previously been trained on ImageNet [85] and fine-tuned them using person Re-ID datasets. IDE considers the training process for person Re-ID tasks as a multi-class classification mission. The contributions of the current research approach are 2-fold. First, designing the baseline deep model architecture based on the pre-trained CNN models for the object recognition problem by modifying the architecture of the CNN models and using a transfer learning approach to re-train them with an available person Re-ID dataset. In performing transfer learning on the off-the-shelf deep representations, it was able to attain good results in the person Re-ID problem. The features that were obtained from the deep models also possessed discriminative properties that were good. Second, exploiting two optimization

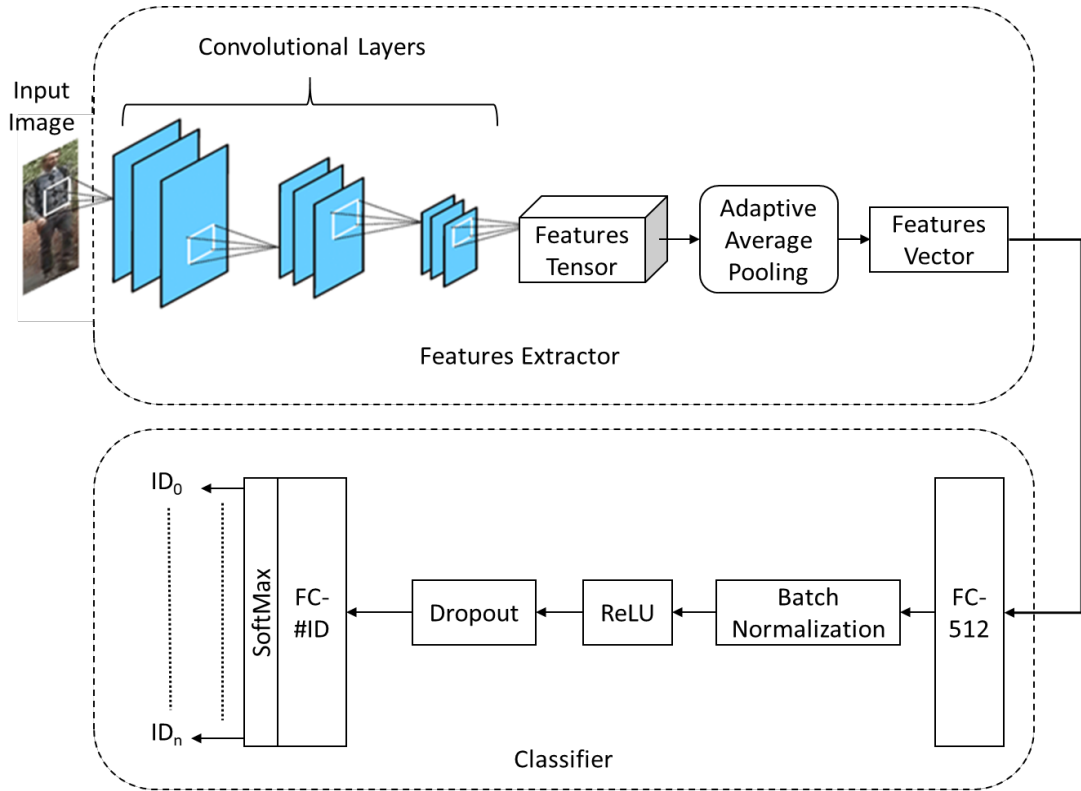
techniques, namely RE [77] and RR [156] for improving the performance of the baseline model proposed herein.

### 3.2.1 Baseline Model Construction

The first step was selecting the pre-trained CNN model as a backbone model to be used in building the baseline model. Since it is relatively impossible to construct a new model from scratch. According to section 2.2.6, ResNet-50 [82] and DenseNet-121 [83] are well-known CNN models in the image recognition domain and are widely used in the task of person Re-ID. Therefore, they were selected to build the baseline model in this research in order to ensure a fair comparison with other methods. These models have been previously trained in recognition of 1000 generic object categories, which are part of the hierarchy of ImageNet. This means that its learned features and weights are part of a broad domain. Moreover, its classification function is targeted so as to minimize the error in that domain. The network was then optimized again in a more specific domain to minimize the error, and its classification function was also replaced. The features and parameters of the network were transferred to the specific domain from the broad one.

To build the baseline model architecture, some crucial modifications to the architecture of the pre-trained model were performed, which made the proposed model able to significantly surpass the original one. In the primary pre-trained model, first, the last 1000-dimensional classification layer was removed. Then, the last pooling layer in the primary model was replaced using a new adaptive average pooling layer that was used as a flattening layer to flatten the features tensor that output from the last Conv layer. Next, a new and custom classifier, which comprised 2 FC layers with random values, was added to the training by using the new data and performing the new task. The first FC layer possessed an output of 512 dimensions and was attached to BatchNorm, ReLU, and also Dropout. The second FC layer possessed an output of N dimensions. Here, N is the number of the pre-defined classes that are in the original training set, consisting of 751 classes for Market-1501, 702 classes for DukeMTMC-reID, and 1041 classes for MSMT17, and it was attached to the SoftMax function, which was used in the prediction of the ID of the input. Each of the numerical values in the output is a representation of the probability that the image being used as the input belongs to

that class, and the one that is maximum will be chosen as the prediction. The overview of the proposed baseline model architecture is presented in Figure 3.2. The model comprises 2 main components, which are the feature extractor and classifier. The feature extractor is similar to that in the original pre-trained model. It is initialized with its weights, while the classifier part includes the newly added FC layers and is initialized with random values.



**Figure 3.2** The proposed baseline model architecture.

According to [79], BatchNorm aid in the current model converging faster and being more stable while it is training. Even though BatchNorm can provide a bit of a regularization effect, it is necessary to use Dropout [80] because it helps to attain a better regularization effect and also avoid over-fitting. It should be mentioned here that the Dropout probability is a hyper-parameter, which can be tuned to gain further improvement in the performance. For the new FC layer's activation function, Leaky ReLU was used instead of the original ReLU. This was because ReLU can easily die if the ReLU activation function's input is continually negative, resulting in an output that is 0, and its gradient will also be 0. Under these circumstances, the error signal

that was propagated from the last layers will be multiplied by the same 0. Therefore, no error signal can pass onto the earlier layers. Contrarily, if leaky ReLU is used, the model can appropriately map the negative input and still maintain the advantage of ReLU, which will not saturate and is able to avoid a vanishing gradient.

### 3.2.2 Baseline Model Training

In this section, the baseline model's training mechanism and the hyperparameters setting are explained. For the baseline model training to perform the person Re-ID task, the training process is simply regarded as a classification task, and it is conducted with the utilizing of person Re-ID datasets. In this research, 3 large-scale person Re-ID datasets were selected for this purpose, namely Market-1501 [22], DukeMTMC-reID [50], and MSMT17 [52]. To simplify the training process, every person's identity was considered as a separate class. In addition, the first image in each of the identities was taken as a validation set, and the rest of the samples were regarded as a training set. To train the baseline model, the mini-batch SGD with momentum [157] was used as an optimization scheme, and the chosen objective function was the cross-entropy loss, which is also called the log loss. This is normally used in the measurement of performance classification models, in which they have an output that is a set of probability values that range from 0 to 1. Using one single image as the input,  $N$  values will be outputted by the model ( $N$  is the number of classes), representing the probability that the image that was used for the input image belongs to each of the classes. The maximum value is then chosen as the predicted label, and calculation of the cross-entropy loss is performed using the ground truth and predicted label. The cross-entropy loss is calculated according to Equation. (2.4). The mini-batch SGD is an iterative method that is used in the optimization of the objective function, that is, the cross-entropy loss. In each of the iterations, calculation of the cross-entropy loss is performed based on the ground truths and predicted labels of each of the images in the mini-batch. As a next step, the loss is backpropagated via calculation of the loss with regards to the weight in each of the layers. As a final step, updating of the weights is performed according to Equation (2.5).

Usually, deep NN training is time-consuming, since there are many hyperparameters to tune, such as the mini-batch size, learning rate, and the number of training epochs.

Therefore, most hyperparameters used later in the experiments were selected with experiences and guidance from the literature [57, 147, 158, 159]. Selecting an appropriate learning rate has difficulties because, if the one that is chosen is too small, this will result in a slow convergence process. Simultaneously, if the one that is chosen is too large, it will result in fluctuation of the objective function at around minimum, or possibly, even diverge. Because the used models were previously trained on ImageNet, the weights of these previously trained parts were close to optimal, and thus could be updated using a small learning rate. At the same time, the 2 added layers were updated with a larger learning rate.

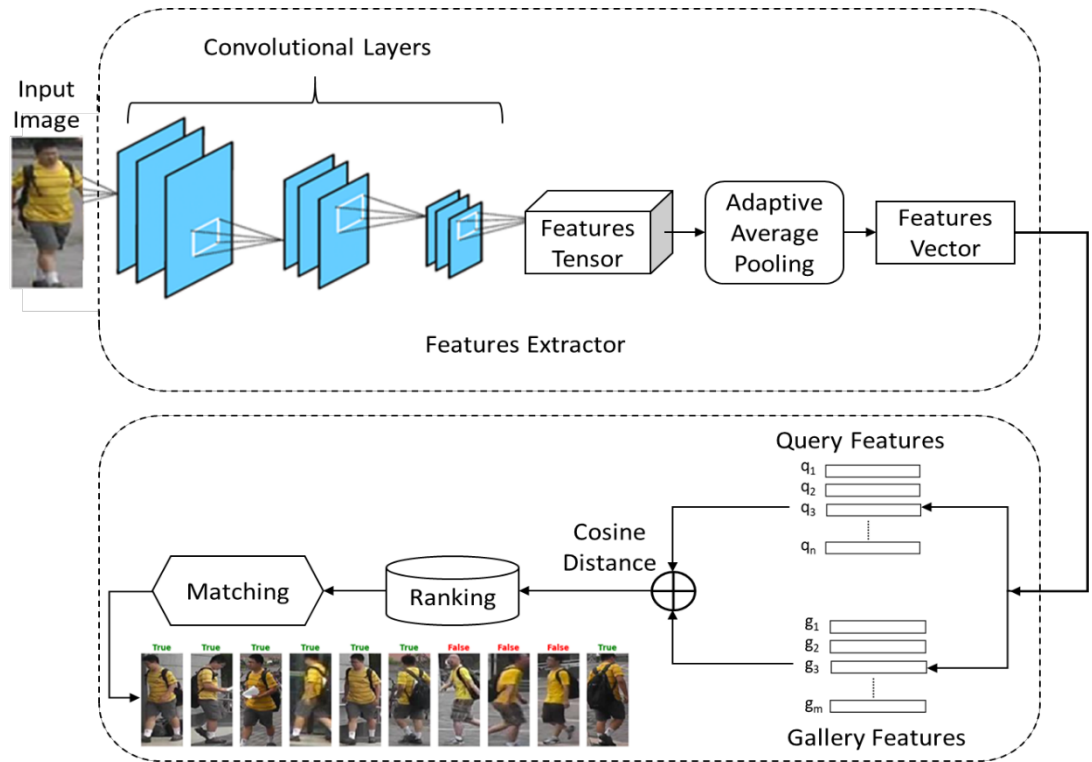
### **3.2.3 Features Extraction and Distance Metric Learning**

Following the training, the baseline model was ready to perform the extraction of the discriminative features of the images in the query and gallery for the person Re-ID task. The model's Conv layers can be considered as a feature extractor. When the image input into the model for extraction of its features, the convolution operation is first performed in each of the Conv layers. As a next step, to make the size of the image smaller, the pooling layer is used to eliminate any redundant features that are present, while keeping the useful ones. Hence, these layers are a fundamental part of the feature extraction process. To use the trained baseline model for feature extraction, the last 2 added layers (classifier layers) are removed from the baseline model to make use of the output that came from the adaptive average pooling layer as the feature representation of the input image. The baseline model built based on the ResNet-50 as a backbone outputs a 2048-dimensional vector as the input image's feature representation, while it outputs a 1024-dimensional vector when built using the DenseNet-121 as a backbone.

Once the features of all of the query and gallery images are extracted, they are saved to a file and prepared for the ranking process. The ranking process aims to determine the similarity scores of a query's feature vector against all of the gallery features. Then the calculation of the similarity scores is performed using the distance metric and is ranked when a smaller distance causes a higher rank. The processes of feature extraction and distance metric learning are summarized in Figure 3.3. The distance metric that was used in this research was cosine distance, which is able to provide the

angular cosine distance that is between vectors  $X$  and  $Y$ , which can be calculated as is shown below:

$$\cos(\theta) = \frac{XY}{|X||Y|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (3.1)$$



**Figure 3.3** Features extraction and distance metric learning.

### 3.2.4 Baseline Model Optimization

To optimize the proposed baseline model's performance and achieve good results in the person Re-ID task, there have been many optimization techniques proposed in recent years. In this research, 2 techniques were selected to optimize the proposed baseline model, namely RE [77] and RR [156], as their implementation and combination with the baseline model is uncomplicated, and no additional parameters need to be learned. The RE is utilized as a data augmentation method to produce some



occlusions in the training data, while the RR is conducted to improve the Re-ID performance further. In the following, the two optimization techniques are explained in detail.

- **Random Erasing:** Occlusion is a common occurrence in surveillance systems because the real world is very complex. Therefore, a person Re-ID system must always deal with a certain level of occlusion in the person images. A model that is robust model will be able to achieve high accuracy even if some areas of the images have been occluded. The training images that are contained in popular person Re-ID datasets, however, usually have limited occlusion variance, and this is a significant factor when trying to avoid overfitting. RE was thus proposed for the optimization of the performance of the person Re-ID model against various occlusions and with the motivation of improving the CNN model's generalization ability. The main idea behind it is to choose rectangle regions randomly, that are of arbitrary sizes, and then assign the pixels some random values or use the mean values. With RE, it is possible to generate varying levels of occlusion without using more parameters for training and learning. This can be thought of as a new method that can be used for data augmentation, which is quite like performing dropout at the image level. The RE technique was adopted to optimize the proposed baseline model, because it is an easy technique that can be implemented and combined with the baseline model. Moreover, no extra parameter learning is required. It can also be combined with random flipping and cropping to be used as a data pre-processing procedure.

In the RE technique, a rectangular region  $I_e$  is randomly selected in an image, and its pixels are erased and replaced with random values. Assuming that the size of the training image is  $W \times H$ , and the image area is  $S = W \times H$ . Following the guidance given in of [77], the erasing rectangle region can be initialized randomly to  $S_e$ , where  $\frac{S_e}{S}$  is within the range that was specified by the minimum  $S_l$  and maximum  $S_h$ . The erasing rectangle region's aspect ratio is initialized randomly from  $r_1$  and  $r_2$ , it is set to  $r_e$ . The size of  $I_e$  is  $H_e = \sqrt{S_e \times r_e}$  and  $W_e = \sqrt{\frac{S_e}{r_e}}$ . Then,

in the original image  $I$  a point  $P = (x_e, y_e)$  is randomly initialized. If  $x_e + W_e \leq W$  and  $y_e + H_e \leq H$ , the region  $I_e = (x_e, y_e, x_e + W_e, y_e + H_e)$  is then chosen as the region to be erased. Otherwise, the above process will be redone until the appropriate rectangle region is selected. In the current study, each of the pixels in  $I_e$  is assigned the value 255 (black), and, respectively, the following values were specified:  $S_l = 0.02$ ,  $S_h = 0.4$ ,  $r_1 = 1$  and  $r_2 = 0.3$  [77]. The procedure used in the selection of the rectangle area and then for erasing this area is given below in Algorithm 3.1.

In the experiments herein, the probability of RE was set to 0.5. This means that RE was used to process half of the training samples. Even though the probability consists of a hyper-parameter that is necessary to tune using a series of very comprehensive experiments, this can be done by simply following the setting of [77]. Figure 3.4 illustrates some erasing results on the Market-1501 dataset.

**Algorithm 3.1** RE procedure

---

<b>Input</b>	: Input image $I$ ; Image size $W$ and $H$ ; Area of image $S$ ; Erasing probability $p$ ; Erasing area ratio range $S_l$ and $S_h$ ; Erasing aspect ratio range $r_1$ and $r_2$ .
<b>Output</b>	: Erased image $I^*$ .
	<b>Initialization:</b> $p_1 \leftarrow \text{Rand}(0, 1)$
1:	<b>if</b> $p_1 \geq p$ <b>then</b>
2:	$I^* \leftarrow I$ ;
3:	<b>return</b> $I^*$ .
4:	<b>else</b>
5:	<b>while</b> <i>True</i> <b>do</b>
6:	$S_e \leftarrow \text{Rand}(S_l, S_h) \times S$ ;
7:	$r_e \leftarrow \text{Rand}(r_1, r_2)$ ;
8:	$H_e \leftarrow \sqrt{S_e \times r_e}$ , $W_e \leftarrow \sqrt{\frac{S_e}{r_e}}$ ;
9:	$x_e \leftarrow \text{Rand}(0, W)$ , $y_e \leftarrow \text{Rand}(0, H)$ ;
10:	<b>if</b> $x_e + W_e \leq W$ and $y_e + H_e \leq H$ <b>then</b>
11:	$I_e \leftarrow (x_e, y_e, x_e + W_e, y_e + H_e)$ ;
12:	$I(I_e) \leftarrow \text{Rand}(0, 255)$ ;
13:	$I^* \leftarrow I$ ;
14:	<b>return</b> $I^*$
15:	<b>end if</b>
16:	<b>end while</b>
17:	<b>end if/else</b>

---



**Figure 3.4** RE results on the Market-1501 dataset.

- **Re-ranking:** When considering the person Re-ID task as the process of data retrieval, the procedure of ranking is always considered as a fundamental step because of its direct association with the results of the Re-ID. Thus, devoting some efforts to Re-ID RR is a critical step in improving its accuracy. Hence, RR [156] was proposed as an optimization technique to optimize the person Re-ID system's performance. The basic idea is to use the  $k$ -reciprocal encoding method [160] for Re-ID RR.

According to [160], suppose that  $N(p, k)$  is a typical sample of the  $k$ -nearest neighbors of the query image  $p$ , which can be defined as:

$$N(p, k) = g_1; g_2; \dots; g_k \quad (3.2)$$

The  $k$ -reciprocal nearest neighbors  $R(p, k)$  can be defined as:

$$R(p, k) = \{g_i \mid (g_i \in N(p, k)) \wedge (p \in N(g_i, k))\} \quad (3.3)$$

It has been proven that the  $k$ -reciprocal nearest neighbors are more closely related to the query  $p$  than the  $k$ -nearest neighbors [156]. However, because of the differences that exist in things such as person poses, occlusions, illuminations, and camera views, it is not necessary to include all of the positive images in the  $k$ -

nearest neighbors, and thus they also do not have to be included in the  $k$ -reciprocal nearest neighbors. Therefore, to deal with this problem, add the  $\frac{1}{2} k$ -reciprocal nearest neighbors of each candidate in  $R(p, k)$  into a more robust set  $R^*(p, k)$  following the condition:

$$R^*(p, k) \leftarrow R(p, k) \cup \left\{ q \mid \left| R(p, k) \cap R\left(q, \frac{1}{2}k\right) \right| \geq \frac{2}{3} \left| R\left(q, \frac{1}{2}k\right) \right|, q \in R(p, k) \right\} \quad (3.4)$$

Then, the  $k$ -reciprocal nearest neighbors set of every image, in the query and gallery sets, is formed as a feature vector, and the Jaccard distance is utilized to calculate the distance between these features. More specifically, the Jaccard distance is calculated as:

$$D_{Jaccard}(p, g_i) = 1 - \frac{|R^*(p, k) \cap R^*(g_i, k)|}{|R^*(p, k) \cup R^*(g_i, k)|} \quad (3.5)$$

Here,  $|\cdot|$  refers to the number of candidates in the corresponding set. A weighted aggregation of the Jaccard distance and the original distance (Cosine distance) is calculated as the final distance for ranking the gallery images. In this work, the weights of both distances were set to 0.5, which is:

$$D_{Final} = \frac{(D_{Original} + D_{Jaccard})}{2} \quad (3.6)$$

The RR technique was adopted to optimize the baseline model proposed in this research, since it is easy to implement and does not require any extra parameters to learn. The procedure of the RR framework for person Re-ID is shown in Figure 3.5.

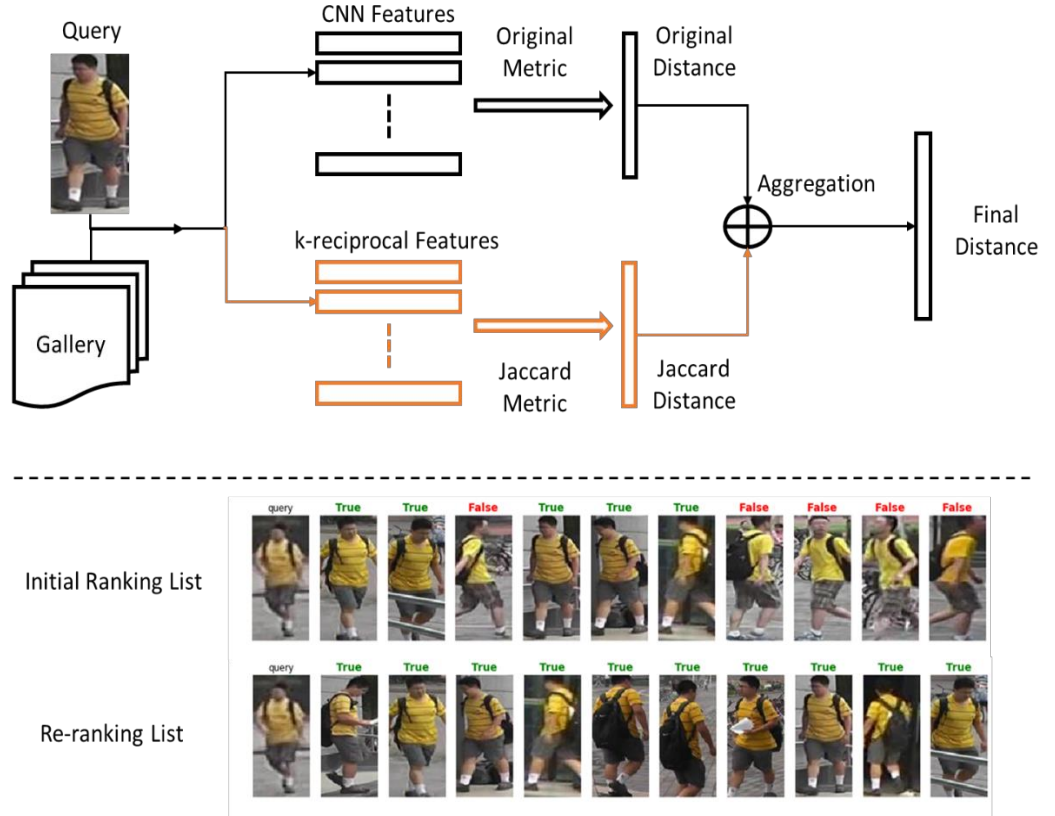


Figure 3.5 RR procedure for person Re-ID.

### 3.3 Style Generative Adversarial Network for Image Generation

Based on the data augmentation concept, and inspired by the excellent results obtained with the StyleGAN architecture when used to produce very diverse and high-quality synthetic pictures of human faces [6]. In the proposed method herein, the StyleGAN architecture was utilized as a data augmentation method to generate high-quality pedestrian images with different styles, and combine them with real pedestrian images to train the proposed baseline model for the person Re-ID task. This section describes the approach using the StyleGAN to generate high-quality and highly diverse person images from the existing person Re-ID datasets.

#### 3.3.1 StyleGAN Architecture

The StyleGAN architecture is one of the latest GAN architectures to achieve superior results in photorealistic high-quality image generation [161]. The StyleGAN architecture is an extension of ProGAN [110] architecture with some generator part

changes motivated by style transfer techniques [162]. The innovative generator architecture was designed to provide control over the style of generated images at various levels of detail. The discriminator and the loss function are quite similar to those used in ProGAN [110].

The architecture of the generator of the StyleGAN model is illustrated in Figure 3.6. In order to generate a synthetic image, the StyleGAN generator used an innovative mapping network as a source of randomness instead of taking a point from the latent space. This mapping network is a NN comprising 8 FC layers, with an output vector ( $w$ ) of size  $512 \times 1$ , similar to the size of the input layer. It was added to the new generator to encode the input vector into an intermediate vector, whose different elements govern various visual features to reduce the phenomenon of feature entanglement. The style vector  $w$ , which outputs from the mapping network, was then transmitted and integrated into every block of the synthesis network to transform it into a visual representation through a new layer named adaptive instance normalization (AdaIN) [162]. The synthesis network consisted of 18 Conv layers, comprising 2 for each resolution (from  $4 \times 4$  to  $1024 \times 1024$ ). Figure 3.7 shows that the AdaIN layer was added to each resolution level of the synthesis network after the Conv layers, and its operation to specify the visual expression of the features at that level was defined as:

$$AdaIN(x_i, y) = y_{s,i} \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \quad (3.7)$$

Here,  $x_i$  represents the feature map,  $y$  is the style input,  $\sigma(x_i)$  the variance of feature map input, and  $\mu(x_i)$  the mean of the feature map input.

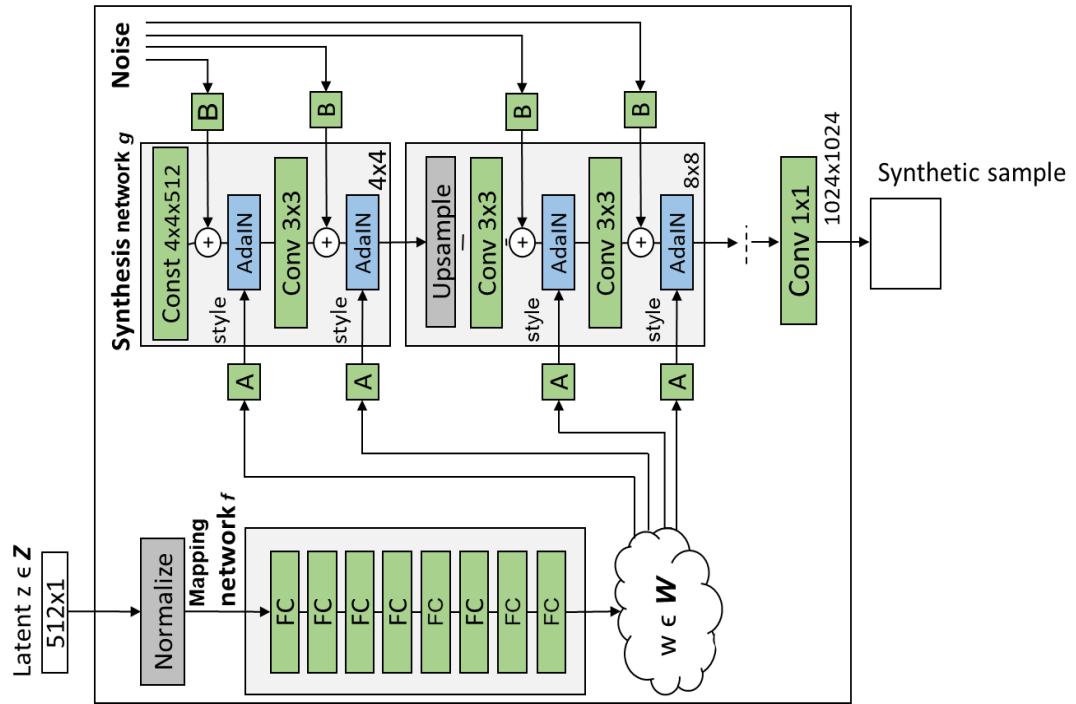


Figure 3.6 StyleGAN generator architecture.

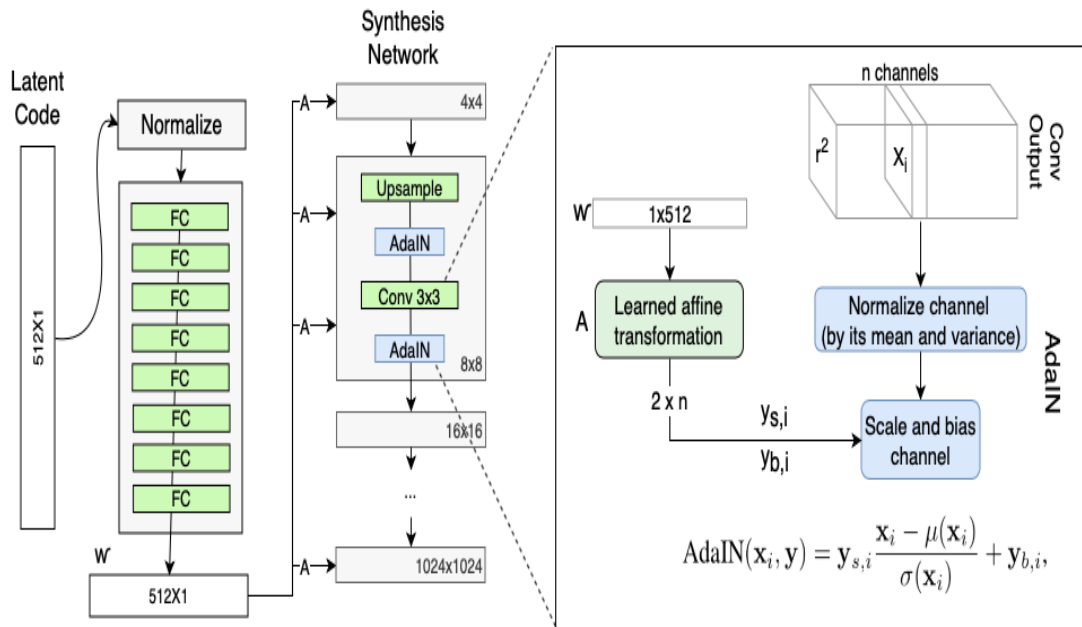


Figure 3.7 AdaIN layer of the generator.

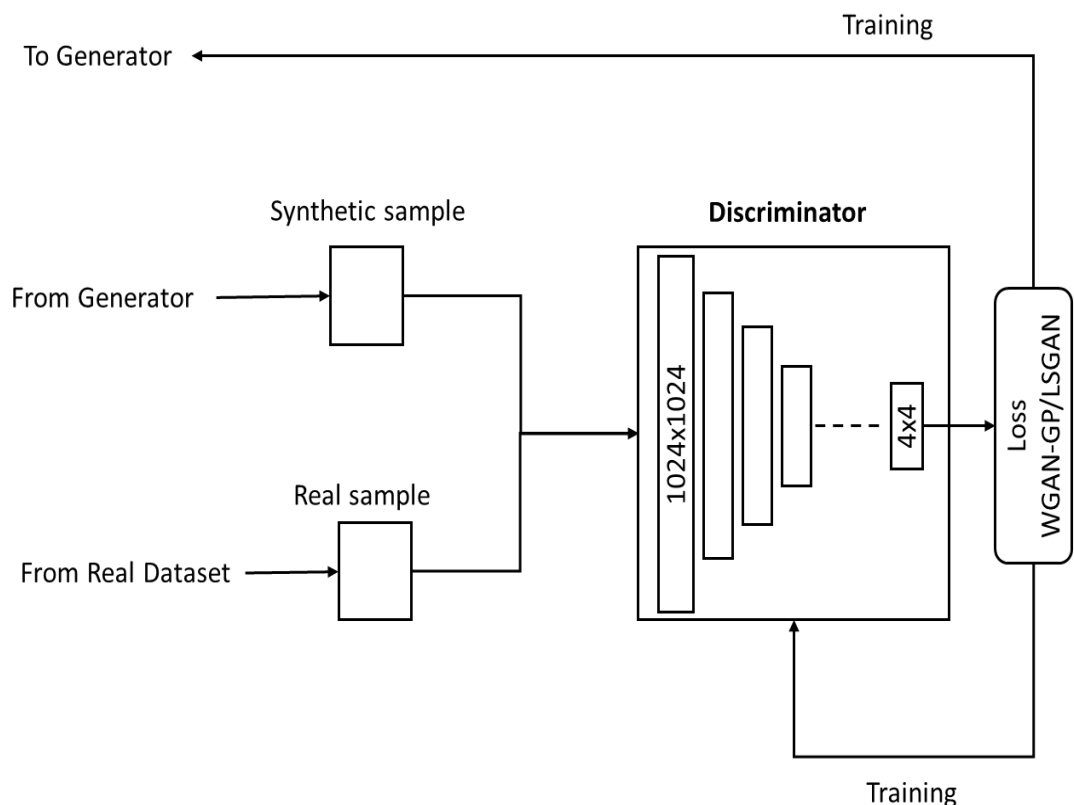
In the synthesis network, Gaussian noise was introduced into the output of each Conv layer before the AdaIN processes. For each block, a distinct noise pattern was generated and interpreted using per-layer scaling factors. This noise was utilized to add style-level differences within the specified level of detail. The last layer output was transformed to RGB by applying a separate  $1 \times 1$  convolution, as proposed by [110]. To prevent StyleGAN from learning that adjacent styles were correlated, the mixing regularization technique was employed, in which 2 random latent codes were used during the training process to produce a given percentage of images rather than one. When generating such an image, the model randomly selects 2 input vectors,  $z_1$  and  $z_2$ , and generates the intermediate vectors,  $w_1$  and  $w_2$ , for them. It then trains some of the levels with the first and switches (at a random point) to the other to train the remaining levels. This operation is referred to as style mixing. Figure 3.8 shows some samples that are generated by mixing 2 latent codes.



**Figure 3.8** Example of style mixing. The fine features in the images within the grid are those that were taken from the images at the top, whereas the coarse features are those that were taken from the images that are on the left.



The architecture of the discriminator of the StyleGAN model is illustrated in Figure 3.9. The discriminator is a classification NN consisting mainly of replicated 3-layer blocks introduced, one-by-one, during the training, and learns to distinguish between real images from the training dataset and the synthetic images generated by the generator. The structure of the discriminator network is explained in Table 3.1. Each layer starts with the Conv of specific kernel size, followed by the down-sampling to the image size that corresponds to the generator network's up-sampling. All of the layers have leaky ReLU activations with  $\alpha=0.2$ . The last layer is an FC layer with an output size equal to 1. This layer returns a decision as to whether the image is real or fake. The training process of the StyleGAN can be considered a zero-sum or minimax game, where the generator is trying to cheat the discriminator while the discriminator is trying not to be deceived by the generator. The discriminator updates its weights through backpropagation and provides a signal that the generator uses to update its weights.

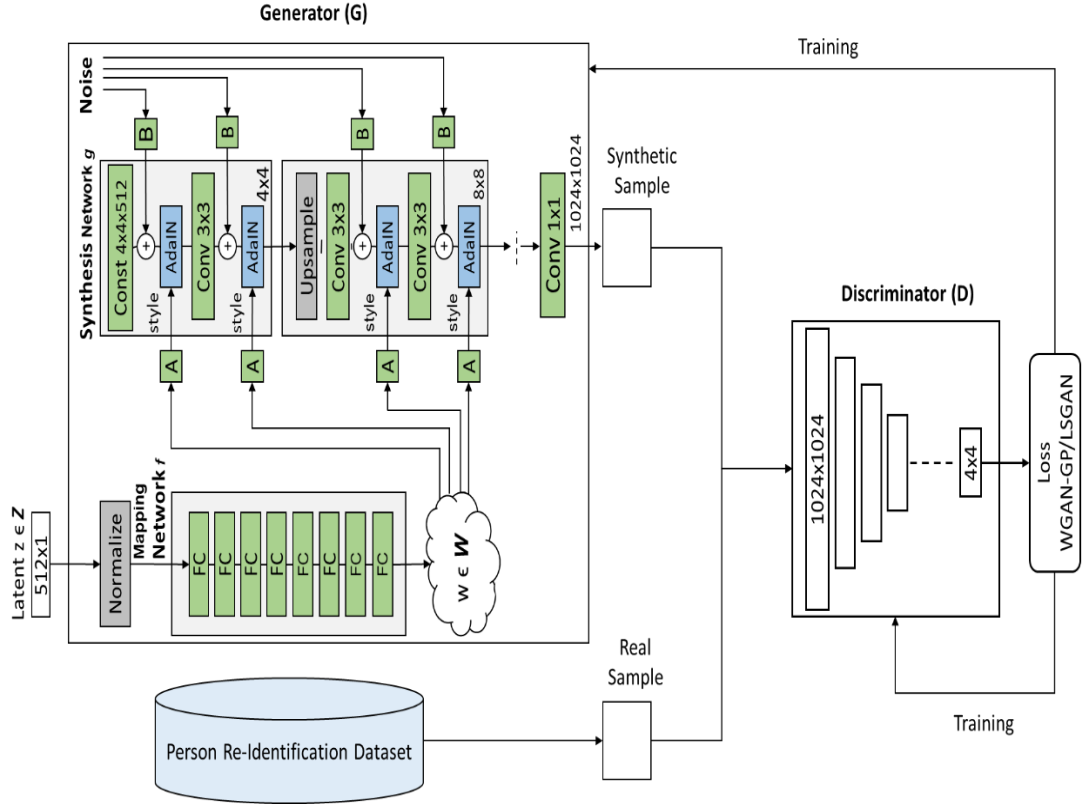


**Figure 3.9** StyleGAN discriminator architecture.

**Table 3.1** Structure of the discriminator network.

Layer	Activation	Output Shape	Params
Input image	-	3×1024×1024	-
Conv 1×1	Leaky ReLU	16×1024×1024	64
Conv 3×3	Leaky ReLU	16×1024×1024	2.3 K
Conv 3×3	Leaky ReLU	32×1024×1024	4.6 K
Downsample	-	32×512×512	-
Conv 3×3	Leaky ReLU	32×512×512	9.2 K
Conv 3×3	Leaky ReLU	64×512×512	18 K
Downsample	-	64×256×256	-
Conv 3×3	Leaky ReLU	64×256×256	37 K
Conv 3×3	Leaky ReLU	128×256×256	74 K
Downsample	-	128×128×128	-
Conv 3×3	Leaky ReLU	128×128×128	148 K
Conv 3×3	Leaky ReLU	256×128×128	295 K
Downsample	-	256×64×64	-
Conv 3×3	Leaky ReLU	256×64×64	590 K
Conv 3×3	Leaky ReLU	512×64×64	1.2 M
Downsample	-	512×32×32	-
Conv 3×3	Leaky ReLU	512×32×32	2.4 M
Conv 3×3	Leaky ReLU	512×32×32	2.4 M
Downsample	-	512×16×16	-
Conv 3×3	Leaky ReLU	512×16×16	2.4 M
Conv 3×3	Leaky ReLU	512×16×16	2.4 M
Downsample	-	512×8×8	-
Conv 3×3	Leaky ReLU	512×8×8	2.4 M
Conv 3×3	Leaky ReLU	512×8×8	2.4 M
Downsample	-	512×4×4	-
Minibatch stddev	-	512×4×4	-
Conv 3×3	Leaky ReLU	512×4×4	2.4 M
Conv 4×4	Leaky ReLU	512×1×1	4.2 M
FC	Linear	1×1×1	513
Total trainable parameters			23.1 M

An overview of the overall StyleGAN architecture is depicted in Figure 3.10. With the presence of novel concepts, such as style mixing and mapping networks, StyleGAN produces high-quality, realistic images with superior control over the visual features of the input images. Moreover, the generated images cover a considerably broader variation in terms of background, color, illumination, and poses. In addition, the synthesis network provides control over the style to different levels of details (or resolution) of the generated image. The generation process starts by creating images at a low resolution ( $4 \times 4$ ) and gradually increases it to the final resolution ( $1024 \times 1024$ ). Various styles can be added at every resolution to produce new synthetic images. At resolution levels between  $4 \times 4$  and  $8 \times 8$ , the coarse styles are introduced, while middle styles are added at resolution levels between  $16 \times 16$  and  $32 \times 32$ . Fine styles are added at resolution levels from  $64 \times 64$  to  $1024 \times 1024$ .



**Figure 3.10** Overall architecture of the StyleGAN model for person images generation.

### 3.3.2 StyleGAN Training

In this section, the loss function and the training mechanism used to train the StyleGAN model to generate person images in the domain of the selected person Re-ID datasets are explained. The StyleGAN model is trained either by the WGAN-GP loss or LSGAN loss. Based on the recommendations of [6], to give the best results, the LSGAN loss function is adopted in this research. The LSGAN loss function can be defined as follows:

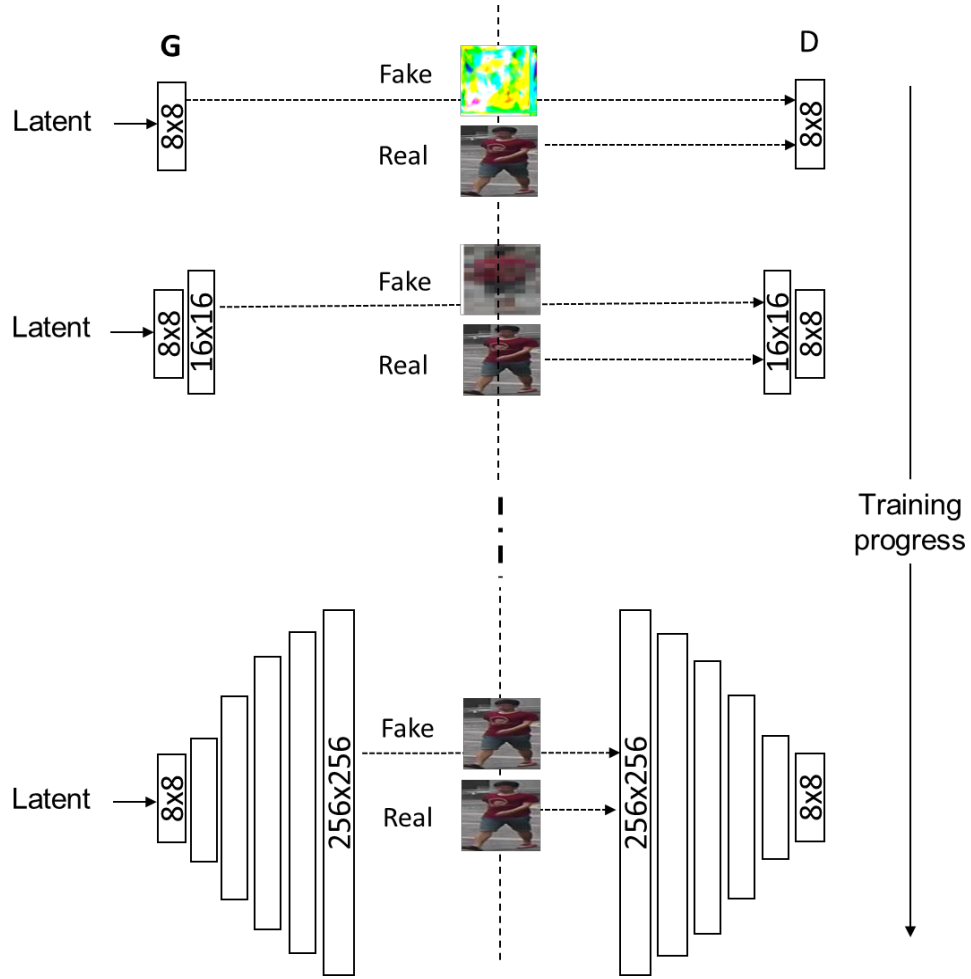
$$\min_D V_{LSGAN}(D) = \frac{1}{2} E_{x \sim p_{data}(x)} \left[ (D(x) - 1)^2 \right] + \frac{1}{2} E_{z \sim p_z(z)} \left[ (D(G(z)))^2 \right] \quad (3.8)$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2} E_{z \sim p_z(z)} \left[ (D(G(z)) - 1)^2 \right] \quad (3.9)$$

Here,  $D(x)$  is the output of the discriminator that denotes the probability of sample  $x$  came from real data,  $E_{x \sim p_{data}(x)}$  represents the expected value over real data

distribution,  $G(z)$  is the output of the generator when given noise  $z$ ,  $D(G(z))$  represents the output of the discriminator that denotes the probability of that the generated fake instance  $G(z)$  is real, and  $E_{z \sim p_z(z)}$  is the expected value over all fake data distribution. The main idea of LSGAN is to use a loss function that provides a smooth and non-saturating gradient in discriminator  $D$ . It is desired that  $D$  pulls data generated by generator  $G$  towards the real data manifold  $P_{data}$ , so that  $G$  generates data that are similar to  $P_{data}$ .

Similar to ProGAN [110], StyleGAN is trained using the progressive growing GAN training method. This means that both the generator and discriminator models start the training by using smaller images, by which only layers in the generator that output this specific size of images are trained. At the same time, only layers with this specific image input size are trained in the discriminator. Generally, the generation process of the StyleGAN starts by creating images at a low resolution ( $4 \times 4$ ) and gradually increases it to the final resolution ( $1024 \times 1024$ ). For the improved StyleGAN model herein to improve the overall result quality, the training process began with a low resolution of  $8 \times 8$  pixels for both the  $G$  and  $D$  models. To increase the resolution of the generated images, more layers were added to  $G$  and  $D$  gradually as the training process progressed, until the desired target image size ( $256 \times 256$ ) was met. The training continued by using the images of the desired  $256 \times 256$  pixel size from the dataset. This technique improved the performance of the training in terms of the speed and stability of StyleGAN drastically [6, 110]. Every added layer was set to continue to be trainable all through the process. This gradually increasing process allowed the training primarily to explore the large-scale structure of the image distribution, and subsequently transferred focus to an increasingly finer scale detail rather than having to master all of the scales concurrently[110]. Figure 3.11 illustrates the progress of the training process. Once the training process was completed, only the trained generator was utilized to generate the synthetic images.



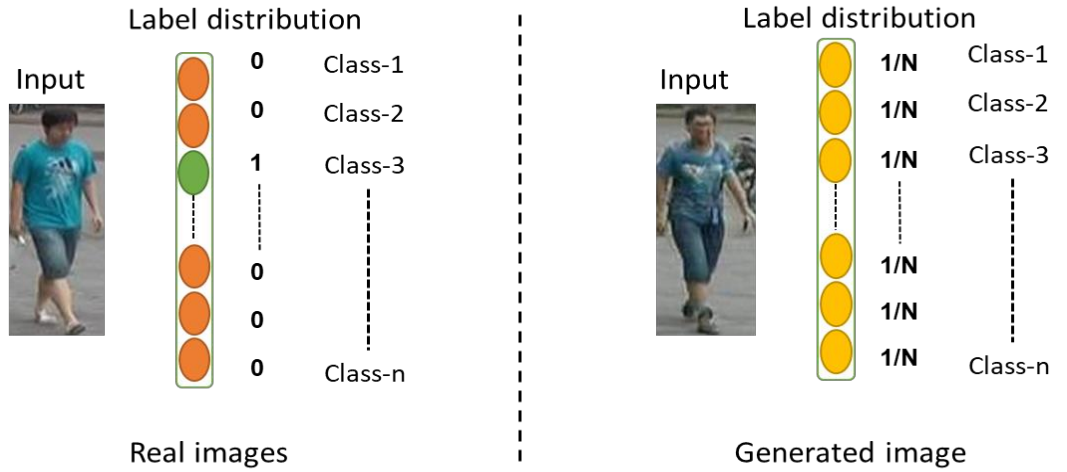
**Figure 3.11** Progress of the training process of the StyleGAN.

### 3.4 Label Assignment Method for StyleGAN Generated Images

This section added the LSRO algorithm [50] to the proposed baseline model for person Re-ID. LSRO was used to integrate the unlabeled images that were generated by StyleGAN into the original labeled training images during the training process by assigning them a uniform label distribution along with the definition of the loss function. LSRO presumed that the newly created images were not affiliated with any existing classes, and it assigned a uniform label distribution (i.e.,  $1/N$ ) to them. LSRO was selected as a label assignment approach to show that the data generated by StyleGAN could help to improve the discriminative learning of the baseline model for person Re-ID tasks. There were 2 reasons why LSRO was adopted:

- 1- StyleGAN was generating the synthetic images under the scenario where no labels are available. Therefore, it was assumed that they were not affiliated with any of the existing classes.
- 2- Since the synthetic images were generated based on mixing style and stochastic variation [6], some visual differences could arise. As a result, LSRO could manage other images positioned close to the actual training images in the sample space and incorporate additional pose variances, color, and illumination for model regularization accompanied by directing the model to find more discriminatory features.

Formally, there were 2 label distributions during the training process. The real image label distribution and the generated image label distribution. These 2 distributions are depicted in Figure 3.12..



**Figure 3.12** Label distributions of a real image and a StyleGAN generated image.

In the real image label distribution, since there is only one true label,  $y$ , intended for each person image in the training set, one item in the corresponding label distribution must be one while the other items are all zero, so the image label distribution  $q(n)$  could be defined as:

$$q(n) = \begin{cases} 1, & n = y \\ 0, & n \neq y \end{cases} \quad (3.10)$$

Here,  $n \in \{1, 2, \dots, N\}$  is the original training classes of the dataset, where  $N$  is the number of classes.

In the generated image label distribution, given that the generated images did not belong to any known training set classes, LSRO assigned the uniform label distribution. The probability that any generated image belonged to any of the known classes was the same by default. Therefore, the label distribution  $q_{LSRO}(n)$  can be written as:

$$q_{LSRO}(n) = \frac{1}{N} \quad (3.11)$$

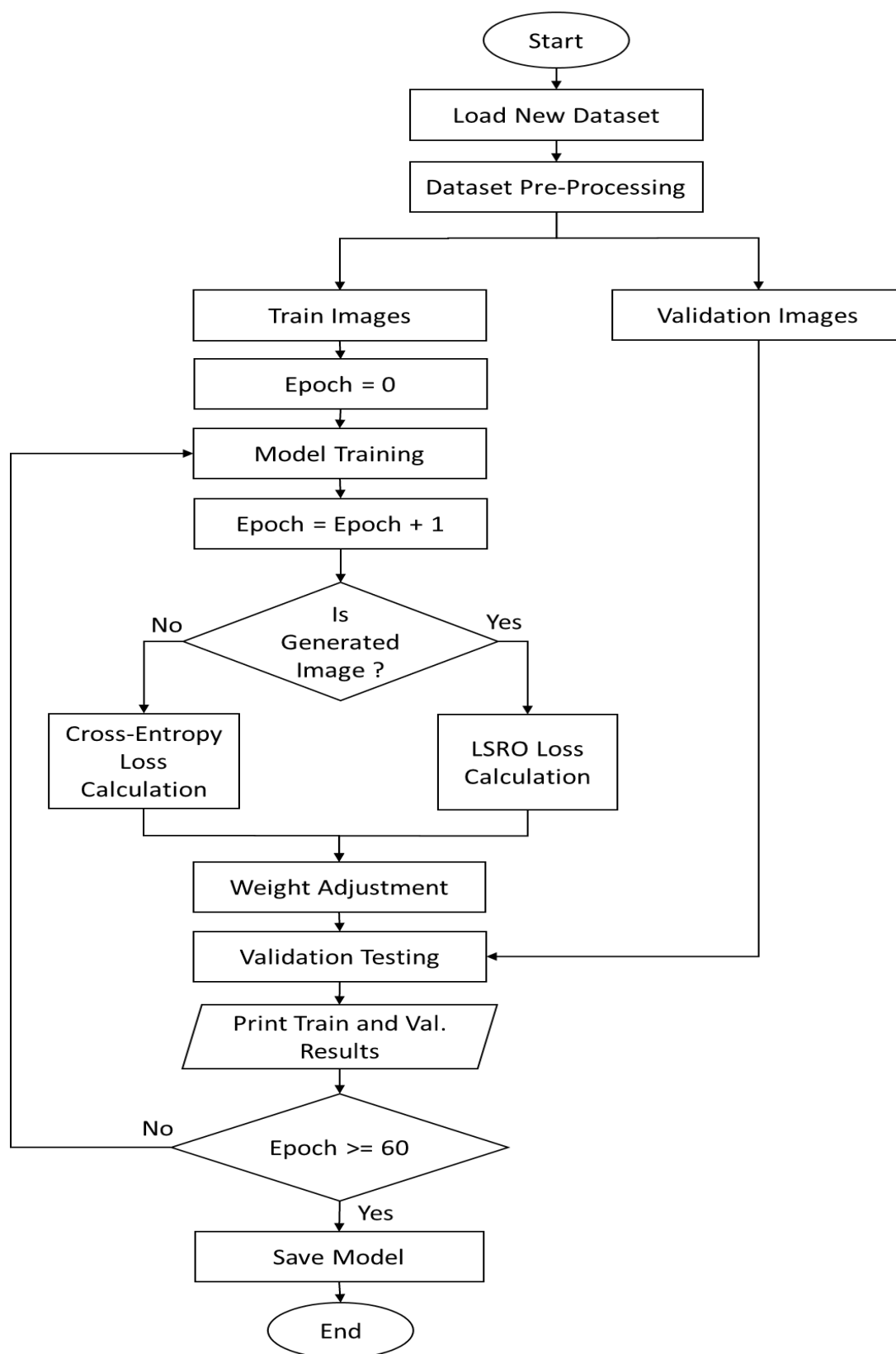
Accordingly, the baseline model needed to have 2 loss functions; one for real images ( $\mathcal{L}_R$ ) and the other for generated images ( $\mathcal{L}_G$ ). The loss function ( $\mathcal{L}_{loss}$ ) that mixed these 2 losses can be defined as:

$$\mathcal{L}_{loss} = \mathcal{L}_R + \mathcal{L}_G \quad (3.12)$$

The proposed baseline model for the person Re-ID was trained by using the cross-entropy loss function stated in Equation (2.5). Thus, integrating Equations (2.5), (3.10), (3.11), and (3.12) resulted in a new cross-entropy loss that could deal with real images and generated images. This cross-entropy loss could be expressed as:

$$\mathcal{L}_{LSRO} = -(1 - \delta) \log(p(y)) - \frac{\delta}{N} \sum_{n=1}^N \log(p(n)) \quad (3.13)$$

Here,  $\delta = 0$  for real training images and  $\delta = 1$  for generated training images. Thus, the baseline model has 2 types of losses, one for real images and one for generated images. The flowchart of the baseline model training using the LSRO cross-entropy loss is shown in Figure 3.13.



**Figure 3.13** Flowchart of training the baseline deep model using real and generated person images.



### 3.5 Experimental Study

This section outlines the experimental settings used to evaluate the proposed method for person Re-ID. Before proceeding to the experiments, the person Re-ID datasets used in the experiments and the preprocessing processes required to prepare these datasets are briefly described. Then, the evaluation protocol used to assess the suitability of StyleGAN to produce new person images of appropriate quality, and the performance of the baseline model for person Re-ID on the selected datasets are described. Finally, the details of the implementation of the proposed baseline model for feature learning and the StyleGAN model for image generation, in addition to their training configuration, are discussed.

#### 3.5.1 Benchmark Datasets

Various datasets for person Re-ID are publicly available. In section 2.6.1, many of these datasets were introduced. To evaluate the proposed method, 3 public standard benchmark person Re-ID datasets were selected for the experiments, namely Market-1501 [22], DukeMTMC-reID [50], and MSMT17 [52]. These datasets were selected because of their relatively large sizes and their popularity in person Re-ID studies. Table 3.2 summarizes the details of these datasets.

**Table 3.2** Details of the selected person Re-ID datasets.

Dataset	Market-1501	DukeMTMC-reID	MSMT17
Release	2015	2017	2018
No. of cams	6	8	15
No. of images	32,668	36,411	126,441
No. of IDs	1,501	1,812	4101
No. of training images	12,936	16,522	32,621
No. of training IDs	751	702	1041
No. of gallery images	19,732	17,661	82,161
No. of gallery IDs	750	1,110	3060
No. of query images	3,368	2,228	11,659
No. of query IDs	750	702	3060
Detector	DPM	Hand	Faster RCNN
Image size	128 × 64	Varies	Varies

In order to properly evaluate person Re-ID models, a good person Re-ID dataset has to mirror the actual video surveillance setting in a real-world scenario, such as the

viewpoint changes, differences in illumination, differences in background, and camera characteristics. A realistic dataset should include images taken from different surveillance cameras to capture the same identity from different viewpoints and trajectories. According to Table 3.2, the datasets for Market-1501, DukeMTMC-reID, and MSMT17 have 12,936, 16,522, and 32,621 training samples, respectively. For a DL-based person Re-ID task, this data size is relatively small. Therefore, in this research, StyleGAN was adopted as the generative model for synthetic image generation to enlarge the training sets.

### 3.5.2 Data Pre-processing

Before utilizing the datasets to train the baseline model for feature learning or to train the StalyGAN model for image generation, it was necessary to perform some pre-processing procedures in order to make the datasets more suitable for the training process, as well as to drive the models to achieve better performance.

**Data pre-processing for the baseline model training:** Before training the baseline model for feature learning, the following pre-processing operations on the training data were required.

- 1- Image resizing and cropping:** Since all of the images in the Market-1501 dataset have a uniform size ( $128 \times 64$ ) and the sizes of the images in the DukeMTMC-reID and MSMT17 datasets are not uniform. It was necessary to make sure that all of the input images had the same size and aspect ratio. All of the input images were first resized to the size of  $288 \times 144$ , and then randomly center-cropped to  $256 \times 128$ . Together, these 2 operations reduced the contribution of the background in the original images while keeping the image size consistent.
- 2- Normalization:** Data normalization is a crucial step to ensure that each input parameter (pixels) has a similar data distribution to make the model's convergence faster while training it. Usually, normalization is conducted by subtracting the mean of the images from each pixel and then dividing it by their standard deviation. After normalization, the distribution of the input data would resemble a Gaussian curve centered at zero. According to [50], the images of Market-1501 must be normalized with a mean of  $[0.485, 0.456, 0.406]$  and standard deviation

of  $[0.229, 0.224, 0.225]$ , which were scaled from  $[0, 255]$  to  $[0, 1]$ . As for DukeMTMC-reID and MSMT17, the same mean and standard deviation were used. Figure 3.14 provides several examples of the normalized images from Market-1501.



**Figure 3.14** Examples of the normalized images from Market-1501.

**3- Data augmentation:** Usually, some data augmentation techniques are applied before the training process, including rotation, flipping, etc. Data augmentation is conducted to expose the model to a wide variety of variations to make sure that it is less likely that the NN will recognize unwanted characteristics in the dataset. In the experiments herein, all of the training images were horizontally flipped, randomly. Moreover, the RE method [77] was utilized to improve the data augmentation efficiency further. Figure 3.15 provides several examples of the horizontally random flipped images from Market-1501.



**Figure 3.15** Examples of the horizontally random flipped images from Market-1501.

**Data pre-processing for the StyleGAN training:** Before training the StyleGAN model for image generation, it was mandatory to prepare the datasets according to the StyleGAN training restrictions. These pre-processing operations included the following.

- 1- Scaling:** All of the images contained in the StyleGAN training set must be exactly uniform and in square dimensions to train precisely. Because all of the images in Market-1501 are  $128 \times 64$  and the images in DukeMTMC-reID and MSMT17 have a non-uniform size, a re-sizing of the images is required. Therefore, the resolution of every training image has been resized to  $256 \times 256$ . Accordingly, the produced images would have the same size. Figure 3.16 illustrates some samples from Market-1501 after resizing to  $256 \times 256$ .



**Figure 3.16** Sample images from the Market1501 dataset after resizing to  $256 \times 256$ .

- 2- Shuffling:** In order to guarantee that all of the images were not in alphabetical order of their species names, the ordering of images was shuffled, and then they were assigned labels before saving them.
- 3- Generating TFRecords:** The training and evaluation of the StyleGAN model were performed on datasets stored as multi-resolution TensorFlow supported data format (TFRecords). Therefore, each dataset was represented by a directory containing the same image data in several resolutions to enable efficient streaming. There was a separate \*.tfrecords file for each resolution, which stores each image as arrays at that resolution.

### 3.5.3 Performance Evaluation Metrics

This section describes the evaluation metrics for assessing the baseline model's performance for the person Re-ID task on the selected datasets. Moreover, it describes the prominent metrics to measure the extent of the quality and diversity of the StyleGAN generated images to confirm the suitability of StyleGAN to produce new person images that can be used to enlarge the selected datasets, in order to improve the performance of the baseline model.

**Baseline model performance evaluation metrics:** To evaluate the performance of the baseline model for the person Re-ID process, the same ranking-based evaluation method introduced in [36] was adopted, which was conducted by matching (by cosine similarity) the feature vectors of all the gallery images against a query representation and then sorting the correspondent similarity scores in decreasing order. The 2 most commonly used evaluation metrics explained in Section 2.6.2, namely CMC and mAP, were used to measure the Re-ID performance. The CMC curve plotted the precision performance versus the rank score and represented the expectation of finding the correct match inside top-ranking results, while mAP evaluated the recall simultaneously. Since there was more than one ground truth in the gallery images, mAP was more comprehensive to describe how well all of the ground truths were ranked. While ranking the gallery images with respect to the query images, all of the images with the same labels and same camera identities were regarded as ‘bad matches’ because it was desired to match the people using different cameras; hence they were removed from their corresponding ranking list. However, it was neither a positive nor negative sample.

In the evaluation process of person Re-ID, there are 2 main query modes: single- and multi-query. In single-query mode, a person Re-ID system only analyzes a single query image. Ranking of the gallery images is performed by using the distance that is between the query image and every other image in the gallery. In multi-query mode, multiple query images are provided, and then the ranking of the gallery images is performed using the average distance that is between the query images and every other image in the gallery. Therefore, the multi-query mode has more robustness and most often achieves higher accuracy because it considers multiple query images. Single-

and multi-query modes were both applied in the evaluation of the Market 1501 dataset, whereas only single-query mode was applied in the evaluation of the DukeMTMC-reID and MSMT17 datasets. Only CMC accuracy at Rank-1, -5, and -10 were reported, rather than plotting the real curves for use in the comparisons. As a result of its success in improving Re-ID accuracy, RR with  $k$ -reciprocal encoding [156] was used in this research, and its evaluation results were also reported. The re-identification evaluation results will be shown later in Chapter 4.

**StyleGAN model performance evaluation metrics:** The goal of StyleGAN is the generation of person images that have high quality, which contributes to the final Re-ID performance. To assess StyleGAN’s suitability for producing new person images that will be of appropriate quality, the newly-generated synthetic images were evaluated both qualitatively and quantitatively. For the qualitative evaluations, the samples that were generated using the StyleGAN approach were visually compared with the samples that were from the real images and also with the samples that were generated by some other generative approaches, which were designed specifically for the person Re-ID task. Visually examining the generated samples via human judgment is among the most common and intuitive methods that are used in the evaluation of the quality that the generated images possess. For the quantitative evaluation, the same methods were followed as those that were introduced by [56]. The Fréchet inception distance (FID) score [163] was used, as well as structural similarity (SSIM) [164], in the quantification of the realism and diversity within the synthetic samples that were produced using the StyleGAN model.

FID is the currently used state-of-the-art metric that was developed for the evaluation of the performance of GANs, because it can measure how close the distribution of the generated image is close to the actual image. A lower FID means smaller distances between the synthetic and real data distribution, thus indicating better quality images. In the process used for the FID calculation, the previously trained Inception-v3 model [84] was used to perform the feature extraction via removal of its output layer and using the 2048 feature vector, which was taken from the last global pooling layer. Then the calculation of these feature vectors was performed to obtain a collection of both the real and the generated images. The 2 groups of feature vectors can be summarized

as multivariate Gaussian distribution by calculating the mean ( $\mu$ ) and covariance ( $\sigma$ ), for both real images ( $r$ ) and generated images ( $g$ ). The FID score is then calculated using Equation (3.14).

$$FID = \|\mu_r - \mu_g\|^2 + T_r(\sigma_r + \sigma_g - 2\sqrt{\sigma_r \sigma_g}) \quad (3.14)$$

Here,  $\mu_r$  and  $\mu_g$  are the feature-wise means, respectively, of the real and generated images,  $\sigma_r$  and  $\sigma_g$  are the covariance matrices, respectively, of real and generated images, and  $T_r$  is the sum of the diagonal elements of the matrix.

SSIM is used to compute the similarity of the intra-class, and indicate the distance between the generated image and the original image. Therefore, it can be used to give an indication of the variety within the generated images. A higher SSIM signifies greater variety in the images that have been generated. The SSIM is based on 3 comparison measurements between the original (real) images and the generated images, namely luminance (l), contrast (c), and structure (s):

$$l(r, g) = \frac{2\mu_r \mu_g + c_1}{\mu_r^2 + \mu_g^2 + c_1} \quad (3.15)$$

$$c(r, g) = \frac{2\sigma_r \sigma_g + c_2}{\sigma_r^2 + \sigma_g^2 + c_2} \quad (3.16)$$

$$s(r, g) = \frac{\sigma_{rg} + c_3}{\sigma_r \sigma_g + c_3} \quad (3.17)$$

The SSIM is then a weighted combination of those comparative measures:

$$SSIM(r, g) = \left[ l(r, g)^\alpha \cdot c(r, g)^\beta \cdot s(r, g)^\gamma \right] \quad (3.18)$$

Setting the weights  $\alpha$ ,  $\beta$ ,  $\gamma$  to 1, the SSIM formula can be reduced to the form shown below.

$$SSIM(r, g) = \frac{(2\mu_r\mu_g + c_1)(2\sigma_{rg} + c_2)}{(\mu_r^2 + \mu_g^2 + c_1)(\sigma_r^2 + \sigma_g^2 + c_2)} \quad (3.19)$$

Here,  $\mu_r$  and  $\mu_g$  are the feature-wise means, respectively, of the real and generated images,  $\sigma_r$  and  $\sigma_g$  are the covariance matrices, respectively, of real and generated images, and  $c_1$  and  $c_2$  are the stabilizing constants.

### 3.5.4 Experimental Setup

The experimental phase involves putting the proposed person Re-ID method into action. It includes the implementation details, and all of the experiments of the training and evaluation, of both the baseline model and the StyleGAN model. The main experiments of the proposed StyleGAN-LSRO method for person Re-ID are:

- 1- Training the baseline model using the pre-processed original person Re-ID datasets.
- 2- Using the trained baseline model to extract the features of query and gallery images and evaluate its performance for person Re-ID.
- 3- Training the StyleGAN model using the original datasets.
- 4- Generating synthetic samples with the trained StyleGAN generator and evaluate its generated samples.
- 5- Adding the generated synthetic samples to the training set of the dataset.
- 6- Training the baseline model with the enlarged training set using the LSRO loss.
- 7- Using the baseline model trained with the new training set to extract the features of query and gallery images and again evaluate its performance for person Re-ID.

**Baseline model implementation details:** The baseline model proposed in Section 3.2 was implemented using the PyTorch framework, which is an open-source Python library for DL. The PyTorch framework is introduced in Appendix C. The experiments were conducted on a personal computer with NVIDIA GeForce GTX 1050 Ti GPU. The operating system used was 64-bit Windows. The pre-trained CNN models used were ResNet-50 and DenseNet-121, which were previously trained on ImageNet. The mini-batch SGD with momentum and cross-entropy loss was employed as an



optimizer to train the baseline model. The momentum was set to 0.9, and the weight decay was set to  $5e-4$ . Training of the model was performed for 60 epochs, with the batch size set to 32, and initialization of the base learning rate for the pre-trained layers set at 0.002, in addition to the added 2 FC layers at 0.02. Next, the learning rate was decreased after 40 epochs by a decay factor of 0.1. The last FC layer outputs were changed to 751, 702, and 1041, respectively, for Market-1501, DukeMTMC-reID, and MSMT17. The RE probability was empirically tuned to 0.8, as was the dropout ratio, to 0.75. Training the baseline model with each of the 3 utilized person Re-ID datasets took about 3 to 4 h. After training, the trained model was saved from being used later for feature extraction.

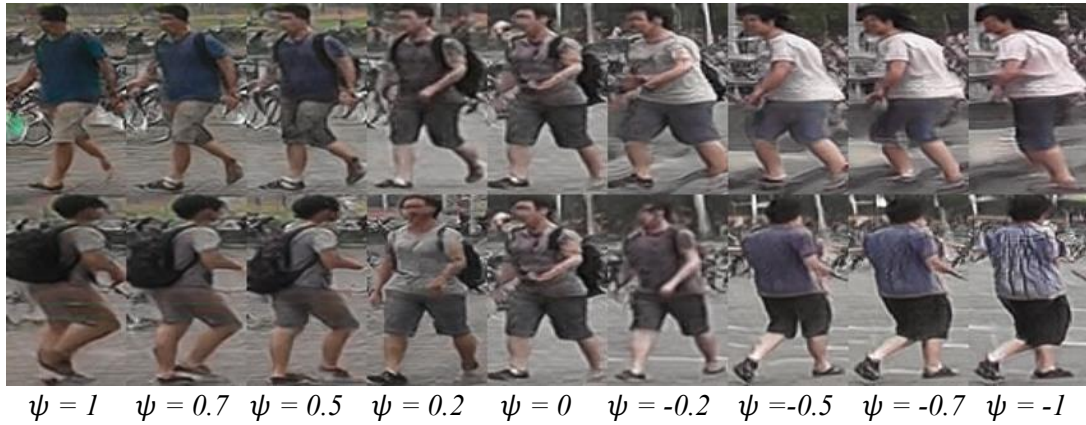
**StyleGAN model implementation details:** The StyleGAN model proposed in Section 3.3 is implemented based on the formal TensorFlow implementation [165] of StyleGAN by Karras et al. [6], from which most of the training aspects were inherited. The TensorFlow platform is also introduced in Appendix D. Several hyperparameters were tuned to attain better-quality generated images. The experiments were conducted on a NVidia Tesla K80 GDDR5 24GB 4992 CUDA CORES GPU. Training of the model was performed via the Adam optimizer using the following parameters:  $\beta_1 = 0.0$  and  $\beta_2 = 0.99$ . The size of the minibatch was set to resolution-dependent sizes that comprised 8:128, 16:64, 32:32, 64:16, 128:8, and 256:8. Implementation of the progressive growth was performed according to the method of Karras et al. [13] with the exception that it began with a resolution of  $8 \times 8$  rather than  $4 \times 4$ . Initially, the learning rate was set to 0.0015, after which it was increased to 0.002 when the resolution had reached  $256 \times 256$ . The training length was set to 15M images. The duration of training was about 3 weeks. During the StyleGAN testing, for the generation of the synthetic images, only the trained generator was used. A 512-dimensional random vector ( $z$ ) was input over the range  $[-1, 1]$ , as well as a Gaussian distribution to the trained StyleGAN generator that was transformed into intermediate vectors ( $w$ ) using the mapping network. The average quality of the generated images can be further improved by drawing the latent vectors from a truncated sampling space [6]. To do so, the mean of the intermediate vector  $w$  must first be computed, as shown below:

$$w_{mean} = E_{Z \sim P(Z)} [f(Z)] \quad (3.20)$$

Then, this point can be used to scale the deviation of a given  $w$  from the center as:

$$\bar{w} = w_{mean} + \psi (w - w_{mean}) \quad (3.21)$$

Here,  $\psi$  is the constant that controls the style scale. Figure 3.17 shows the effect of the truncation trick as a function of  $\psi$  where 9 images were generated at different values of  $\psi$  (1, 0.7, 0.5, 0.2, 0, -0.2, -0.5, -0.7, and -1). From this figure, it can be observed that the best image quality was at  $\psi = 0.7$  and  $\psi = 0.5$ . Therefore, these 2 values were recommended to be used to generate the images.



**Figure 3.17** Effect of truncation trick as a function of style scale  $\psi$ .

In conclusion, in this chapter, the proposed StyleGAN-LSRO method for person Re-ID was explained in detail. Moreover, the conducted experimental study and the steps required to implement the proposed method were described. The procedure for person Re-ID using the proposed StyleGAN-LSRO method included the development and training of the baseline CNN model to learn the discriminative features, as well as the training of the StyleGAN model using original person Re-ID datasets to generate high-quality images that were then used to enlarge the training sets to increase the discriminative ability of the baseline model. The LSRO algorithm was used to integrate the generated images into original labeled training images during this training process via the assignment of a uniform label distribution as well as the definition of the loss function. In the next chapter, the experimental results obtained from training and evaluation of the proposed method will be presented and also compared to the state-of-the-art methods.

# CHAPTER 4

## EXPERIMENTAL RESULTS AND DISCUSSION

In this chapter, toward the validation and evaluation of the performance of the proposed method for person Re-ID, a series of systematic experiments were conducted. All of the experiments were implemented according to those given in Section 3.5.4. The experiments were carried out using the Market-1501, DukeMTMC-reID, and MSMT17 datasets. The performance was evaluated according to evaluation metrics explained in Section 3.5.3. The experimental results of the training and evaluation of the baseline and StyleGAN models were shown in detail. First, the results of the training and evaluation of the proposed baseline model on the original datasets are presented, moreover, with the optimization methods that were previously mentioned in Section 3.2.4. Afterward, the StyleGAN model's suitability in producing images that are of high quality is evaluated. Then, an assessment of the performance of our baseline model is made to show how the new StyleGAN-generated high-quality synthetic images contributed to performance improvement. Moreover, to reveal the proposed method's superiority, the findings that were obtained using the proposed method were compared with those of the state-of-the-art methods.

### 4.1 Baseline Model Training and Evaluation

This section presents the results of training and evaluation of the proposed baseline model. The training results included the accuracy and loss of both the training and validation data using the original person Re-ID datasets. To prove the efficiency and generality of the proposed approach to build the baseline model, 2 CNN architectures were used, namely Resnet-50 and Densenet-121, which are widely used for a person Re-ID task. The results of both are presented herein. To evaluate the Re-ID performance of the proposed baseline model, query and gallery images from the 3 datasets were used. The query and gallery person images were passed through the baseline model. Their extracted features were saved to a file. Then, they were ranked, and their similarities were calculated according to Cosine similarity. The results were

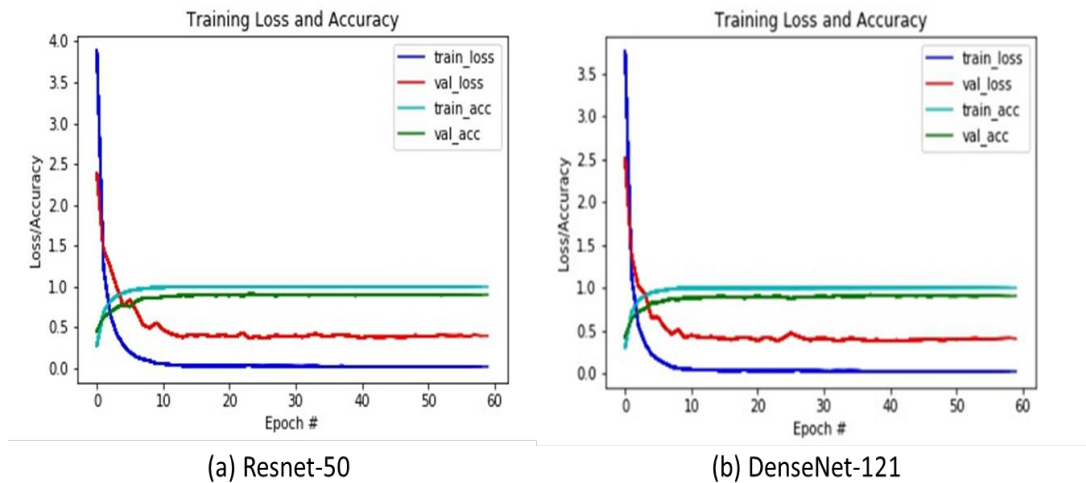
presented using CMC curves and mAP. For comparison, first, evaluation of the proposed baseline model was performed using the original dataset, without the addition of the RE and RR procedures. Next, the efficiency of RE and RR on the proposed baseline model was explored.

#### 4.1.1 Baseline Model Training Results

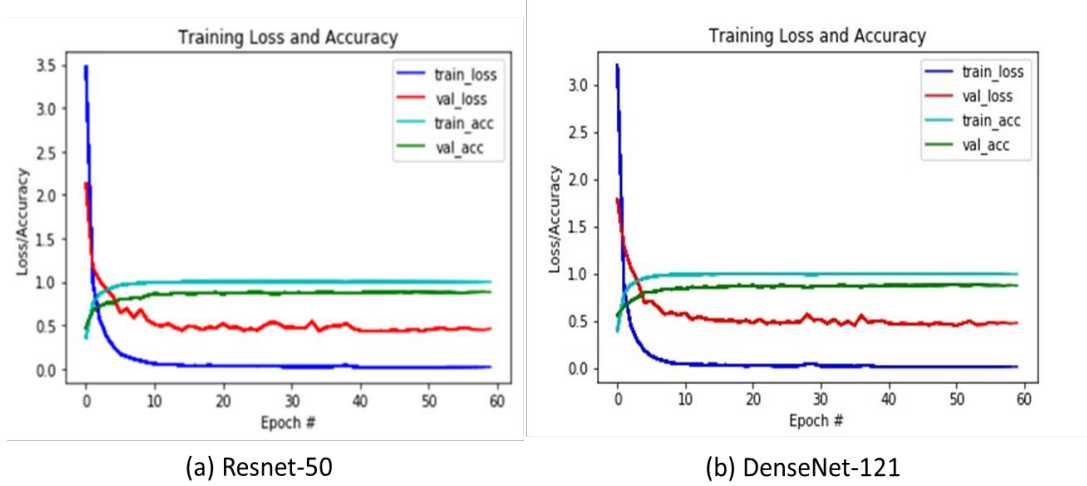
During the training of the baseline model for the person Re-ID task, the current approach with regard to the training process was that of a classification task, where each person's identity was considered as a separate class. The first image of every identity was used as a validation set, and the remaining samples were regarded as a training set. The classification accuracy results on validation sets can be seen in Table 4.1 below. The learning progress of the baseline model on the 3 datasets can be seen in Figures 4.1, 4.2, and 4.3, respectively.

**Table 4.1** Validation accuracy.

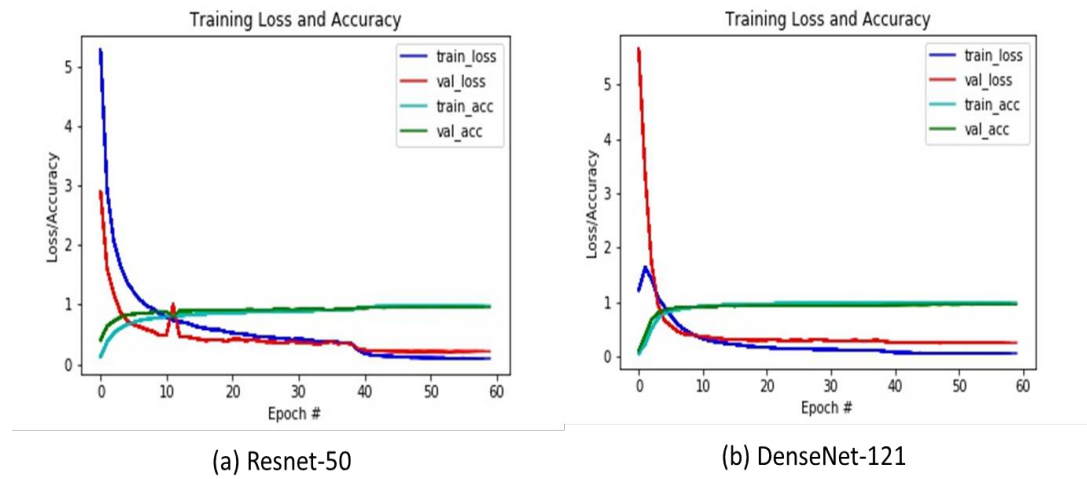
Backbone	Dataset		
	Market-1501	DukeMTMC-reID	MSMT17
ResNet-50	91.21%	88.32%	95.95%
DenseNet-121	90.41%	87.75%	<b>94.75%</b>



**Figure 4.1** Training process of the baseline model on the Market-1501 dataset.



**Figure 4.2** Training process of the baseline model on the DukeMTMC-reID dataset.



**Figure 4.3** Training process of the baseline model on the MSMT17 dataset.

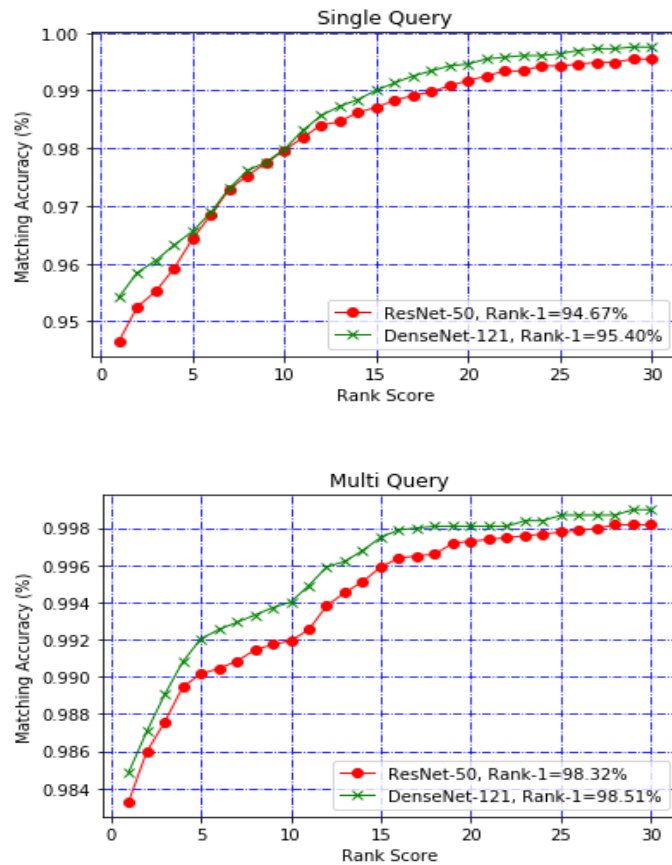
From the loss and accuracy curves of the training and validation shown in Figures 4.1, 4.2, and 4.3, it can easily be seen that in each situation, the baseline model was very close to saturation and was not able to reach optima after training for 40 epochs. However, after the learning rate was decreased, the baseline model was able to continue to learn for a few more epochs. Finally, the baseline model almost reached 100% accuracy with both the training set and the validation set. Therefore, it is possible to say that these results provided a good indication that the proposed baseline model performed well with regard to avoiding overfitting, and as a result, it would be able to

work well for the person Re-ID task, as it would be able to provide satisfactory discriminative features from the images of people.

#### 4.1.2 Baseline Model Evaluation Results

To demonstrate the effectiveness of the proposed baseline model in the person Re-ID task, the performance of the baseline model was first evaluated on the original dataset, without the addition of RE and RR procedures. After that, the efficiency of adding RE and RR to the baseline model was explored. Ranks-1, -5, and -10 on the CMC curves were presented for easier comparison.

**The performance of the baseline model without RE and RR:** On the Market-1501 dataset, and since it supports multi-query mode, the CMC curves of the single- and multi-query mode are shown in Figure 4.4. The achieved Rank-1, -5, and -10 accuracies, and mAP, are given in Table 4.2.

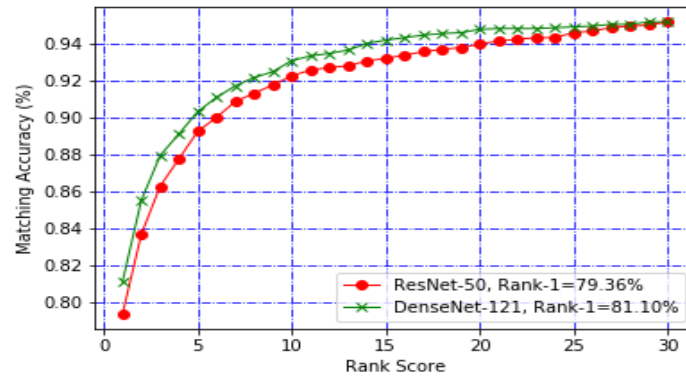


**Figure 4.4** CMC curves on the Market-1501 dataset.

**Table 4.2** Market-1501 dataset evaluation results.

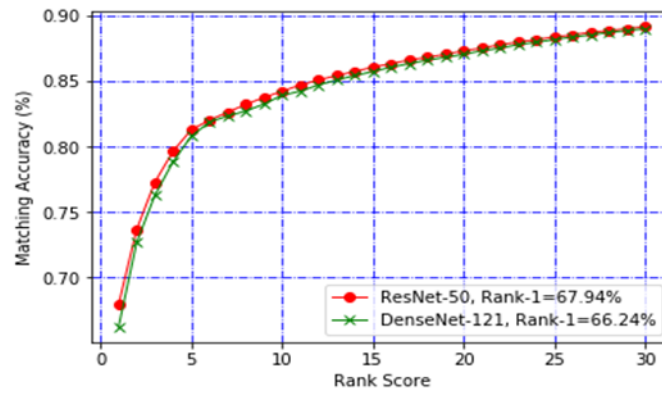
Backbone	Single-query				Multi-query			
	Rank-1	Rank-5	Rank-10	mAP	Rank-1	Rank-5	Rank-10	mAP
ResNet-50	94.67	96.40	97.90	74.50	98.32	99.01	99.19	86.38
DenseNet-121	95.40	96.50	97.90	75.30	98.51	99.20	99.39	87.77

The DukeMTMC-reID dataset was only tested with the single-query setting, since it does not allow for the multi-query mode. The CMC curves are illustrated in Figure 4.5. The achieved Rank-1, -5, and -10 accuracies, and mAP, are given in Table 4.3.

**Figure 4.5** CMC curves on the DukeMTMC-reID dataset.**Table 4.3** DukeMTMC-reID dataset evaluation results.

Backbone	Rank-1	Rank-5	Rank-10	mAP
ResNet-50	79.36	89.22	92.23	62.60
DenseNet-121	81.10	90.30	93.05	63.40

The MSMT17 dataset also does not allow for multi-query mode, so it was evaluated with the single-query setting. The CMC curves are illustrated in Figure 4.6. The achieved Rank-1, -5, and -10 accuracies, and mAP, are given in Table 4.4.



**Figure 4.6** CMC curves on the MSMT17 dataset.

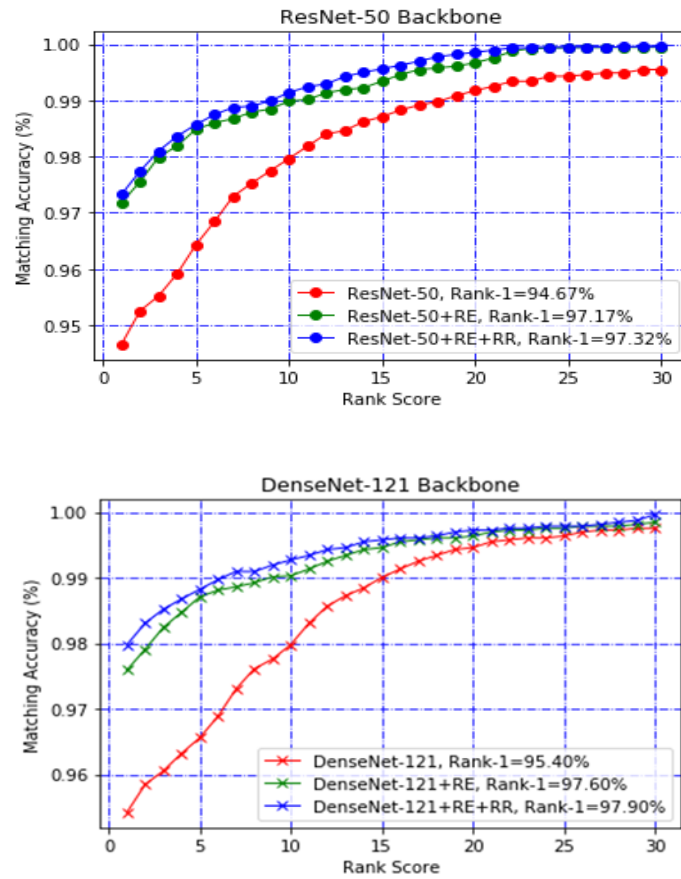
**Table 4.4** MSMT17 dataset evaluation results.

Backbone	Rank-1	Rank-5	Rank-10	mAP
ResNet-50	67.94	81.77	84.26	45.66
DenseNet-121	66.24	81.37	83.96	44.26

For verification of the effectiveness and generality of the proposed baseline model, it was applied to the Resnet-50 and Densenet-121 architectures, which are used widely in person Re-ID tasks. It was obvious from the findings presented above that the proposed baseline model was able to achieve an appreciably good performance. These results provided a glimpse into the ability of the proposed baseline model to avoid overfitting.

**Performance of the baseline model with RE and RR:** Here, the results related to the impact of the optimization methods mentioned in Section 3.2.4 are presented, namely RE and RR, on the efficiency of the proposed baseline model. It is essential to note that all of the tests were performed with the single-query setting only, because the RR technique cannot be applied to the model with a multi-query setting. For the Market-1501, DukeMTMC-reID, and MSMT17 datasets, the CMC curves are shown in Figures 4.7, 4.8, and 4.9, respectively. Moreover, the achieved Rank-1, -5, and -10 accuracies, and mAP, are shown in Tables 4.5, 4.6, and 4.7, respectively. Briefly, the performance evaluation results of the proposed baseline model after adding RE and RR demonstrated that the proposed method was effective in performing the person Re-ID task.





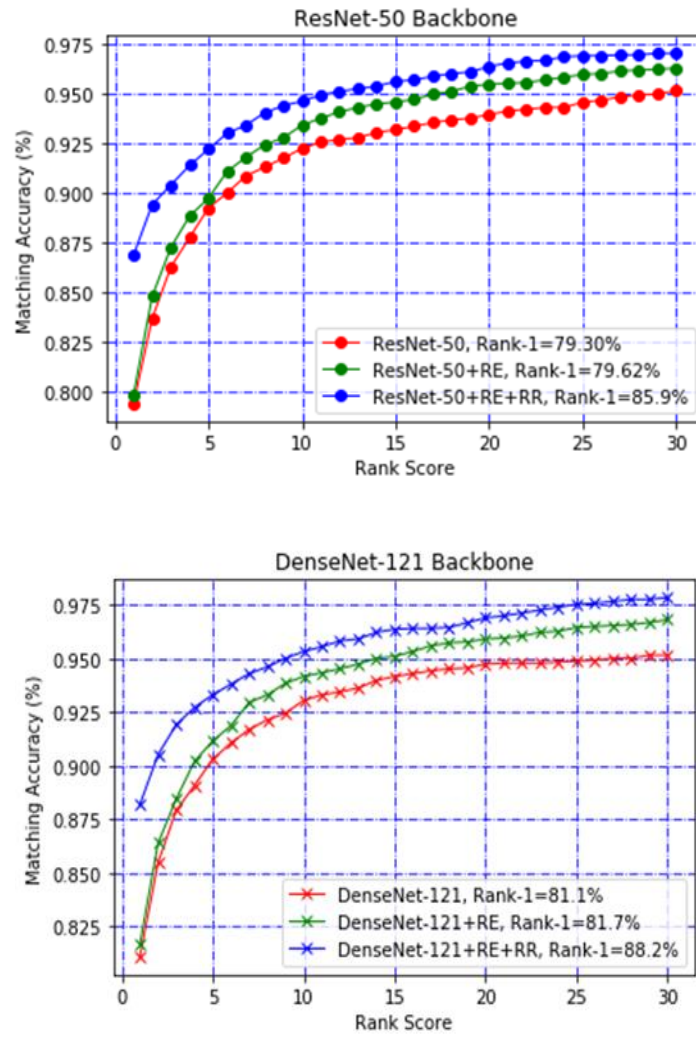
**Figure 4.7** CMC curves on the Market-1501 dataset with RE and RR.

**Table 4.5** Market-1501 dataset evaluation results with RE and RR.

Backbone	Rank-1	Rank-5	Rank-10	mAP
ResNet-50 + RE	97.17	98.48	98.98	79.23
ResNet-50 + RE + RR	97.32	98.66	99.03	90.74
DenseNet-121+ RE	97.60	98.70	99.02	80.20
DenseNet-121+ RE+ RR	97.90	98.80	99.30	91.80

**Table 4.6** DukeMTMC-reID dataset evaluation results with RE and RR.

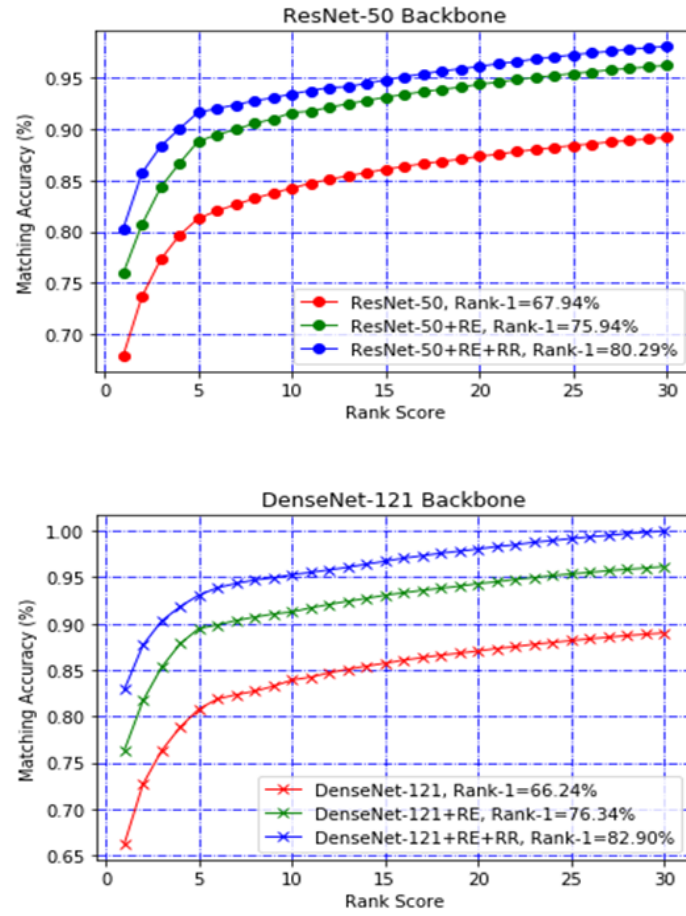
Backbone	Rank-1	Rank-5	Rank-10	mAP
ResNet-50 + RE	79.62	89.72	93.40	64.50
ResNet-50 + RE + RR	85.90	91.33	93.76	82.19
DenseNet-121+ RE	81.70	91.20	94.20	66.30
DenseNet-121+ RE+ RR	88.20	93.30	95.30	85.00



**Figure 4.8** CMC curves on the DukeMTMC-reID dataset with RE and RR.

**Table 4.7** MSMT17 dataset evaluation results with RE and RR.

Backbone	Rank-1	Rank-5	Rank-10	mAP
ResNet-50 + RE	75.94	88.77	91.56	51.66
ResNet-50 + RE + RR	80.29	91.50	93.38	59.29
DenseNet-121+ RE	76.34	89.47	91.26	52.26
DenseNet-121+ RE+ RR	82.90	93.90	95.28	61.59



**Figure 4.9** CMC curves on the MSMT17 dataset with RE and RR.

After adding the RE, the baseline model based on the ResNet-50 backbone gained improvements of 2.5% on Rank-1 and 4.73% on mAP with the Market-1501 dataset, 0.26% on Rank-1 and 1.9% on mAP with the DukeMTMC-reID dataset, and 8.0% on Rank-1 and 6.0% on mAP with the MSMT17 dataset. Likewise, the baseline model based on the DenseNet-121 gained improvements of 2.2% on Rank-1 and 4.9% on mAP with the Market-1501 dataset, 0.60% on Rank-1 and 2.9% on mAP with the DukeMTMC-reID dataset, and 10.1% on Rank-1 and 8.0% on mAP with the MSMT17 dataset. This proved that RE was an effective data augmentation technique.

After the RR procedure, the performance of the baseline model based on the ResNet-50 improved by an amount of 0.15% on Rank-1 and 11.51% on mAP with the Market-1501 dataset, 6.28% on Rank-1 and 17.69% on mAP with the DukeMTMC-reID dataset, and 4.35% on Rank-1 and 7.63% on mAP with the MSMT17 dataset.

Likewise, the performance of the baseline model based on the DenseNet-121 improved by an amount of 0.3% on Rank-1 and 11.6% on mAP with the Market-1501 dataset, 6.5% on Rank-1 and 18.7% on mAP with the DukeMTMC-reID dataset, and 9.33% on Rank-1 and 7.63% on mAP with the MSMT17 dataset. From these results, it can be concluded that the RR procedure had an effective role in improving the performance of the person Re-ID task, especially on the mAP, because it effectively improved the recall. For a visual presentation of the RR technique's efficiency, a random query image was chosen, and its Rank-10 gallery images prior to the application of RR are given in Figure 4.10. The original ranking results had already achieved good results, and RR improved it further.



**Figure 4.10** Example of rank-10 results before and after applying RR.

Finally, the proposed baseline model's final performance evaluation results on the Market-1501 and DukeMTMC-reID datasets were compared with the DL approaches, which used the pre-trained models as their backbone. Since it was recently released, no reported results exist on the MSMT17 dataset with these approaches. The comparison is shown in Table 4.8 and was made utilizing the CMC Rank-1 accuracy and mAP in the single query mode.

**Table 4.8** Comparison of the baseline performance with the state-of-the-art DL approaches.

Approach	Backbone	Market-1501		DukeMTMC-reID	
		Rank-1	mAP	Rank-1	mAP
IDE [12]	ResNet-50	73.9	47.8	65.2	45.0
SVDNet [130]	ResNet-50	82.3	62.1	76.7	56.8
DaRe [39]	ResNet-50	86.4	69.3	75.2	57.4
AlignedReID [166]	ResNet-50	90.6	77.7	81.2	67.4
DuATM [167]	DenseNet-121	91.4	76.6	81.8	64.6
Good practices [129]	ResNet-50	91.7	78.8	83.4	68.8
	DenseNet-121	92.5	79.8	83.5	68.5
BFE [168]	ResNet-50	94.4	85.0	88.7	75.1
MGN [140]	ResNet-50	95.7	86.9	88.7	78.4
<b>Ours</b>	ResNet-50	97.3	90.7	85.9	82.1
	DenseNet-121	97.9	91.8	88.2	85.0

From Table 4.8, the proposed baseline model significantly outperformed all of the classic DL approaches, including the IDE, SVDNet, and DaRe. It was far superior to many of the state-of-the-art methods, including AlignedReID and DuATM. It was also superior to the most recent BFE and MGN methods on the Market-1501 dataset and gave a comparable performance on the DukeMTMC-reID dataset. In addition, on MSMT17, Rank-1 was 80.29%, and mAP was 59.29%, which was very competitive. In conclusion, the performance evaluation results of the proposed baseline model, as well as the comparison that was presented in Table 4.8, demonstrated both the effectivity and superiority of the proposed baseline model for the task of person Re-ID.

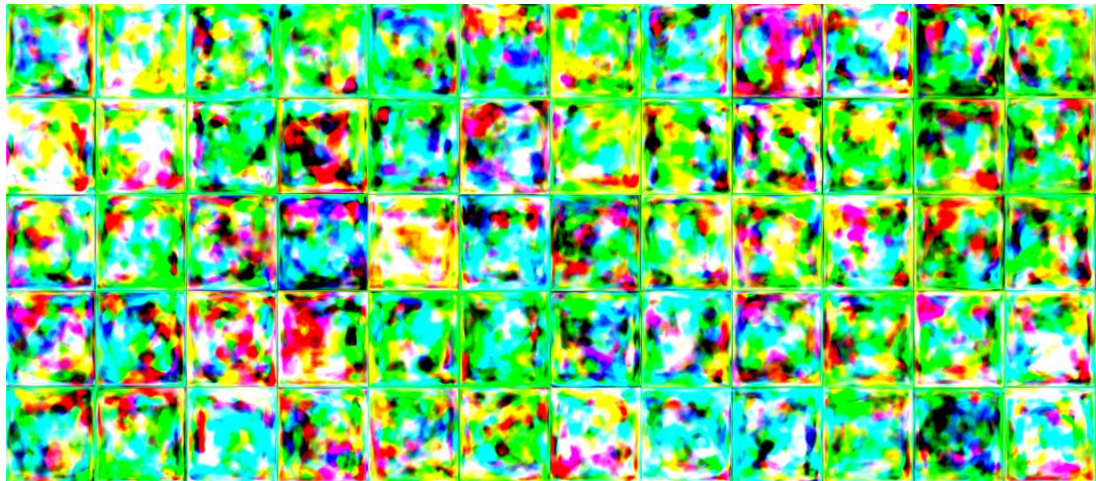
## 4.2 StyleGAN Training and Evaluation

StyleGAN was trained on each of the 3 selected datasets, Market-1501, DukeMTMC-reID, and MSMT17, separately, for 465 GPU hours using a 1 NVIDIA Tesla K80 GPU. Only the training sets were used for the training process. The training set of Market-1501 consisted of 12,936 images for 751 identities, the training set of DukeMTMC-reID contained 16,522 training images for 702 identities, and the training set of MSMT17 contained 32,621 training images for 1041 identities. This section

presents the results of the training and evaluation of the proposed StyleGAN model. First, the results of the StyleGAN training are presented by presenting the generated synthetic images during the different stages of the training process and then providing a qualitative and quantitative evaluation of the synthetic images generated by the trained generator.

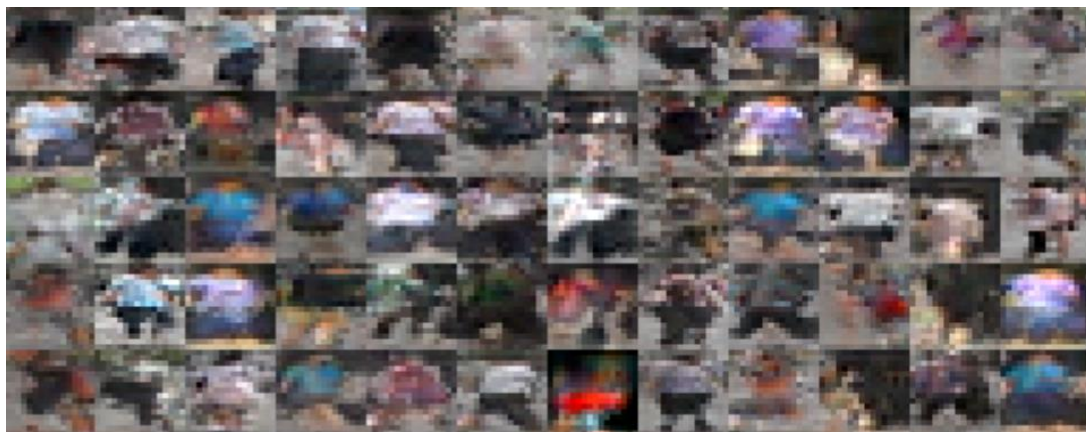
#### 4.2.1 StyleGAN Training Results

When training the StyleGAN with the configurations described in Section 3.5.4 and according to the person Re-ID datasets, the generator model initially started off by generating random noise images with a shallow resolution of  $8 \times 8$  and eventually building its way up to a final resolution of  $256 \times 256$ , which was actually the resolution of the images in the real person Re-ID dataset. Within the training time, the model generated the adapted fake image related to the dataset used. The figures below show some outputs obtained from the StyleGAN generator during the training process using the Market-1501 dataset. Each figure represents an example of a generator output at each resolution.



**Figure 4.11** Samples of initial noise images that were generated using the StyleGAN generator during training on the Market-1501 dataset.





**Figure 4.12** Samples of images that were generated using the StyleGAN generator at a resolution of  $8 \times 8$  during training on the Market-1501 dataset.



**Figure 4.13** Samples of images that were generated using the StyleGAN generator at a resolution of  $16 \times 16$  during training on the Market-1501 dataset.



**Figure 4.14** Samples of images that were generated using the StyleGAN generator at a resolution of  $32 \times 32$  during training on the Market-1501 dataset.





**Figure 4.15** Samples of images that were generated using the StyleGAN generator at a resolution of  $64 \times 64$  during training on the Market-1501 dataset.



**Figure 4.16** Samples of images that were generated using the StyleGAN generator at a resolution of  $128 \times 128$  during training on the Market-1501 dataset.



**Figure 4.17** Samples of images that were generated using the StyleGAN generator at a resolution of  $256 \times 256$  during training on the Market-1501 dataset.



The images of the people shown in the figures above were shortened from a long training process that took about 3 weeks. After the end of the StyleGAN training process, only the trained generator was made of use for the generation of synthetic images. A 512-dimensional random vector ( $z$ ) was input over the range  $[-1, 1]$ , as well as the Gaussian distribution of the trained StyleGAN generator, which was transformed into the intermediate vectors ( $w$ ) using the mapping network. The average quality of the generated images can be further improved by drawing the latent vectors from a truncated sampling space as described in Section 3.4.5. All of the generated images had a size of  $256 \times 256$ , and were used in the training of the baseline model with the LSRO algorithm.

#### 4.2.2 StyleGAN Evaluation Results

Before using the StyleGAN generated images for evaluating the final Re-ID performance of the proposed baseline model, the diversity and quality of the newly-generated images were assessed first, to be able to confirm if they were suitable to improve the performance of the person Re-ID task. The synthetic images that were generated by the StyleGAN generator were qualitatively and quantitatively evaluated by using the evaluation settings described in Section 3.5.3.

**Qualitative evaluations:** First, the quality of the images that were generated was evaluated visually. Figures 4.18 and 4.19 exhibit some samples of the synthetic images that were generated by the StyleGAN generator model using the Market-1501 and DukeMTMC-reID datasets. The images that are presented in these 2 figures provide a visual demonstration showing that StyleGAN was able to consistently generate both sensible and varied images using many different datasets.



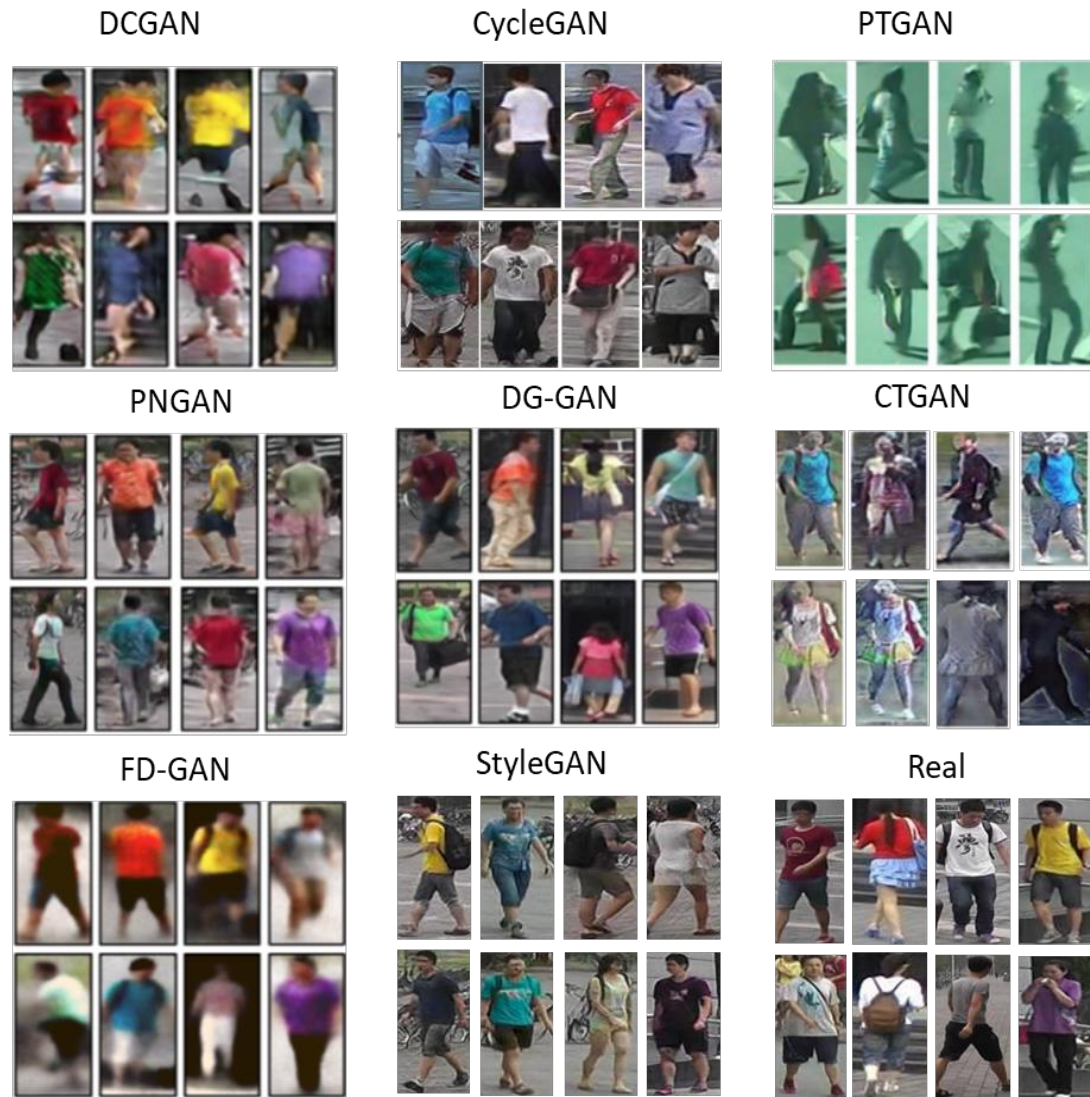
**Figure 4.18** Sample images generated by the StyleGAN generator trained on the Market-1501 dataset.



**Figure 4.19** Sample images generated by the StyleGAN generator trained on the DukeMTMC-reID dataset.

Next, a visual comparison of the samples that were generated by the StyleGAN approach was performed using the samples of the real images and that those that were generated via the use of other generative approaches that have been specifically designed for use in the person Re-ID task, in order to show the high quality of the images that were generated by StyleGAN. Figure 4.20 presents a visual comparison of some of the samples that were generated by StyleGAN, as well as some samples that were taken from the real dataset, in addition to groups of samples that were generated by the other generative methods that were specifically designed for the person Re-ID task, such as DCGAN [50], CycleGAN [51], PTGAN [52], PNGAN [53], FD-GAN [54], CTGAN [55] and DG-GAN [56]. These images were all chosen or generated from the Market-1501 dataset.

In Figure 4.20, it should be mentioned that the images that were generated by previous generative approaches were still quite poor with regard to quality, and they also have a great deal of noise. Even the images that are visually comparatively acceptable have a quite noticeable blurring and flaws, especially in the background. On the other hand, the synthetic images that were generated using the StyleGAN model had an appearance that was closer to that of the real images and also had a visual appearance that was much better in the foreground and in the background.



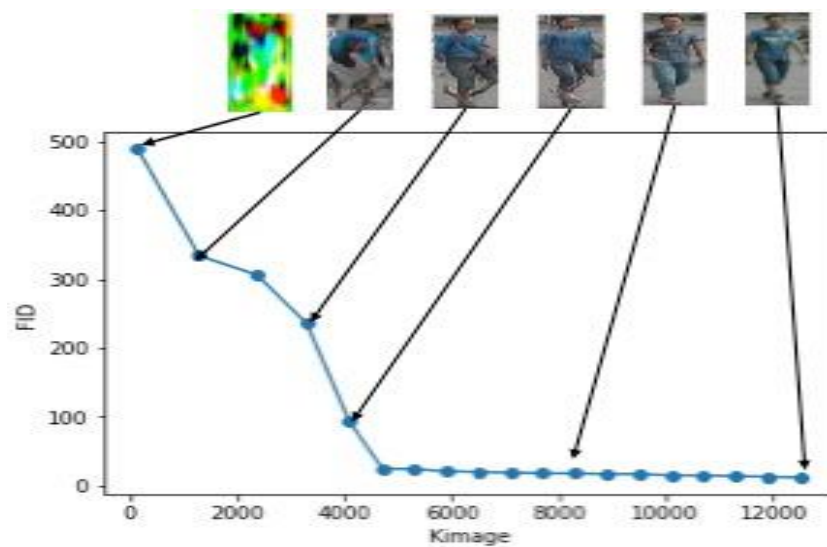
**Figure 4.20** Visual comparison of the images generated with the StyleGAN model with those generated with various state-of-the-art person generation methods using the Market-1501 dataset and the real ones, concentrating on the image backgrounds and the foregrounds.

**Quantitative evaluations:** To confirm the above qualitative assessments, the FID score and SSIM were measured to quantify the diversity and realism of StyleGAN's synthetically produced samples. It can easily be seen in Table 4.9 that the StyleGAN-generated images surpassed those that were generated by the other generative methods with regard to their diversity, realism, and large margins, which confirmed that the StyleGAN model was suitable for producing images of excellent quality. The higher SSIM that was obtained as a result of the various poses, backgrounds, occlusions, and so on, was obtained as the result of style mixing and stochastic variation. Moreover,

FID was used during the training process in the assessment of StyleGAN's performance via the measurement of the realism observed in the images that were generated. Figure 4.21 presents how the FID score improved throughout the training on the Market-1501 dataset.

**Table 4.9** Comparison of the FID and SSIM for the Market-1501 dataset images, both real and the generated. The lowest FID score indicated better quality, whereas the highest SSIM meant that there was more variety within the generated images.

Method	FID	SSIM
Real	7.22	0.350
FD-GAN [54]	257.00	0.247
PG <sup>2</sup> -GAN [105]	151.16	-
DCGAN [50]	136.26	-
PN-GAN [53]	54.23	0.335
DG-GAN [56]	18.24	0.360
<b>StyleGAN</b>	<b>12.67</b>	<b>0.387</b>



**Figure 4.21** FID score that was taken while training the StyleGAN model using the Market-1501 dataset.

According to the results, the StyleGAN model was capable of synthesizing person images that were very much like the real images, that had the correct attributes in both the foreground and background. Its effectiveness in the person Re-ID task is investigated further in the sections below.

### 4.3 Re-identification Evaluation using Expanded Datasets

To show the effectiveness of the images generated using the StyleGAN model in improving the efficiency of the proposed deep baseline model for the task of person Re-ID, and their generalizability, the images that were generated were added to the real image training set, and the LSRO regularization function was used to train the baseline model. All of the experiments were implemented according to Section 3.5.4, using only the ResNet-50-based model. The experiments were carried out using the Market-1501, DukeMTMC-reID, and MSMT17 datasets, after the images that were generated using the StyleGAN were added to the training sets in each of them. The performance was evaluated according to the evaluation metrics explained in Section 3.5.3.

#### 4.3.1 Baseline Evaluation using Different Numbers of StyleGAN Images

To fully explore the effectiveness of StyleGAN generated images, first, an evaluation was conducted to determine the effect of the number of StyleGAN images on the Re-ID performance. It was expected that increasing the size of the training set via the addition of the synthetic images will result in an improvement in the baseline model's performance. However, an observation was made that, when too few synthetic images were added to the original training set, the ability of LSRO to regularize became unsatisfactory. Contrarily, when too many synthetic images were added, there was a tendency wherein the learning procedure converged toward a uniform likelihood of prediction, and this was noticed for all of the training samples, which was an undesirable result. Thus, it was concluded that it was necessary for there to be a trade-off in order to prevent the unsatisfactory regularization of uniform label distributions. Therefore, the Re-ID performance evaluation was performed via the addition of 6,000, 8,000, 12,000, 18,000, 24,000, and 30,000 synthetic images to the training set. On the Market-1501 dataset, and since it supports multi-query mode, the findings of the Re-ID evaluation in single- and multi-query mode are presented in Tables 4.10 and Table 4.11, respectively. Moreover, the findings of the Re-ID evaluation on the DukeMTMC-reID dataset are presented in Table 4.12.

**Table 4.10** Performance improvement contributions (%) via adding the different numbers of StyleGAN images to the Market-1501 dataset under a single query mode.

No. of StyleGAN Images	Single query				Single query + RR			
	Rank-1	Rank-5	Rank-10	mAP	Rank-1	Rank-5	Rank-10	mAP
0 (Basel.)	97.32	98.48	98.98	79.23	97.17	98.66	99.03	90.74
6000	98.30	99.19	99.49	80.19	97.83	99.10	99.31	91.22
8000	98.42	99.28	99.52	80.92	97.86	99.13	99.34	91.32
12000	98.48	99.40	99.55	81.21	97.92	99.22	99.40	91.37
<b>18000</b>	<b>98.57</b>	<b>99.46</b>	<b>99.61</b>	<b>81.88</b>	<b>98.18</b>	<b>99.28</b>	<b>99.63</b>	<b>91.86</b>
24000	98.54	99.31	99.43	81.34	97.71	99.02	99.37	91.42
30000	98.33	99.28	99.31	80.77	97.50	98.99	99.22	90.75
<b>Improvement</b>	<b>+1.25</b>	<b>+0.98</b>	<b>+0.63</b>	<b>+2.65</b>	<b>+1.01</b>	<b>+0.62</b>	<b>+0.6</b>	<b>+1.12</b>

**Table 4.11** Performance improvement contributions (%) via the addition of the different StyleGAN image numbers to the Market-1501 dataset under a multi-query mode.

No. of StyleGAN Images	Multi query			
	Rank-1	Rank-5	Rank-10	mAP
0 (Basel.)	98.32	99.01	99.19	86.38
6000	98.99	99.40	99.52	86.94
8000	99.02	99.49	99.61	87.64
12000	99.04	99.52	99.64	87.68
<b>18000</b>	<b>99.16</b>	<b>99.58</b>	<b>99.64</b>	<b>88.02</b>
24000	99.07	99.58	99.61	87.71
30000	98.84	99.52	99.58	87.11
<b>Improvement</b>	<b>+0.84</b>	<b>+0.57</b>	<b>+0.45</b>	<b>+1.64</b>

**Table 4.12** Performance improvement contributions (%) via the addition of the different StyleGAN image numbers to the DukeMTMC-reID dataset under a single query mode.

No. of StyleGAN Images	Single query				Single query + RR			
	Rank-1	Rank-5	Rank-10	mAP	Rank-1	Rank-5	Rank-10	mAP
0 (Basel.)	79.62	89.72	92.81	63.76	85.90	91.33	93.76	82.19
6000	80.56	91.11	93.62	64.54	85.95	92.14	93.80	82.37
8000	80.74	90.52	93.13	64.90	86.66	92.59	94.47	83.38
12000	81.78	90.88	93.58	65.84	86.49	92.36	94.16	83.46
<b>18000</b>	<b>81.86</b>	<b>91.16</b>	<b>93.80</b>	<b>66.05</b>	<b>87.02</b>	<b>92.54</b>	<b>94.65</b>	<b>83.88</b>
24000	81.41	90.79	93.17	65.40	86.13	91.78	94.29	82.32
30000	81.05	90.35	93.08	64.92	85.81	91.65	93.89	82.23
<b>Improvement</b>	<b>+2.24</b>	<b>+1.44</b>	<b>+0.99</b>	<b>+2.29</b>	<b>+1.12</b>	<b>+1.21</b>	<b>+0.89</b>	<b>+2.69</b>

According to the findings in Tables 4.10, 4.11, and 4.12, the performance of the baseline model showed improvements with regard to both the CMC accuracy and the mAP as a result of introducing the different numbers of StyleGAN samples, and also was able to achieve peak performance using 18,000 StyleGAN images in the 2 datasets. When a comparison was made with the LSRO results that were reported previously by Zheng et al. [50], in which the peak performance was obtained via the addition of 24,000 DCGAN images. The baseline model herein only needed 18,000 StyleGAN images for it to be able to attain peak performance. Such an early convergence was the result of the high-quality StyleGAN images that were used. Moreover, it was observed that when too many were added, that is, >18,000 StyleGAN images, in training, the performance slumped because the model likely converged toward the synthetic data instead of the real data. Therefore, all of the comparisons were performed using the findings that were obtained using 18,000 StyleGAN samples.

The performance of the proposed baseline model was also evaluated on the MSMT17 dataset by adding 18,000 StyleGAN samples. The findings of the Re-ID evaluation are presented in Table 4.13.

**Table 4.13** Performance improvement contributions (%) via the addition of the different StyleGAN image numbers to the MSMT17 dataset under a single query mode.

No. of StyleGAN Images	Single query				Single query + RR			
	Rank-1	Rank-5	Rank-10	mAP	Rank-1	Rank-5	Rank-10	mAP
0 (Basel.)	75.94	88.77	91.56	51.66	80.29	91.50	93.38	59.29
<b>18000</b>	<b>78.32</b>	<b>89.98</b>	<b>92.59</b>	<b>54.13</b>	<b>81.57</b>	<b>92.61</b>	<b>94.25</b>	<b>60.96</b>
<b>Improvement</b>	<b>+2.38</b>	<b>+1.21</b>	<b>+1.03</b>	<b>+2.47</b>	<b>+1.28</b>	<b>+1.11</b>	<b>+0.87</b>	<b>+1.67</b>

The StyleGAN-generated images improved the deep baseline model. The tables above showed that utilizing 18,000 StyleGAN images in the training provided overall efficacy of the baseline model was dramatically improved for the person Re-ID. The Market-1501 dataset gained improvements of 1.25%, 0.98%, 0.63%, and 2.65%, respectively, in single-query mode, and 0.84%, 0.57%, 0.45%, and 1.64%, respectively, in multi-query mode, on Rank-1, -5 and -10 of the CMC curve and mAP. Also noticed were improvements of 1.01%, 0.62%, 0.60%, and 1.12%, respectively,



on Rank-1, -5, and -10 of the CMC curve and mAP with RR. The DukeMTMC-reID dataset gained improvements of 2.24%, 1.44%, 0.99%, and 2.29%, respectively, on Rank-1, -5 and -10 of the CMC curve and mAP. With RR, also noticed were improvements of 1.12%, 1.21%, 0.89%, and 2.69%, respectively, on Rank-1, -5, and -10 of the CMC curve and mAP. The MSMT17 dataset gained improvements of 2.38%, 1.21%, 1.03%, and 2.47%, respectively, on Rank-1, -5, and -10 of the CMC curve and mAP. With RR, also noticed were improvements of 1.28%, 1.11%, 0.87%, and 1.67%, respectively, on Rank-1, -5, and -10 of the CMC curve and the mAP.

Such findings were able to prove that the high-quality synthetic data that was produced using StyleGAN had the ability to successfully improve the performance of the person Re-ID baseline when it was used in conjunction with LSRO regularization.

#### **4.3.2 Baseline Model Results Comparison and Analysis**

To conduct a further assessment of the proposed method and be able to show the usefulness of the proposed baseline model, the model's performance was compared using the StyleGAN-generated data with the other state-of-the-art methods that have also used the synthetic data that was generated by the DCGAN model, such as that reported by [50] and [148]. In these methods, the baseline models shared some similarities with the baseline model designed herein, in that they were also built based on the ResNet-50 model, and they also adopted LSRO for adding the generated data to the training set. However, they did differ in the fact that they used the synthetic data that was generated by the DCGAN model. A comparison was made on the Market-1501 and DukeMTMC-reID datasets and presented in Table 4.14 and Table 4.15, respectively. The comparisons showed that the baseline model proposed herein with StyleGAN data provided encouraging results.

It was observed that using the Market-1501 dataset in single-query mode allowed the proposed baseline model to outperform the other models by 14.60% and 15.81%, respectively, on Rank-1 accuracy and mAP. Contrarily, in the multi-query mode, it exceeded the other models by 10.74% and 11.92%, respectively, on Rank-1 accuracy and mAP. Using the DukeMTMC-reID dataset, the baseline model was able to achieve an improvement of 14.18% on Rank-1 accuracy and 18.92% on mAP accuracy. Such



an improvement may have been the result of the strength of the architectural design of the model and the effect that resulted from the use of the synthetic images that were generated by StyleGAN. It was concluded that the proposed baseline model was able to boost the total efficiency of the person Re-ID task, and thus, it can be applied in practical surveillance applications.

**Table 4.14** Comparison of the baseline model that was proposed herein with the baseline models in [50] and [148] on the Market-1501 dataset. Rank-1 accuracy and mAP under single query mode are listed without RR.

Method	Single query		Multi query	
	Rank-1	mAP	Rank-1	mAP
DCGAN data + baseline model in [50]	83.97	66.07	88.42	76.10
DCGAN data + baseline model in [148]	88.63	74.95	91.42	79.87
<b>StyleGAN data + our baseline model</b>	<b>98.57</b>	<b>81.88</b>	<b>99.16</b>	<b>88.02</b>
<b>Improvement</b>	<b>+14.60</b>	<b>+15.81</b>	<b>+10.74</b>	<b>+11.92</b>

**Table 4.15** Comparison of the baseline model that was proposed herein with the baseline models in [50] and [148] on the DukeMTMC-reID dataset. Rank-1 accuracy and mAP under single query mode are listed without RR.

Method	Rank-1	mAP
DCGAN data + baseline model in [50]	67.68	47.13
DCGAN data + baseline model in [148]	76.53	60.79
<b>StyleGAN data + our baseline model</b>	<b>81.86</b>	<b>66.05</b>
<b>Improvement</b>	<b>+14.18</b>	<b>+18.92</b>

### 4.3.3 Re-Identification Qualitative Analysis

To give a visual presentation of the effectiveness of the proposed method, Figure 4.22 presents some of the query images that were randomly chosen, as well as their top-20 retrieved images from the gallery set of the Market-1501, which were arranged from left to right based on their similarity scores with the query. It should be noted that the proposed method was able to work accurately for Re-ID persons, despite variations in the lighting, poses, or background, which confirmed the robustness of the proposed person Re-ID model. It should also be noted that the situations of failure were the result of confusion between people who had similar clothing, which is considered as the major challenge in a person Re-ID task.



**Figure 4.22** Top 20 retrieved results for specific queries belonging to the Market 1501 dataset via the application of the proposed method. Queries are shown in the column furthest to the left, while the images retrieved from the gallery are given in order, from left to right, based on the similarity scores. The letters T (in green) and F (in red) at the top of each image indicate the true and false equivalents.

#### 4.3.4 Comparison with State-Of-The-Art Methods

A comparison was performed between the proposed StyleGAN-LSRO method, and the best state-of-art methods currently being used on the 3 benchmark person Re-ID datasets, namely Market-1501, DukeMTMC-reID, and MSMT17. The comparative methods were divided into 2 groups based on whether the data that was generated was used or not. For a reasonable comparison, only the best single-query mode results for Rank-1 accuracy and mAP were used. Table 4.16 shows a comparison of the proposed method and the state-of-the-art methods that did not use the generated data. In contrast, Table 4.17 shows a comparison with the state-of-the-art methods that did include the generated data.

**Table 4. 16** Comparison of the StyleGAN-LSRO method proposed herein with the existing state-of-the-art non-generative person Re-ID methods. The Rank-1 accuracy and mAP in single-query mode are given.

Method	Market-1501		DukeMTMC-reID		MSMT17	
	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
Verif-Identif [46]	79.5	59.8	-	-	-	-
OIM [133]	82.1	-	68.1	-	-	-
SVDNet [130]	82.3	62.1	76.7	56.8	-	-
PAN [131]	82.8	63.4	71.6	51.5	-	-
HA-CNN [147]	91.2	75.7	80.5	63.8	-	-
PCB [57]	93.8	81.6	83.3	69.2	-	-
Manacs [146]	93.1	82.3	84.9	71.8	-	-
IANet [169]	94.4	83.1	87.1	73.4	75.5	46.8
CAMA [143]	94.7	84.5	85.8	72.9	-	-
OSNet [170]	94.8	84.9	88.6	73.5	78.7	52.9
BAT-net [171]	95.1	87.4	87.7	77.3	79.5	56.8
Adaptive L2 Reg. [48]	95.3	88.3	88.9	79.9	79.6	59.4
SCAL [172]	95.8	89.3	88.9	79.1	-	-
RGA-SC [49]	96.1	88.4	-	-	80.3	57.5
<b>StyleGAN-LSRO (Ours)</b>	<b>98.5</b>	<b>91.8</b>	<b>87.0</b>	<b>83.8</b>	<b>81.5</b>	<b>60.9</b>

According to Table 4.16, when the non-generative approaches were compared, the proposed method was able to achieve competitive results, and it was also able to outperform most previous state-of-the-art methods. For the Market-1501 dataset, the proposed method actually exceeded the best result on rank-1 accuracy, which was reported for RGA-SC by a factor of 2.4%, as well as the best result on mAP accuracy reported for SCAL by 2.5%. For the DukeMTMC-reID dataset, the proposed method achieved Rank-1 accuracy, which was close to the best result on Rank-1 accuracy reported for Adaptive L2 Reg. by a factor of 1.9%, as well as the best result on mAP accuracy also reported for Adaptive L2 Reg. by a factor of 3.9%. For the MSMT17 dataset, the proposed method exceeded the best result on rank-1 accuracy reported for RGA-SC by a factor of 1.2%, as well as the best result on mAP accuracy reported for Adaptive L2 Reg. by a factor of 1.5%.

**Table 4.17** Comparison of the StyleGAN-LSRO method proposed herein with the existing state-of-the-art person Re-ID methods that use the data generated. Rank-1 accuracy and mAP in single-query mode are given.

Method	Market-1501		DukeMTMC-reID		MSMT17	
	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
CTGAN [55]	56.7	23.6	42.6	21.1	-	-
LSRO [50]	83.9	66.0	67.6	47.1	-	-
Multi-pseudo [173]	87.9	81.1	76.8	58.5	-	-
PN-GAN [53]	89.4	72.5	73.5	53.2	-	-
CamStyle [51]	89.4	71.5	78.3	57.6	-	-
FD-GAN [54]	90.5	77.7	80.0	64.5	-	-
SLSR [148]	91.5	88.1	82.6	79.2	-	-
DG-GAN [56]	94.8	86.0	86.6	74.8	77.2	52.3
<b>StyleGAN-LSRO (Ours)</b>	<b>98.5</b>	<b>91.8</b>	<b>87.0</b>	<b>83.8</b>	<b>81.5</b>	<b>60.9</b>

According to Table 4.17, when compared with the generative approaches, the proposed method outperformed all of the previous state-of-the-art methods by a clear margin when compared with the generative approaches. On the Market-1501 dataset, the proposed method achieved 98.5% Rank-1 accuracy and 91.8% mAP, exceeding the best result on Rank-1 accuracy reported for DG-GAN by a factor of 3.7% and exceeding the best result on mAP reported for SLSR by a factor of 3.7%. On the DukeMTMC-reID dataset, the proposed method achieved an 87.0% Rank-1 accuracy and 83.8% mAP, exceeding the best result on Rank-1 accuracy reported for DG-GAN by a factor of 0.4%, as well as the best result on mAP reported for SLSR by a factor of 4.6%. Since it was recently released, not many reported results exist on the MSMT17 dataset, as shown in Table 4.17. However, the proposed method achieved an 81.5% Rank-1 accuracy and 60.9% mAP, exceeding the existing works. Compared to DG-GAN, the proposed method exceeded their results by a factor of 4.3% on Rank-1 accuracy and by a factor of 8.6% on mAP.

In this chapter, the experimental analysis and discussion were performed to evaluate the Re-ID performance of the proposed person Re-ID method. Three different public datasets, namely Market-1501, DukeMTMC-reID, and MSMT17, were used to test the performance of the proposed method. The findings showed the contribution of data

augmentation using StyleGAN synthesis in improving person Re-ID. The images generated by the StyleGAN model were at a resolution of  $256 \times 256$  and improved the results of the person Re-ID successfully. Thus, increasing the resolution of the generated images can lead to greater success in this task. Although the improvements achieved by the proposed method on the DukeMTMC-reID and MSMT17 datasets were not as significant as that achieved on the Market-1501, these were still competitive results. This matter can be attributed to the fact that the data normalization procedure for the 3 datasets was done according to the mean and standard deviation of the Market-1501 dataset. Obviously, it should be different from that. On the other hand, since the proposed baseline model used RGB images as the input, nearly all of the miss-ranked images were influenced by the gallery images, which looked very similar to the query image. The people usually wore clothes of the same or similar colors, which was visually confusing.

# CHAPTER 5

## CONCLUSION AND FUTURE WORK

### 5.1 Conclusion

Person Re-ID has quickly gained its place among the most valuable and most helpful approaches for the identification of human subjects in today's video surveillance systems. This modality involves the recognition of individuals as captured by networked video surveillance cameras that may have possible fields of view that do not overlap. The main challenges that make this approach difficult are variations in lighting, different viewpoints, occlusion, changes in individuals' poses, and background chaos. Recently, DL methods have boosted the performances of person Re-ID systems, diminishing the effect of the aforementioned challenges to some extent. At the same time, however, they have added a new challenge in that these deep methods need considerably significant volumes of labeled data for training. The underlying motivation of this thesis has been to propose a new method for the enlargement of existing person Re-ID datasets that will permit the successful establishment of deep models for person Re-ID tasks that offer heightened power and efficacy.

This thesis has developed an improved method called the StyleGAN-LSRO method to improve the quality of the outcome of person Re-ID tasks. This method utilizes the StyleGAN model for the generation of novel images, surpassing the generative approaches previously suggested in the literature in terms of both quality and variety. At the same time, LSRO has been applied in the present work for assigning virtual labels to the aforementioned novel images generated with the StyleGAN-LSRO method. It then mixes those images with real training images, which are also labeled, with the aim of training an improved CNN baseline model and learning the relevant discriminative features. In this process, RE and RR have been used in conjunction with the suggested baseline model to generate additional performance improvement. Three benchmark datasets for person Re-ID, namely Market-1501, DukeMTMC-reID, and

MSMT17, were used for the evaluation of the efficiency of the newly established method. The performance of this novel method was evaluated systematically by applying a well-developed evaluation benchmark. The experimental results conducted on real datasets and a mix of synthetic and real datasets illustrated that the proposed method could deal with person Re-ID tasks with competitive performance. However, the proposed method can still be improved. For example, the heavy reliance on RGB information might be considered a shortcoming, and the inconsistent performances on different datasets should ideally be addressed, although these limitations are shared by other leading approaches.

The following are the contributions made by this dissertation, as originally highlighted in Section 1.3.

1. **Proposing a successful baseline model to extract strong discriminative features to be utilized in tasks of person Re-ID by modifying the general CNN model developed for object recognition and then fine-tuning it using the transfer learning approach to make it more suitable for the person Re-ID problem domain. Besides, RE and RR are combined with the proposed baseline model to further achieve a significant performance improvement and avoid overfitting.** The proposed baseline model is presented in Section 3.2 in detail. The experimental analysis and the discussion offered in Chapter 4, Section 4.1, have confirmed the superior results and effectivity of this novel baseline model for tasks of person Re-ID.
2. **Applying, for the first time to the best of our knowledge, a StyleGAN model as a generative model for the generation of images of individuals that are of high quality while also being notably diverse, working with input from the person Re-ID datasets that are currently freely available in the relevant literature. The images produced in this process are subsequently employed for the expansion of the training sets with the introduction of increased variation of background designs, colors, illuminations, and types of poses.** A StyleGAN model was presented in Section 3.3 in detail. As shown in Chapter 4, Section 4.2.2, the conducted experiments, and the qualitative and quantitative evaluations clearly showed

the efficiency of the StyleGAN in generating high-quality and highly diverse person images while utilizing the given person Re-ID datasets.

3. **Proposing the application of the LSRO method for integrating the images of individuals which was produced by StyleGAN into original labeled training images, giving those images a uniform label distribution, and defining a regularized loss function to be applied as a part of the training process.** This contribution was achieved in Chapter 3, in Section 3.4, where the proposed regularized loss function to train the proposed baseline model on real and generated data was implemented using LSRO. The experimental analysis and discussion offered in Chapter 4, Section 4.3, showed the value of performing data augmentation with the combined power of StyleGAN and LSRO for the improved outcome of person Re-ID tasks.
4. **Developing an evaluation protocol to measure the effectiveness of the proposed framework. This evaluation protocol includes both quantitative and qualitative evaluation processes to assess StyleGAN's ability to produce novel person images that are acceptable in terms of both diversity and quality. This work also follows the commonly used standard evaluation protocol, adopting the CMC curve together with mAP as the metrics selected for the careful evaluation of the performances of the person Re-ID baseline model, guaranteeing reliable comparisons between previous approaches and the novel strategy proposed in this work.** A protocol for evaluation was presented in Section 3.5.3, which includes evaluation metrics for assessing the performances of the baseline model in person Re-ID tasks with the chosen datasets. Prominent metrics to measure the extent of the quality and diversity of the images produced with StyleGAN have additionally been described here. The evaluation protocol guarantees reliable comparisons of the novel strategy suggested here and state-of-the-art approaches.



## 5.2 Future Work

For future work, a more experimental analysis could be conducted for further improvement of the performances of the proposed baseline model using an RGB-D camera to obtain RGB information and depth information at the same time. Moreover, it is possible to explore part-level features, jointly train the model with more than one dataset, and carefully consider mechanisms to calibrate misaligned images. In the novel strategy presented in the present dissertation, images produced with the application of the StyleGAN model had a resolution of  $256 \times 256$ , notably improving the final outcomes of person Re-ID tasks. Another possible future work will involve efforts to increase resolutions of images being produced; that may sequentially provide higher levels of success in the completion of such tasks. Moreover, evaluating the newly proposed StyleGAN model, the observed FID score is larger than the range of scores presented for the StyleGAN model for face generation. Possible future work can be done using larger person Re-ID datasets, approximately doubling the size of the current dataset, to better estimate FID. Also, in future research, taking into account the potential of StyleGAN, we can explore better label assigning strategies that will work better to assign a more suitable label distribution to the created images. Furthermore, the present research has mainly focused on short-term image-based person Re-ID in a relatively simple environment, ignoring person detection procedures. The proposed method also requires some time to perform, which will fail to fulfill the real-time requirements of real-world applications. Thus, future research must also try to explore video-based person Re-ID models, systems for real-time person Re-ID tasks, and the execution of person Re-ID tasks in complex environments.

# REFERENCES

- [1] Lavi, B., Fatan Serj, M., and Ullah, I., *Survey on Deep Learning Techniques for Person Re-Identification Task*. [Online]. Available: <https://arxiv.org/abs/1807.05284>., 2018
- [2] Bedagkar-Gala, A., and Shah, S. K., *A survey of approaches and trends in person re-identification*. Image and Vision Computing, **(32)**, (4) 270-286. 2014
- [3] Saghafi, M.A., Hussain, A., Zaman, H.B. and Saad, M.H.M., *Review of person re-identification techniques*. IET Computer Vision, **(8)**, (6) 455-474. 2014
- [4] Wu, D., Zheng, S., Zhang, X., Yuan, C., Cheng, F., Zhao, Y., Lin, Y., Zhao, Z., Jiang, Y., and Huang, D., *Deep learning-based methods for person re-identification: A comprehensive review*. Neurocomputing, **(337)**, 354-371. 2019
- [5] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., *Generative Adversarial Networks*, in *Proceedings of Advances in Neural Information Processing Systems (NIPS2014)*. Montreal, Canada. p. 241-258. 2014
- [6] Karras, T., Laine, S. and Aila, T., *A Style-Based Generator Architecture for Generative Adversarial Networks*, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA. p. 4396-4405. 2019
- [7] *The European Union. The general data privacy regulation*. (n.d.). Retrieved from <https://www.gdpr-info.eu/>, accessed 07-03-2019.
- [8] Solichin, A., Harjoko, A., and Putra, A., *A Survey of Pedestrian Detection in Video*. International Journal of Advanced Computer Science and Applications, **(5)**, (10) 41-47. 2014

- [9] Wu, Y., Lim, J., and Yang, M., *Online Object Tracking: A Benchmark*, in *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA. p. 2411-2418. 2013
- [10] Zajdel, W., Zivkovic, Z., and Krose, B. J. A., *Keeping Track of Humans: Have I Seen This Person Before?*, in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. Barcelona, Spain. p. 2081-2086. 2005
- [11] Gheissari, N., Sebastian, T. B., and Hartley, R., *Person Reidentification Using Spatiotemporal Appearance*, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. New York, NY, USA. p. 1528-1535. 2006
- [12] Zheng, L., Yang, Y., and Hauptmann A. G., *Person Re-identification: Past, Present and Future*. October 01, 2020, <https://arxiv.org/abs/1610.02984>. 2016
- [13] Wang, K., Wang, H., Liu, M., Xing, X., and Han, T., *Survey on person re-identification based on deep learning*. CAAI Transactions on Intelligence Technology, **(3)**, (4) 219-227. 2018
- [14] Almasawa, M.O., Elrefaei, L. A., and Moria, K., *A Survey on Deep Learning-Based Person Re-Identification Systems*. IEEE Access, **(7)**, 175228-175247. 2019
- [15] Wang, H., Du, H., Zhao, Y., and Yan, J., *A Comprehensive Overview of Person Re-Identification Approaches*. IEEE Access, **(8)**, 45556-45583. 2020
- [16] Kviatkovsky, I., Adam, A., and Rivlin, E., *Color Invariants for Person Reidentification*. IEEE Transactions on Pattern Analysis and Machine Intelligence, **(35)**, (7) 1622-1634. 2013
- [17] Zhao, R., Ouyang, W., and Wang, X., *Person Re-identification by Saliency Matching*, in *2013 IEEE International Conference on Computer Vision*. Sydney, NSW, Australia. p. 2528-2535. 2013

- [18] Bazzani, L., Cristani, M., and Murino, V., *Symmetry-driven accumulation of local features for human characterization and re-identification*. Computer Vision and Image Understanding, (117), (2) 130–144. 2013
- [19] Liao, S., Hu, Y., Xiangyu, Z., and Li, S. Z., *Person re-identification by Local Maximal Occurrence representation and metric learning*, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA. p. 2197-2206. 2015
- [20] Yang, Y., Yang, J., Yan, J., Liao, S., Yi, D., and Li, S. Z., *Salient Color Names for Person Re-identification*, in *13th European Conference on Computer Vision (ECCV)*. Zurich, Switzerland. p. 536-551. 2014
- [21] Zhao, R., Ouyang, W., and Wang, X., *Unsupervised Saliency Learning for Person Re-identification*, in *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA. p. 3586-3593. 2013
- [22] Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., and Tian, Q., *Scalable Person Re-identification: A Benchmark*, in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile. p. 1116-1124. 2015
- [23] Ma, B., Su, Y., and Jurie, F., *BiCov: A novel image representation for person re-identification and face verification*, in *Proceedings of the British Machine Vision Conference (BMVC)*. Surrey, BC, Canada. p. 57.1-57.11. 2012
- [24] Xiong, F., Gou, M., Camps, O., and Sznaiier, M., *Person Re-Identification Using Kernel-Based Metric Learning Methods*, in *13th European Conference on Computer Vision (ECCV)*. Zurich, Switzerland. p. 1-16. 2014
- [25] Ojala, T., Pietikainen, M., and Maenpaa, T., *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. IEEE Transactions on Pattern Analysis and Machine Intelligence, (24), (7) 971-987. 2002
- [26] Lowe, D.G., *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, (60), (2) 91-110. 2004

- [27] Liao, S., and Li, S. Z., *Efficient PSD Constrained Asymmetric Metric Learning for Person Re-Identification*, in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile. p. 3685-3693. 2015
- [28] Fogel, I., and Sagi, D., *Gabor filters as texture discriminator*. *Biological Cybernetics*, **(61)**, (2) 103–113. 1989
- [29] Schmid, C., *Constructing models for content-based image retrieval*, in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Kauai, HI, USA. p. II-II. 2001
- [30] Gray, D., and Tao, H., *Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features*, in *Proceedings of the 10th European Conference on Computer Vision (ECCV08)*. Berlin, Heidelberg. p. 262-275. 2008
- [31] Köstinger, M., Hirzer, M., Wohlhart, P., Roth, P. M., and Bischof, H., *Large scale metric learning from equivalence constraints*, in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, USA. p. 2288-2295. 2012
- [32] Li, W., Zhao, R., Xiao, T. and Wang, X., *Human reidentification with transferred metric learning*, in *Proceedings of the 11th Asian Conference on Computer Vision (ACCV)*. Daejeon, Korea. p. 31-44. 2012
- [33] Li, Z., Chang, S., Liang, F., Huang, T. S., Cao, L., and Smith, J. R., *Learning Locally-Adaptive Decision Functions for Person Verification*, in *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA. p. 3610-3617. 2013
- [34] Pedagadi, S., Orwell, J., Velastin, S., and Boghossian, B., *Local Fisher Discriminant Analysis for Pedestrian Re-identification*, in *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, USA. p. 3318-3325. 2013

- [35] Zheng, W., Gong, S., and Xiang, T., *Reidentification by Relative Distance Comparison*. IEEE Transactions on Pattern Analysis and Machine Intelligence, (35), (3) 653-668. 2013
- [36] Prosser, B.J., Zheng, W. S., Gong, S., Xiang, T., and Mary, Q., *Person re-identification by support vector ranking*, in *Proceedings of the British Machine Vision Conference (BMVC)*. Aberystwyth, UK. p. 21.1-21.11. 2010
- [37] Ahmed, E., Jones, M., and Marks, T. K., *An improved deep learning architecture for person re-identification*, in *Proceedings of 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA. p. 3908-3916. 2015
- [38] Chen, W., Chen, X., Zhang, J., and Huang, K., *A multi-task deep network for person re-identification*, in *Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. San Francisco, California, USA. p. 3988-3994. 2017
- [39] Cheng, D., Gong, Y., Zhou, S., Wang, J., and Zheng, N., *Person re-identification by multi-channel parts-based CNN with improved triplet loss function*, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, USA. p. 1335-1344. 2016
- [40] Geng, M., Wang, Y., Xiang, T., and Tian, Y., *Deep Transfer Learning for Person Re-identification*. November 01, 2020, <https://arxiv.org/abs/1611.05244>. 2016
- [41] Li, D., Chen, X., Zhang, Z. and Huang, K., *Learning deep context-Aware features over body and latent parts for person re-identification*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA. p. 7398-7407. 2017
- [42] Li, W., Zhao, R., Xiao, T., and Wang, X., *DeepReID: Deep filter pairing neural network for person re-identification*, in *Proceedings of the IEEE Computer*

*Society Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA. p. 152-159. 2014

- [43] Wu, L., Shen, C., and Hengel, A., *PersonNet: Person Re-identification with Deep Convolutional Neural Networks*. January 01, 2021, <https://arxiv.org/abs/1601.07255>. 2016
- [44] Wu, S., Chen, Y.C., Li, X., Wu, A.C., You, J.J., and Zheng, W.S. , *An Enhanced Deep Feature Representation for Person Re-identification*, in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Lake Placid, NY, USA. p. 1-8. 2016
- [45] Xiao, T., Li, H., Ouyang, W., and Wang, X., *Learning Deep Feature Representations with Domain Guided Dropout for Person Re-identification*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA. p. 1249-1258. 2016
- [46] Zheng, Z., Zheng, L., and Yang, Y., *A Discriminatively Learned CNN Embedding for Person Reidentification*. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, **(14)**, (1) Article 13. 2017
- [47] Zhong, W., Jiang, L., Zhang, T., Ji, J., and Xiong, H., *Combining multilevel feature extraction and multi-loss learning for person re-identification*. *Neurocomputing*, **(334)**, 68-78. 2019
- [48] Ni, X., Fang, L., and Huttunen, H. , *Adaptive L2 Regularization in Person Re-Identification*. December 01, 2020, <https://arxiv.org/abs/2007.07875>. 2020
- [49] Zhang, Z., Lan, C., Zeng, W., Jin, X., and Chen, Z., *Relation-Aware Global Attention for Person Re-Identification*, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA. p. 3183-3192. 2020
- [50] Zheng, Z., Zheng, L., and Yang, Y., *Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in Vitro*, in *Proceedings of the*

- IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy. p. 3774-3782. 2017
- [51] Zhong, Z., Zheng, L., Zheng, Z., Li, S., and Yang, Y., *Camera Style Adaptation for Person Re-identification*, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA. p. 5157-5166. 2018
  - [52] Wei, L., Zhang, S., Gao, W., and Tian, Q., *Person Transfer GAN to Bridge Domain Gap for Person Re-identification*, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA. p. 79-88. 2018
  - [53] Qian, X., Fu, Y., Xiang, T., Wang, W., Qiu, J., Wu, Y., Jiang, Y., and Xue, X., *Pose-normalized image generation for person re-identification*, in *Proceedings of the European conference on computer vision (ECCV)*. Munich, Germany. p. 650-667. 2018
  - [54] Yixiao, G., Zhuowan, L., Haiyu, Z., Guojun, Y., Shuai, Y., Xiaogang, W., and Hongsheng L., *FD-GAN: Pose-guided Feature Distilling GAN for Robust Person Re-identification*, in *Conference on Advances in Neural Information Processing Systems*. Montreal, CANADA. p. 1229--1240. 2018
  - [55] Zhou, S., Ke, M., and Luo, P., *Multi-camera transfer GAN for person re-identification*. *Journal of Visual Communication and Image Representation*, (59), 393-400. 2019
  - [56] Zheng, Z., Yang, X., Yu, Z., Zheng, L., Yang, Y., and Kautz, J., *Joint Discriminative and Generative Learning for Person Re-Identification*, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, United States. p. 2133-2142. 2019
  - [57] Sun, Y., Zheng, L., Yang, Y., Tian, Q., and Wang, S., *Beyond Part Models: Person Retrieval with Refined Part Pooling (and A Strong Convolutional Baseline)*, in *Proceedings of European Conference on Computer Vision (ECCV)*. Munich, Germany. p. 501-518. 2018



- [58] LeCun, Y., Bengio, Y., and Hinton, G., *Deep learning*. Nature, **(521)**, (7553) 436-444. 2015
- [59] Schmidhuber, J., *Deep learning in neural networks: An overview*. Neural Networks, **(61)**, 85-117. 2015
- [60] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P., *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, **(86)**, (11) 2278-2324. 1998
- [61] Hacibeyoglu, M., *Human Gender Prediction on Facial Mobil Images using Convolutional Neural Networks*. International Journal of Intelligent Systems and Applications in Engineering, **(3)**, 203-208. 2018
- [62] İnik, Ö., and Ülker, E. , *Deep Learning and Deep Learning Models Used in Image Analysis*. Journal of Gaziosmanpasa Scientific Research, **(6)**, (3) 85-104. 2017
- [63] He, K., Zhang, X., Ren, S., and Sun, J., *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, **(37)**, (9) 1904-1916. 2015
- [64] Sermanet, P., Chintala, S., and Lecun, Y., *Convolutional Neural Networks Applied to House Numbers Digit Classification*, in *21st International Conference on Pattern Recognition (ICPR 2012)*. Tsukuba, Japan. p. 3288-3291. 2012
- [65] Rawat, W., and Wang, Z., *Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review*. Neural Computation, **(29)**, (9) 2352-2449. 2017
- [66] Hodgkin, A.L., and Huxley, A. F., *A quantitative description of membrane current and its application to conduction and excitation in nerve*. The Journal of physiology, **(117)**, (4) 500-544. 1952

- [67] Krizhevsky, A., Sutskever, I., and Hinton, G. E., *ImageNet classification with deep convolutional neural networks*. Communications of the ACM, (**60**), (6) 84-90. 2017
- [68] Nair, V., and Hinton, G. E., *Rectified linear units improve restricted boltzmann machines*, in *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress. Haifa, Palestine. p. 807–814. 2010
- [69] Pedamonti, D., *Comparison of non-linear activation functions for deep neural networks on MNIST classification task*. [Online]. Available: <https://arxiv.org/abs/1804.02763>, 2018
- [70] Xu, B., Wang, N., Chen, T., and Li, M., *Empirical Evaluation of Rectified Activations in Convolutional Network*. [Online]. Available: <https://arxiv.org/abs/1505.00853>, 2015
- [71] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*. 3/5/2020, <https://www.deeplearningbook.org>. 2016
- [72] Kingma, D.P., and Ba, J., *Adam: A Method for Stochastic Optimization*, in *International Conference on Learning Representations (ICLR)*. San Diego, CA, USA. 2015
- [73] Claesen, M., and De Moor, B., *Hyperparameter Search in Machine Learning*. March 10, 2021, <https://ui.adsabs.harvard.edu/abs/2015arXiv150202127C>. 2015
- [74] Van Rijn, J.N., and Hutter, F., *Hyperparameter Importance Across Datasets*, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery. London, United Kingdom. p. 2367–2376. 2018
- [75] Krizhevsky, A., Sutskever, I., and Hinton, G. E., *ImageNet classification with deep convolutional neural networks*, in *Proceedings of the 25th International*

*Conference on Neural Information Processing Systems*. Lake Tahoe, Nevada, USA. p. 1097-1105. 2012

- [76] Simonyan, K., and Zisserman, A., *Very Deep Convolutional Networks for Large-Scale Image Recognition*. September 04, 2020, <https://arxiv.org/abs/1409.1556>. 2014
- [77] Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y., *Random Erasing Data Augmentation*. August 01, 2019, <https://arxiv.org/abs/1708.04896>. 2017
- [78] Hastie, T., Tibshirani, R., & Friedman, J. H., (Ed.). *The elements of statistical learning: data mining, inference, and prediction (2nd ed.)*. New York: Springer. 2009
- [79] Ioffe, S., and Szegedy, C., *Batch normalization: accelerating deep network training by reducing internal covariate shift*, in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. Lille, France. p. 448–456. 2015
- [80] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. *Journal of Machine Learning Research*, (15), 1929-1958. 2014
- [81] Gal, Y., and Ghahramani, Z., *A theoretically grounded application of dropout in recurrent neural networks*, in *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Barcelona, Spain. p. 1027–1035. 2016
- [82] He, K., Zhang, X., Ren, S., and Sun, J., *Deep Residual Learning for Image Recognition*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA. p. 770-778. 2016
- [83] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. , *Densely Connected Convolutional Networks*, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA. p. 2261-2269. 2017

- [84] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, W., *Rethinking the Inception Architecture for Computer Vision*, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA. p. 2818-2826. 2016
- [85] Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L., *ImageNet: A large-scale hierarchical image database*, in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Miami, FL, USA. p. 248-255. 2009
- [86] Abdulatif, S., Aziz, F., Armanious, K., Kleiner, B., Yang, B., and Schneider, U., *Person Identification and Body Mass Index: A Deep Learning-Based Study on Micro-Dopplers*, in *2019 IEEE Radar Conference (RadarConf)*. Boston, MA, USA. p. 1-6. 2019
- [87] Tan, T., Li, Z., Liu, H., Zanjani, F. G., Ouyang, Q., Tang, Y., Hu, Z., and Li, Q., *Optimize Transfer Learning for Lung Diseases in Bronchoscopy Using a New Concept: Sequential Fine-Tuning*. *IEEE Journal of Translational Engineering in Health and Medicine*, (6), 1-8. 2018
- [88] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., *Going deeper with convolutions*, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA. p. 1-9. 2015
- [89] Chang, W., Chen, L., Hsu, C., Lin, C., and Yang, T., *A Deep Learning-Based Intelligent Medicine Recognition System for Chronic Patients*. *IEEE Access*, (7), 44441-44458. 2019
- [90] Torrey, L., and Shavlik, J., Transfer learning. *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. (1st Edition) pp. (242–264). IGI Global. 2010
- [91] Pan, J., and Yang, Q., *A Survey on Transfer Learning*. *IEEE Transactions on Knowledge and Data Engineering*, (22), (10) 1345-1359. 2010

- [92] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S. Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., and Fei-Fei, L., *ImageNet Large Scale Visual Recognition Challenge*. International Journal of Computer Vision (IJCV), **(115)**, (3) 211–252. 2015
- [93] Shin, H.C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., and Summers, R. M., *Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning*. IEEE transactions on medical imaging, **(35)**, (5) 1285-98. 2016
- [94] Chen, H., Wang, Y., Shi, Y., Yan, K., Geng, M., Tian, Y., and Xiang, T., *Deep Transfer Learning for Person Re-Identification*, in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*. Xi'an, China. p. 1-5. 2018
- [95] Xiao, Q., Cao, K., Chen, H., Peng, F., and Zhang, C., *Cross Domain Knowledge Transfer for Person Re-identification*. March 16, 2021, <https://arxiv.org/abs/1611.06026>. 2016
- [96] Buda, M., Maki, A., and Mazurowski, M. A., *A systematic study of the class imbalance problem in convolutional neural networks*. Neural Networks, **(106)**, 249-259. 2018
- [97] López, V., Fernández, A., García, S., Palade, V., and Herrera, F., *An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics*. Information Sciences, **(250)**, 113-141. 2013
- [98] Fernández, A., García, S., Herrera, F., and Chawla, N. V., *SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary*. Journal of Artificial Intelligence Research, **(61)**, (1) 863–905. 2018

- [99] Kingma, D., and Welling, M., *Auto-Encoding Variational Bayes*, in *International Conference on Learning Representations (ICLR)*. Banff, Canada. p. 1-14. 2014
- [100] Gandhi, V., Chapter 2 - Interfacing Brain and Machine. *Brain-Computer Interfacing for Assistive Robotics*, V. Gandhi, Editor. pp. (7-63). Academic Press: San Diego. 2015
- [101] Kobyzev, I., Prince, S., and Brubaker, M., *Normalizing Flows: An Introduction and Review of Current Methods*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1-17. 2020
- [102] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X., *Improved techniques for training GANs*, in *Proceedings of the International Conference on Neural Information Processing Systems*. Barcelona, Spain. p. 2234–2242. 2016
- [103] Radford, A., L. Metz, and S. Chintala, *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. March 17, 2021, <https://arxiv.org/abs/1511.06434>. 2015
- [104] Arjovsky, M., Chintala, S., and Bottou, L., *Wasserstein GAN*. January 01, 2021, <https://arxiv.org/abs/1701.07875>. 2017
- [105] Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., and Van Gool, L., *Pose Guided Person Image Generation*. Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS). Long Beach, CA, USA. 405-415. 2017
- [106] Zhang, C., Wu, L., and Wang, Y., *Crossing generative adversarial networks for cross-view person re-identification*. Neurocomputing, (340), 259-269. 2019
- [107] Isola, P., Zhu, J., Zhou, T., and Efros, A. A., *Image-to-Image Translation with Conditional Adversarial Networks*, in *Proceedings of the IEEE Conference on*

*Computer Vision and Pattern Recognition (CVPR)*. Honolulu, Hawaii, USA. p. 5967-5976. 2017

- [108] Kim, T., Cha, M., Kim, H., Lee, G. L., and Kim, J., *Learning to Discover Cross-Domain Relations with Generative Adversarial Networks*, in *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Sydney, Australia. p. 1857--1865. 2017
- [109] Arjovsky, M., Chintala, S., and Bottou, L., *Wasserstein Generative Adversarial Networks*, in *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Sydney, Australia. p. 214--223. 2017
- [110] Karras, T., Aila, T., Laine, S., and Lehtinen, J., *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. October 01, 2020, <https://arxiv.org/abs/1710.10196>. 2017
- [111] Huang, X., Li, Y., Poursaeed, O., Hopcroft, J., and Belongie, S., *Stacked generative adversarial networks*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, Hawaii, USA. p. 1866-1875. 2017
- [112] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W., *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, Hawaii, USA. p. 105-114. 2017
- [113] Yeh, R.A., Chen, C., Lim, T. Y., Schwing, A. G., Hasegawa-Johnson, M., and Do, M. N., *Semantic Image Inpainting with Deep Generative Models*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, Hawaii, USA. p. 6882-6890. 2017
- [114] Odena, A., Olah, C., and Shlens, J., *Conditional Image Synthesis with Auxiliary Classifier GANs*, in *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Sydney, Australia. p. 2642--2651. 2017

- [115] Zhu, J.Y., Park, T., Isola, P., and Efros, A. A., *Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks*, in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy. p. 2242-2251. 2017
- [116] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A., *Generative Adversarial Networks: An Overview*. IEEE Signal Processing Magazine, **(35)**, (1) 53-65. 2018
- [117] Hong, Y., Hwang, U., Yoo, J., and Yoon, S., *How Generative Adversarial Networks and Their Variants Work: An Overview*. ACM Computing Surveys, **(52)**, (1) Article 10. 2019
- [118] Mirza, M., and Osindero, S., *Conditional Generative Adversarial Nets*. November 01, 2020, <https://arxiv.org/abs/1411.1784>. 2014
- [119] Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., and Abbeel, P., *InfoGAN: interpretable representation learning by information maximizing generative adversarial nets*, in *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Barcelona, Spain. p. 2180–2188. 2016
- [120] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A., *Improved training of wasserstein GANs*, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, California, USA. p. 5769–5779. 2017
- [121] Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Smolley, S. P., *Least Squares Generative Adversarial Networks*, in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy. p. 2813-2821. 2017
- [122] Berthelot, D., Schumm, T., and Metz, L., *BEGAN: Boundary Equilibrium Generative Adversarial Networks*. March 01, 2021, <https://arxiv.org/abs/1703.10717>. 2017



- [123] Qi, G.J., *Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities*. International Journal of Computer Vision, **(128)**, (5) 1118-1140. 2020
- [124] Tejeda-Ocampo, C., López-Cuevas, A., and Terashima-Marin, H., *Improving Deep Interactive Evolution with a Style-Based Generator for Artistic Expression and Creative Exploration*. Entropy, **(23)**, (1) 11. 2021
- [125] Yang, J., *Review of Image-Based Person Re-Identification in Deep Learning*. Journal of New Media, **(2)**, (4) 137-148. 2020
- [126] Wu, L., Shen, C., and Hengel, A., *Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification*. Pattern Recognition, **(65)**, 238-250. 2017
- [127] Zheng, L., Zhang, H., Sun, S., Chandraker, M., Yang, Y., and Tian, Q., *Person Re-identification in the Wild*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA. p. 3346-3355. 2017
- [128] Hussin, S., and Yildirim, R., *StyleGAN-LSRO Method for Person Re-Identification*. IEEE Access, **(9)**, 13857-13869. 2021
- [129] Xiong, F., Xiao, Y., Cao, Z., Gong, K., Fang, Z., and Zhou, J., *Good practices on building effective CNN baseline model for person re-identification*. Proceedings of the Tenth International Conference on Graphic and Image Processing (ICGIP). Vol. 11069. Chengdu, China. 2019
- [130] Sun, Y., Zheng, L., Deng, W., and Wang, S., *SVDNet for Pedestrian Retrieval*, in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy. p. 3820-3828. 2017
- [131] Zheng, Z., Zheng, L., and Yang, Y., *Pedestrian Alignment Network for Large-scale Person Re-Identification*. IEEE Transactions on Circuits and Systems for Video Technology, **(29)**, (10) 3037-3045. 2019

- [132] Xiao, T., Li, S., Wang, B., Lin, L., and Wang, X., *End-to-End Deep Learning for Person Search*. January 20, 2021, <https://arxiv.org/pdf/1604.01850>. 2016
- [133] Xiao, T., Li, S., Wang, B., Lin, L., and Wang, X., *Joint Detection and Identification Feature Learning for Person Search*, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA. p. 3376-3385. 2017
- [134] Zheng, Z., Zheng, L., and Yang, Y., *A Discriminatively Learned CNN Embedding for Person Reidentification*. ACM Transactions on Multimedia Computing, Communications, and Applications, **(14)**, (1) Article 13. 2018
- [135] Ding, S., Lin, L., Wang, G., and Chao, H., *Deep feature learning with relative distance comparison for person re-identification*. Pattern Recognition, **(48)**, (10) 2993-3003. 2015
- [136] Hermans, A., Beyer, L., and Leibe, B., *In Defense of the Triplet Loss for Person Re-Identification*. March 01, 2021, <https://arxiv.org/abs/1703.07737>. 2017
- [137] Zhao, C., Chen, K., Wei, Z., Chen, Y., Miao, D., and Wang, W., *Multilevel triplet deep learning model for person re-identification*. Pattern Recognition Letters, **(117)**, 161-168. 2019
- [138] Liu, H., Feng, J., Qi, M., Jiang, J., and Yan, S., *End-to-End Comparative Attention Networks for Person Re-Identification*. IEEE Transactions on Image Processing, **(26)**, 3492-3506. 2017
- [139] Bai, X., Yang, M., Huang, T., Dou, Z., Yu, R., and Xu, Y., *Deep-Person: Learning Discriminative Deep Features for Person Re-Identification*. November 01, 2020, <https://arxiv.org/abs/1711.10658>. 2017
- [140] Wang, G., Yuan, Y., Chen, X., Li, J., and Zhou, X., *Learning discriminative features with multiple granularities for person re-identification*, in *Proceedings of the 2018 ACM Multimedia Conference*. Seoul, Republic of Korea. p. 274-282. 2018

- [141] Wang, F., Zuo, W., Lin, L., Zhang, D., and Zhang, L., *Joint learning of single-image and cross-image representations for person re-identification*, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV, USA. p. 1288-1296. 2016
- [142] Yao, H., Zhang, S., Hong, R., Zhang, Y., Xu, C., and Tian, Q., *Deep Representation Learning With Part Loss for Person Re-Identification*. IEEE Transactions on Image Processing, **(28)**, (6) 2860-2871. 2019
- [143] Yang, W., Huang, H., Zhang, Z., Chen, X., Huang, K., and Zhang, S., *Towards Rich Feature Discovery With Class Activation Maps Augmentation for Person Re-Identification*, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA. p. 1389-1398. 2019
- [144] Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., and Hoi, S. C. H., *Deep Learning for Person Re-identification: A Survey and Outlook*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1-1. 2021
- [145] Li, W., Zhu, X., and Gong, S., *Person re-identification by deep joint learning of multi-loss classification*, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. Melbourne, Australia. p. 2194–2200. 2017
- [146] Wang, C., Zhang, Q., Huang, C., Liu, W., and Wang, X., *Manacs: A Multi-task Attentional Network with Curriculum Sampling for Person Re-Identification*, in *Proceedings of the European Conference on Computer Vision (ECCV)*. Munich, Germany. p. 384-400. 2018
- [147] Li, W., Zhu, X., and Gong, S., *Harmonious Attention Network for Person Re-identification*, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA. p. 2285-2294. 2018
- [148] Ainam, J., Qin, K., Liu, G., and Luo, G., *Sparse Label Smoothing Regularization for Person Re-Identification*. IEEE Access, **(7)**, 27899-27910. 2019

- [149] Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D., *Object Detection with Discriminatively Trained Part-Based Models*. IEEE Transactions on Pattern Analysis and Machine Intelligence, **(32)**, (9) 1627-1645. 2010
- [150] Schwartz, W.R., and Davis, L. S., *Learning Discriminative Appearance-Based Models Using Partial Least Squares*, in *2009 XXII Brazilian Symposium on Computer Graphics and Image Processing*. Rio de Janeiro, Brazil. p. 322-329. 2009
- [151] Hirzer, M., Beleznaï, C., Roth, P. M., and Bischof, H., *Person Re-identification by Descriptive and Discriminative Classification*, in *Proceedings of the Scandinavian Conference on Image Analysis (SCIA)*. Ystad, Sweden. p. 91-102. 2011
- [152] Wang, T., Gong, S., Zhu, X., and Wang, S., *Person Re-identification by Video Ranking*, in *European Conference on Computer Vision (ECCV)*. Springer International Publishing. Cham. p. 688-703. 2014
- [153] karanam, S., Gou, M., Wu, Z., Rates-Borras, A., Camps, O., and Radke, R. J., *A Systematic Evaluation and Benchmark for Person Re-Identification: Features, Metrics, and Datasets*. IEEE Transactions on Pattern Analysis and Machine Intelligence, **(41)**, (3) 523-536. 2019
- [154] Wang, X., Doretto, G., Sebastian, T., Rittscher, J., and Tu, P., *Shape and Appearance Context Modeling*, in *2007 IEEE 11th International Conference on Computer Vision*. Rio de Janeiro, Brazil. p. 1-8. 2007
- [155] Liu, X., Liu, W., Mei, T., and Ma, H., *A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance*, in *Proceedings of European Conference on Computer Vision (ECCV)*. Springer International Publishing. Cham. p. 869-884. 2016
- [156] Zhong, Z., Zheng, L., Cao, D., and Li, S., *Re-ranking Person Re-identification with k-Reciprocal Encoding*, in *Proceedings of the IEEE Conference on*

- Computer Vision and Pattern Recognition (CVPR)*. Honolulu, Hawaii, USA. p. 3652-3661. 2017
- [157] Qian, N., *On the momentum term in gradient descent learning algorithms*. Neural Networks, **(12)**, (1) 145-151. 1999
  - [158] Chen, Y., Zhu, X., and Gong, S., *Person Re-identification by Deep Learning Multi-scale Representations*, in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. Venice, Italy. p. 2590-2600. 2017
  - [159] Hussin, S., Elashkek, K., and Yildirim, R., *Convolutional neural network baseline model building for person re-identification*, in *Proceedings of the International Conference on Engineering Technologies (ICENTE'19)*. Konya, Turkey. p. 53-57. 2019
  - [160] Qin, D., Gammeter, S., Bossard, L., Quack, T., and van Gool, L., *Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors*, in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Colorado Springs, CO, USA. p. 777-784. 2011
  - [161] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T., *Analyzing and Improving the Image Quality of StyleGAN*, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA. p. 8107-8116. 2020
  - [162] Huang, X., and Belongie, S., *Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization*, in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy. p. 1510-1519. 2017
  - [163] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S., *GANs trained by a two time-scale update rule converge to a local nash equilibrium*, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, California, USA. p. 6629–6640. 2017

- [164] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. , *Image quality assessment: from error visibility to structural similarity*. IEEE Transactions on Image Processing, (13), (4) 600-612. 2004
- [165] *StyleGAN — Official TensorFlow Implementation*. Retrieved from <https://github.com/NVLabs/stylegan>. Accessed on: 25/3/2020.
- [166] Zhang, X., Luo, H., Fan, X., Xiang, W., Sun, Y., Xiao, Q., Jiang, W., Zhang, C., and Sun, J., *AlignedReID: Surpassing Human-Level Performance in Person Re-Identification*. November 01, 2020, <https://arxiv.org/abs/1711.08184>. 2017
- [167] Si, J., Zhang, H., Li, C., Kuen, J., Kong, X., Kot, A. C., and Wang, J., *Dual Attention Matching Network for Context-Aware Feature Sequence Based Person Re-identification*, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA. p. 5363-5372. 2018
- [168] Fan, H., Zheng, L., Yan, C., and Yang, Y., *Unsupervised Person Re-identification: Clustering and Fine-tuning*. ACM Transactions on Multimedia Computing, Communications, and Applications, (14), (4) Article 83. 2018
- [169] Hou, R., Ma, B., Chang, H., Gu, X., Shan, S., and Chen, X., *Interaction-And-Aggregation Network for Person Re-Identification*, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA. p. 9309-9318. 2019
- [170] Zhou, K., Yang, Y., Cavallaro, A., and Xiang, T., *Omni-Scale Feature Learning for Person Re-Identification*, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, South Korea p. 3701-3711. 2019
- [171] Fang, P., Zhou, J., Roy, S., Petersson, L., and Harandi, M., *Bilinear Attention Networks for Person Retrieval*, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, South Korea p. 8029-8038. 2019

- [172] Chen, G., Lin, C., Ren, L., Lu, J., and Zhou, J., *Self-Critical Attention Learning for Person Re-Identification*, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South). p. 9636-9645. 2019
- [173] Huang, Y., Xu, J., Wu, Q., Zheng, Z., Zhang, Z., and Zhang, J., *Multi-Pseudo Regularized Label for Generated Data in Person Re-Identification*. *IEEE Transactions on Image Processing*, **(28)**, (3) 1391-1403. 2019

# APPENDICES

**Appendix A:** The Mathematical Foundation of GANs

**Appendix B:** Label Smoothing Regularization (LSR)

**Appendix C:** PyTorch Framework

**Appendix D:** TensorFlow Platform



## Appendix A – The Mathematical Foundation of GANs

In this appendix, we provide an overview of GANs from a mathematical point of view. It is possible to conceive of GAN as a sort of interplay occurring between two distinct models: a discriminator and a generator. As a result, both of the specified models possess their own distinct loss functions.

**The Discriminator:** The discriminator serves the purpose of accurately labeling images that have been generated as being false while labeling the empirical data points as being true. Accordingly, the following loss function of the discriminator may be applied:

$$L_D = \text{Error}(D(x), 1) + \text{Error}(D(G(z)), 0) \quad (\text{A.1})$$

**The Generator:** The generator's goal is to confuse the discriminator to the largest possible extent so that generated images will be labeled falsely as true.

$$L_G = \text{Error}(D(G(z)), 1) \quad (\text{A.2})$$

The most important point to be considered here is that we need to ensure the minimization of the loss function. The generator needs to pursue the minimization of the difference of values between 1, the label assigned to true data, and the label that the discriminator assigns to the generated fake data. Binary cross-entropy is a particularly popular loss function applied in cases such as these. The formula for binary cross-entropy is defined as shown below:

$$H(p, q) = \mathbb{E}_{x \sim p(x)} [-\log q(x)] \quad (\text{A.3})$$

For tasks of classification, random variables will be discrete. The expected result can accordingly be written as follows.

$$H(p, q) = - \sum_{x \in X} p(x) \log q(x) \quad (\text{A.4})$$

In situations of binary cross-entropy, two labels are relevant: zero and one. This expression can be simplified even further, as shown below:

$$H(y, \hat{y}) = -\sum y \log(\hat{y}) + (1-y) \log(1-\hat{y}) \quad (\text{A.5})$$

Applying this to the discriminator loss function,

$$L_D = - \sum_{x \in X, z \in Z} \log(D(x)) + \log(1 - D(G(z))) \quad (\text{A.6})$$

Doing the same for the generator loss function,

$$L_G = - \sum_{z \in Z} \log(D(G(z))) \quad (\text{A.7})$$

With these specified loss functions, the training of both the discriminator and the generator becomes possible.

The original paper on GAN by Goodfellow presented a somewhat varied picture of the loss functions compared to that presented above. That paper proposed that the GAN can be formulated mathematically by the minimax of a target function between the discriminator function and the generator function. The target loss function is proposed to be as shown below:

$$V(D, G) := E_{x \sim X} [\log D(x)] + E_{z \sim Z} [\log(1 - D(G(z)))] \quad (\text{A.8})$$

Here, E denotes the expectation with respect to a distribution specified in the subscript. The discriminator, in this case, must pursue the maximization of the established quantity, but the goal of the generator is the opposite. This can be illustrated as follows,

$$\min_G \max_D V(D, G) := \min_G \max_D \left( E_{x \sim X} [\log D(x)] + E_{z \sim Z} [\log(1 - D(G(z)))] \right) \quad (\text{A.9})$$

The min-max formulation can be portrayed concisely in one line. It demonstrates, in an intuitive manner, the adversarial quality of conflict that occurs between generator and discriminator. In reality, though, we may write distinct loss functions for generator and discriminator, precisely as has been done here. The gradient of function  $y=\log(x)$  is more steep as we approach  $x=0$  than the gradient of function  $y=\log(1-x)$ . As a result, any efforts that we make for maximizing  $\log(D(G(z)))$  or, conversely, minimizing  $-\log(D(G(z)))$  will provide us with faster and more valuable improvements in the performances of the generator rather than merely making efforts to minimize  $\log(1-D(G(z)))$ .

**Model Optimization:** After defining the generator's and the discriminator's loss functions, we need to undertake some mathematical operations for solving the optimization problem. In other words, we need to find the parameters of the generator and the discriminator that will allow for the optimization of the relevant loss functions. Practically speaking, this means that we are training the model.

**Training the Discriminator:** In the process of undertaking the training of a GAN, a single model is usually trained at one single point in time. That is to say, while we are training a discriminator, we assume that the generator must be fixed. Based on the min-max game, we are able to define the quantity of interest as a function of  $G$  and  $D$ , as shown below:

$$V(G, D) = E_{x \sim p_{data}} [\log(D(x))] + E_{z \sim p_z} [\log(1 - D(G(z)))] \quad (A.10)$$

In the present work, our focus is more heavily on the distribution modeled by the generator than on  $p_z$ . Accordingly, we can suggest a novel variable,  $y=G(z)$ , and we can utilize this novel substitution with the aim of rewriting the value function as follows:

$$\begin{aligned} V(G, D) &= E_{x \sim p_{data}} [\log(D(x))] + E_{y \sim p_g} [\log(1 - D(y))] \\ &= \int_{x \in X} p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) dx \end{aligned} \quad (A.11)$$

The discriminator serves the purpose of maximizing this value function. By utilizing the partial derivative of  $V(G,D)$  in light of  $D(x)$ , it is clear that we have the optimal discriminator, denoted as  $D^*(x)$ , when the following holds,

$$\frac{p_{data}(x)}{D(x)} - \frac{p_g(x)}{1-D(x)} = 0 \quad (A.12)$$

If we rearrange Equation (A.12) above,

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \quad (A.13)$$

Here, we have the necessary condition for the optimal discriminator.

**Training the Generator:** For the training of the generator, we need to make the assumption that the discriminator is fixed. Accordingly, we can proceed to analyze the value function.

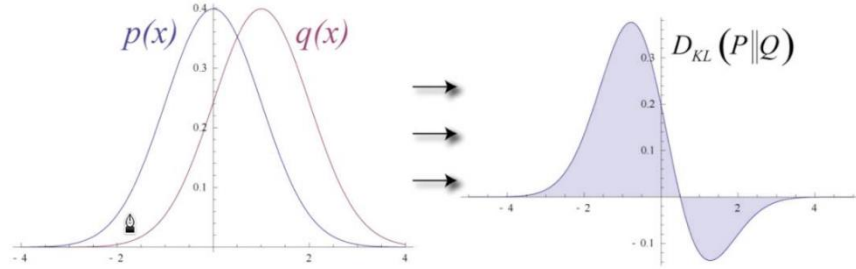
$$\begin{aligned} V(G, D^*) &= E_{x \sim p_{data}} [\log(D^*(x))] + E_{x \sim p_g} [\log(1 - D^*(x))] \\ &= E_{x \sim p_{data}} \left[ \log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right] + E_{x \sim p_g} \left[ \log \frac{p_g(x)}{p_{data}(x) + p_g(x)} \right] \end{aligned} \quad (A.14)$$

By exploiting the properties of logarithms,

$$\begin{aligned} V(G, D^*) &= E_{x \sim p_{data}} \left[ \log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right] + E_{x \sim p_g} \left[ \log \frac{p_g(x)}{p_{data}(x) + p_g(x)} \right] \\ &= -\log 4 + E_{x \sim p_{data}} \left[ \log p_{data}(x) - \log \frac{p_{data}(x) + p_g(x)}{2} \right] \\ &\quad + E_{x \sim p_g} \left[ \log p_g(x) - \log \frac{p_{data}(x) + p_g(x)}{2} \right] \end{aligned} \quad (A.15)$$

The expectations can be interpreted as a function of the Kullback-Leibler (KL) divergence,

$$V(G, D^*) = -\log 4 + D_{KL}\left(p_{data} \parallel \frac{p_{data} + p_g}{2}\right) + D_{KL}\left(p_g \parallel \frac{p_g + p_{data}}{2}\right) \quad (A.16)$$



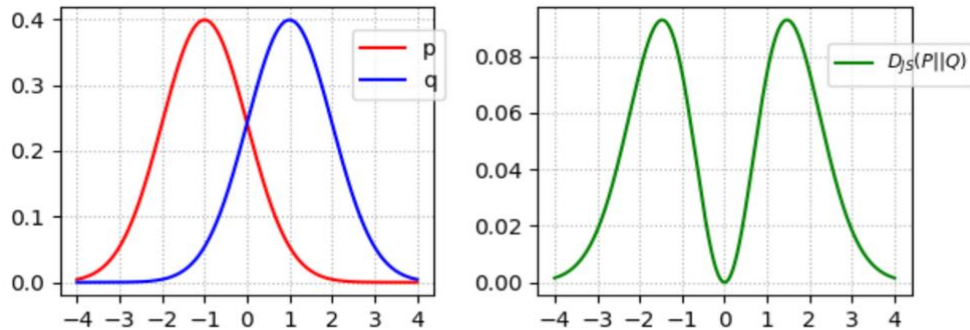
**Figure A.1** KL Divergence

This gives rise to the Jensen-Shannon (JS) divergence, and we can define JS divergence in the following way:

$$J(P, Q) = \frac{1}{2}(D(P \parallel R) + D(Q \parallel R)) \quad (A.17)$$

Here,  $R = \frac{1}{2}(P + Q)$ . As a result, the function can be given in the form of JS divergence:

$$V(G, D^*) = -\log 4 + 2 \cdot D_{JS}(p_{data} \parallel p_g) \quad (A.18)$$



**Figure A.2** JS Divergence

Equation (A.18) illustrates for us that, whenever  $D$  is free of a capacity limitation and is furthermore optimal, we know that the GAN loss function will measure any similarity between  $p_{data}$  and  $p_g$  using JS divergence. It must be noted, though, that the results explained here to give us a convenient theoretical outcome, but, in reality,  $D$  is almost never completely optimal while we are optimizing  $G$ . Thus, other potential GAN architectures are suggested in the literature with variants of loss functions to address this problem and pursue the possibility of full optimality. Below, Table A.1 lists several GAN loss variants that have been proposed over the years.

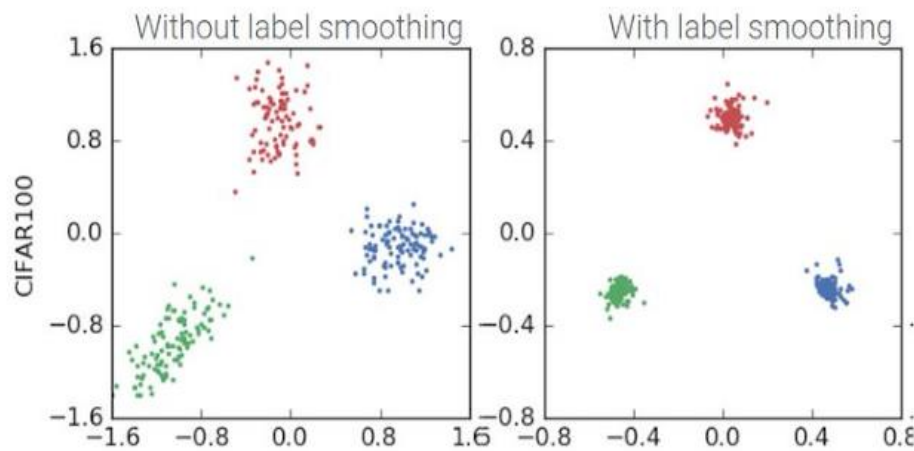
**Table A.1** GAN loss variants

Name	Value Function
GAN	$L_D^{GAN} = E[\log(D(x))] + E[\log(1 - D(G(z)))]$ $L_G^{GAN} = E[\log(D(G(z)))]$
LSGAN	$L_D^{LSGAN} = E[(D(x) - 1)^2] + E[D(G(z))^2]$ $L_G^{LSGAN} = E[(D(G(z)) - 1)^2]$
WGAN	$L_D^{WGAN} = E[D(x)] - E[D(G(z))]$ $L_G^{WGAN} = E[D(G(z))]$ $W_D \leftarrow clip\_by\_value(W_D, -0.01, 0.01)$
WGAN_GP	$L_D^{WGAN\_GP} = L_D^{WGAN} + \lambda E[( \nabla D(\alpha x - (1 - \alpha G(z)))  - 1)^2]$ $L_G^{WGAN\_GP} = L_G^{WGAN}$
DRAGAN	$L_D^{DRAGAN} = L_D^{GAN} + \lambda E[( \nabla D(\alpha x - (1 - \alpha x_p))  - 1)^2]$ $L_G^{DRAGAN} = L_G^{GAN}$
CGAN	$L_D^{CGAN} = E[\log(D(x, c))] + E[\log(1 - D(G(z), c))]$ $L_G^{CGAN} = E[\log(D(G(z), c))]$
infoGAN	$L_{D,Q}^{infoGAN} = L_D^{GAN} - \lambda L_I(c, c')$ $L_G^{infoGAN} = L_G^{GAN} - \lambda L_I(c, c')$
ACGAN	$L_{D,Q}^{ACGAN} = L_D^{GAN} + E[P(class = c x)] + E[P(class = c G(z))]$ $L_G^{ACGAN} = L_G^{GAN} + E[P(class = c G(z))]$
EBGAN	$L_D^{EBGAN} = D_{AE}(x) + \max(0, m - D_{AE}(G(z)))$ $L_G^{EBGAN} = D_{AE}(G(z)) + \lambda \cdot PT$
BEGAN	$L_D^{BEGAN} = D_{AE}(x) - k_t D_{AE}(G(z))$ $L_G^{BEGAN} = D_{AE}(G(z))$ $k_{t+1} = k_t + \lambda(\gamma D_{AE}(x) - D_{AE}(G(z)))$

## Appendix B – Label Smoothing Regularization (LSR)

In this appendix, we explain label smoothing regularization (LSR), which is used for learning in cases that are fully supervised. LSR has been extended for scenarios of unlabeled learning, which has yielded the valuable LSRO method.

**Label Smoothing:** This can be understood as a helpful technique for regularization introducing noises for labels. Such an understanding takes into consideration the reality that datasets may contain mistakes. Thus, directly maximizing the likelihood of  $\log p(y|x)$  is quite risky. In contrast, we must assume that, for a small constant  $\varepsilon$ , training set label  $y$  is not false with the probability  $1 - \varepsilon$ ; it is otherwise false. Label smoothing will regularize any model that is based on SoftMax with  $k$  output values with replacement of the hard 0 and 1 classification targets by respective targets of  $\frac{\varepsilon}{k-1}$  and  $1 - \varepsilon$ .



**Figure B.1** Effect of label smoothing on the accuracy of the classification deep models.

In brief, LSR assigns smaller values to non-ground truth classes rather than simply 0. Such an approach will help to ensure that the network is not simply tuned toward the ground truth class and it will accordingly reduce any chance of overfitting. Accordingly, LSR suggested to be applied with the cross-entropy loss. We can formally suggest that  $k \in \{1, 2, \dots, K\}$  represents the pre-defined classes of training

data, wherein  $K$  signifies numbers of classes. Cross-entropy loss may accordingly be represented as follows:

$$L = -\sum_{k=1}^K \log(p(k))q(k) \quad (\text{B.1})$$

Here, we understand that  $p(k) \in [0,1]$  is the predicted probability of input belonging to class  $k$ , able to be output by the CNN. We can derive this from the SoftMax function, which we utilize for normalization of the output of the previous FC layer. In this case,  $q(k)$  is the ground truth distribution. We take  $y$  to represent the ground truth class label. Accordingly, it is possible to see that  $q(k)$  can be defined as follows:

$$q(k) = \begin{cases} 0, & k \neq y \\ 1, & k = y \end{cases} \quad (\text{B.2})$$

Discarding the 0 terms in Equation (B.1), we quickly see that cross-entropy loss is equal to consideration of merely the ground truth term in Equation (B.3).

$$L = -\log(p(y)) \quad (\text{B.3})$$

As a result of all this, the minimization of cross-entropy loss is clearly equal to maximization of the predicted ground-truth class probability. LSR can be introduced here for better considering the distribution of the non-ground truth classes. As a result, we are encouraging the network to not be overly confident regarding the ground truth. Label distribution  $q_{LSR}(k)$  may be written in the following manner:

$$q_{LSR}(k) = \begin{cases} \frac{\varepsilon}{K}, & k \neq y \\ 1 - \varepsilon + \frac{\varepsilon}{K}, & k = y \end{cases} \quad (\text{B.4})$$



Here,  $\varepsilon \in [0,1]$  is a hyperparameter. In the event that  $\varepsilon$  is zero, Equation (B.4) will be reduced to Equation (B.2). However, in the event that  $\varepsilon$  is overly large, we may see the model failing in its ability to predict the ground truth label. As a result, in the majority of cases  $\varepsilon$  is taken as 0.1. If we assume the non-ground truth classes to be taking on uniform label distributions, we can consider Equation (B.1) and Equation (B.4), and cross-entropy loss will become the following:

$$L_{LSR} = -(1-\varepsilon)\log(p(y)) - \frac{\varepsilon}{K} \sum_{k=1}^K \log(p(k)) \quad (\text{B.5})$$

Compared with Equation (B.3), Equation (B.5) is more relevant to the other classes, not simply merely the ground truth class. The present dissertation has not employed LSR on the IDE baseline because doing so would generate a relatively worse performance compared to that of Equation (B.2). LSR is introduced again in this work simply because it constitutes the foundation in the design of the LSRO method.

## Appendix C – PyTorch Framework

PyTorch is an AI framework developed by Facebook. This scientific computational package is based on Python and it makes good use of the power of GPUs. Furthermore, it is one of the most widely preferred DL research platforms, having been designed for the provision of maximum levels of both flexibility and speed. Its popularity largely derives from the fact that it can provide two of the most desired high-level features: tensor computations alongside serious support for GPU acceleration while constructing deep NNs via tape-based auto-grad systems.

The key highlights that make PyTorch perfect for research experiments include:

- **A simple interface:** Its API is decidedly easy to utilize; as a result, it is very easy to operate and to run, very similarly to Python.
- **Pythonic in nature:** Because the library is “Pythonic,” it can integrate very smoothly with Python’s data science stack. Accordingly, it is able to leverage all possible functionalities and services that the Python environment offers.
- **Computational graphs:** PyTorch offers an unparalleled platform with dynamic computational graphs that may be changed during runtime. Such an option is particularly valuable when you do not know in advance the extent of the memory that will be necessary to create an NN model.
- PyTorch is able to solve significant problems that arise during the course of research work. It is a deeply favored library for DL and artificial intelligence. Please consult <https://www.pytorch.org/> for more details.

## Appendix D – TensorFlow Platform

Open-source TensorFlow is an end-to-end platform facilitating the processes of machine learning. It is a symbolic math library designed around the basics of dataflow and differentiable programming. This tool is popular at Google for both research and production purposes. It offers flexible and uniquely extensive tools, community resources, and libraries to allow researchers to advance the realm of state-of-the-art machine learning while also allowing developers to more easily create and apply machine learning-powered applications. It is particularly focused on the training and inference of deep NNs. The Google Brain team initially deployed TensorFlow for internal Google use. It was officially released under the Apache License 2.0 in 2015. It performs well on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units).

This platform is able to accept data as multi-dimensional arrays of higher dimensions, referred to as tensors. Such arrays of multiple dimensions are valuable in settings that require the handling of huge volumes of data. TensorFlow operates according to data flow graphs that possess both nodes and edges. With the execution mechanism being designed in graph form, executing TensorFlow code in a distributed manner is considerably easier across clusters of computers when using GPUs. TensorFlow has the capability of supporting fine-grained layers of networks, allowing users to build complex types of novel layers without needing to implement them in any low-level languages. Subgraph execution permits the introduction and retrieval of the results of discretionary data on any edges of graphs. This feature is of high value in the process of debugging more complicated computational graphs.

TensorFlow offers the use of APIs in many languages for constructing TensorFlow graphs as well as executing them. The Python API is currently the most fully complete and most easiest API to be used in this context. Other possible language APIs such as C++, Java, GO, R, and Haskell are also supported. They can be integrated into projects to offer some performance advantages in graph execution. Furthermore, TensorFlow enjoys support in both Google and Amazon Cloud environments. Please refer to <https://www.tensorflow.org/> for more information.



## **PUBLICATIONS**

- 1) **Saleh Hussin Salem HUSSIN**, and Remzi YILDIRIM. “StyleGAN-LSRO Method for Person Re-identification”. In: IEEE Access, vol. 9, pp. 13857-13869, **2021**, doi: 10.1109/ACCESS.2021.3051723.
- 2) **Saleh Hussin Salem HUSSIN**, and Remzi YILDIRIM, “Developing an Effective Baseline Model for Person Re-Identification Based on Convolutional Neural Networks and Transfer Learning”. The journal Tehnički vjesnik/Technical Gazette, vol. 29, no. 1. (Accepted and to be published towards the end of February 2022).
- 3) Remzi YILDIRIM, **S. H. S. HUSSIN**, K. M. ALLASHIK, A. ALGUTAR, and A. HAZER. “Optimizing Higher Education with Economic Layers”. In: American Journal of Engineering Research (AJER), vol. 10, iss. 5. pp. 202-208, **2021**.
- 4) Remzi YILDIRIM, **S. H. S. HUSSIN**, K. M. ALLASHIK, A. ALGUTAR, and A. HAZER. “Some Strategic Critical Sizes of China”. In: American Journal of Engineering Research (AJER), vol. 10, iss. 5. pp. 209-223, **2021**.
- 5) A. EFE, and **S. HUSSIN**, “Malware Visualization Techniques”, International Journal of Applied Mathematics Electronics and Computers, vol. 8, no. 1, pp. 7-20, 2020, doi:10.18100/ijamec.526813.
- 6) **S. HUSSIN**, K. ALASHIK, and R. YILDIRIM, “Convolutional neural network baseline model building for person re-identification”, In: Proceedings of the International Conference on Engineering Technologies (ICENTE'19), pp 53-57, Konya, TURKEY, October 25-27, 2019. (**Awarded best presentation**).
- 7) K. ALASHIK, **S. HUSSIN**, and R. YILDIRIM, “Observations on the Evaluation of Dorsal Hand Vein (DHV) Recognition and Identification”, In: Proceedings of the International Conference on Engineering Technologies (ICENTE'19), pp 62-68, Konya, TURKEY, October 25-27, 2019.
- 8) **S. HUSSIN**, K. ALASHIK and R. YILDIRIM, “An improved deep learning-based person re-identification system”, In: Proceedings of the International Conference on

Access to Recent Advances in Engineering and Digitalization (ARACONF'20), pp 61, Kayseri, TURKEY, March 05-06, 2020.

- 9) K. ALASHIK, **S. HUSSIN**, and R. YILDIRIM, “Dorsal Hand Vein Identification Based on Deep Convolutional Neural Networks and Visualizing Intermediate Layer Activations”, In: Proceedings of the International Conference on Access to Recent Advances in Engineering and Digitalization (ARACONF'20), pp 62, Kayseri, TURKEY, March 05-06, 2020.
- 10) A. ALGUTTAR, **S. HUSSIN**, K. ALASHIK, and R. YILDIRIM, “An Observation of Intrusion Detection Techniques in Cyber Physical Systems”, Avrupa Bilim ve Teknoloji Dergisi, pp 277-284. 2020.
- 11) M. A. M. ESHTAWIE, **S. H. S. HUSSIN**, and M. OTHMAN, “Analysis of results obtained with a new proposed low area low power high-speed fixed-point adder”, In: 2010 IEEE International Conference on Semiconductor Electronics (ICSE2010), Melaka, MALAYSIA, 2010, pp. 127-130, doi: 10.1109/SMELEC.2010.5549387.
- 12) **S. H. HUSSIN**, E. A. BABIKER, A. R. RAMLI and S. A. R. AL-HADDAD, “A Novell Image Watermarking System Based on Discrete Wavelet Transform and Fibonacci Numbers for Copyright Protection”, In: Proceedings of Brunei International Conference on Engineering and Technology (BICET 2005), BRUNEI, August 15-18, 2005.



---

**Saleh Hussin Salem HUSSIN**

---

**Department of Computer Engineering**

---

**June 2021 ANKARA**

---