İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KANAL TÜMLEŞTİRME MEKANİZMALI DİNAMİK KONVOLÜSYON İLE DERİN ÖĞRENME

YÜKSEK LİSANS TEZİ

Elif Ecem AKBABA

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

OCAK 2023

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KANAL TÜMLEŞTİRME MEKANİZMALI DİNAMİK KONVOLÜSYON İLE DERİN ÖĞRENME

YÜKSEK LİSANS TEZİ

Elif Ecem AKBABA
(504191315)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Bilge GÜNSEL

OCAK 2023

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DEEP LEARNING VIA DYNAMIC CONVOLUTION WITH CHANNEL FUSION MECHANISM

M.Sc. THESIS

Elif Ecem AKBABA
(504191315)

Department of Electronical and Communication Engineering

Telecommunication Engineering Programme

Thesis Advisor: Prof. Dr. Bilge GÜNSEL

JANUARY 2023

İTÜ, Lisansüstü Eğitim Enstitüsü'nün 504191315 numaralı Yüksek Lisans Öğrencisi Elif Ecem AKBABA, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "KANAL TÜMLEŞTİRME MEKANİZMALI DİNAMİK KONVOLÜSYON İLE DERİN ÖĞRENME" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Prof. Dr. Bilge GÜNSEL** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Doç. Dr. Yusuf YASLAN** İstanbul Teknik Üniversitesi	.....
	Dr. Öğr. Üyesi İnci M. BAYTAŞ Boğaziçi Üniversitesi	.....

Teslim Tarihi : **30 Aralık 2022**
Savunma Tarihi : **18 Ocak 2023**

Dedeme,

ÖNSÖZ

Yüksek lisans eğitimim boyunca bilgi ve deneyimlerini benimle paylaşan ve akademik hayatımda bana destek olan danışmanım Prof. Dr. Bilge GÜNSEL'e teşekkür ederim. Çalışmalarımda bilgisi ve manevi desteğiyle yanımda olan Dr. Filiz GÜRKAN GÖLCÜK'e ve yüksek lisans hayatım boyunca bulunduğum İTÜ Çoğul Ortam İşaret İşleme ve Örüntü Tanıma Laboratuvarı'ndaki çalışma arkadaşlarıma teşekkürlerimi sunarım.

Hayatım boyunca sürekli ve koşulsuz şekilde beni destekleyen ve her zaman yanımda olan anneme, babama ve kardeşime, en büyük motivasyon kaynağım dedeme, hayatımı güzelleştiren ve daha iyi bir insan olmamı sağlayan Alper GİRGİN'e ve bütün arkadaşlarıma sonsuz teşekkür ederim.

Aralık 2022

Elif Ecem AKBABA
(Elektronik ve Haberleşme Mühendisi)

İÇİNDEKİLER

KISALTMALAR

Re-ID : Kişi Tanılama (Person Re-identification)
CNN : Konvolüsyonel Sinir Ağı (Convolutional Neural Network)
mAP : Ortalama Hassasiyet (Mean Average Precision)
FC : Tam Bağlantılı (Fully Connected)
ReLU : Doğrultulmuş Lineer Birim (Rectified Linear Unit)
ep : Döngü (Epoch)
BN : "Batch" Normalizasyonu (Batch Normalization)
SVD : Tekil Değer Ayrıştırma (Singular Value Decomposition)
DCD : Dinamik Konvolüsyon Ayrıştırma (Dynamic Convolution Decomposition)
RPN : Bölge Öneri Ağı (Region Proposal Network)
BON : "Bottleneck"
GT : Gerçek Referans Değer (Ground Truth)
GCN : Graf Konvolüsyon Ağı (Graph Convolutional Network)
AP : Ortalama Kesinlik (Average Precision)
SGD : Rasgele Gradyan İnişi (Stochastic Gradient Descent)
LR : Öğrenme Oranı (Learning Rate)
CA : Kosinüs Yumuşatma (Cosine Annealing)
SR : Başarım Oranı (Success Rate)

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

KANAL TÜMLEŞTİRME MEKANİZMALI DİNAMİK KONVOLÜSYON İLE DERİN ÖĞRENME

ÖZET

Derin öğrenme ağları son yıllarda otonom sürüş, doğal dil işleme, medikal görüntüleme gibi birçok alanda kullanılmaktadır. Bu ağlarda yüksek performans elde edebilmek açısından en önemli işlem, öğrenmede kullanılan büyük boyutlu veriyi iyi temsil eden özniteliklerin çıkarılabilmesidir. Literatürde statik konvolüsyonlu ağlara göre daha ayırt edici öznitelikler çıkarılması amacıyla, dinamik konvolüsyonlu mimariler önerilmiştir. Dinamik konvolüsyonlu ağlarda eğitim aşamasında aynı katmanda kullanılan birden fazla kernelin ve bu kernel çıktılarındaki öznitelikleri tümleştiren ağırlıkların öğrenilmesi gerçeklenir. Çıkarım aşamasında, öğrenilen tümleştirme ağırlıkları giriş verisine bağlı olarak güncellenir. Bu sayede dinamik konvolüsyonlu ağlar, benzer işlem maliyetine sahip ve eğitim aşamasında güncellenen, çıkarım aşamasında sabit tutulan kernel parametreleri ile işlem yapan statik konvolüsyonlu derin öğrenme ağları ile karşılaştırıldığında, daha ayırt edici öznitelikler çıkarılabilmesine ve bu sayede daha yüksek performans sağlanmasına olanak tanır.

Tez kapsamında, dinamik konvolüsyonun kişi tanılama (person re-identification (Re-ID)) uygulamasında kullanılması önerilmektedir. Kişi tanılama, bir kişiye ait görüntünün, aynı kişinin farklı kameralardan farklı zamanlarda alınan görüntüleri ile eşlenmesi olarak tanımlanabilir. Ayırt edici özniteliklerin çıkarımı, kişi tanılama modellerinin kamera bakış açısı değişimi, ortamdaki ışıklılık değişimi, kişilerin benzer kıyafetler giymiş olması, kısmi örtüşme gibi eşlemeyi zorlaştırıcı etkilere karşı gürbüzlüğünü sağlayabilmek açısından önemlidir. Literatürde bu amaçla, yerel ve genel özniteliklerin birlikte kullanımı, vücut parçaları bazında çıkarılan özniteliklerin kullanımı, koşullu özniteliklerin kullanımı gibi birçok farklı yöntem önerilmiştir. Literatürdeki kişi tanılama modellerinde statik konvolüsyonlu ağ mimarileri kullanılmıştır ve bilindiği kadarıyla dinamik konvolüsyonlu ağlar ile bir gerçekleme henüz bulunmamaktadır.

Tez kapsamında kullanılan kişi tanılama ağlarında dinamik omurga ağ mimarisi, literatürde bulunan kanal tümleştirme mekanizmalı dinamik konvolüsyon ile gerçeklenmektedir. Bu yaklaşımda her bir katmanda, öncelikle, katman girişindeki öznitelik haritasının kanal sayısı eğitim aşamasında öğrenilen bir statik konvolüsyon kerneli kullanılarak düşürülür. Elde edilen kanal sayısı düşük öznitelik haritasına, katman girişinin genel bilgileri kullanılarak elde edilen kanal tümleştirme matrisi uygulanır ve kanallar, giriş verisine bağlı şekilde tümleştirilir. Sonrasında bir statik konvolüsyon kerneli ile öznitelik haritasının kanal sayısı, istenilen çıkış kanal sayısına yükseltilir. Böylece, giriş verisine adapte edilmiş ağ parametreleri kullanılarak daha ayırt edici öznitelikler elde edilebilmektedir.

Tez çalışmasında kişi tanılama ağı olarak öncelikle literatürde Baseline olarak adlandırılan basit ve az katmanlı bir ağ mimarisi kullanılmıştır. Buna ek olarak son dönemde önerilmiş yüksek performanslı CaceNet kişi tanılama modelinde dinamik konvolüsyonun performansa etkileri incelenmiştir. Baseline mimarisi, omurga ağın sorgulama görüntüsünden çıkardığı öznitelik haritaları üzerinde karşıt-ilinti ve üçlü-kayıp fonksiyonları kullanarak eğitilen temel bir tanılama mimarisidir. CaceNet mimarisi sorgulama görüntüsünün yanı sıra karşılaştırma yapılan ikinci bir görüntüyü kullanmaktadır. Baseline'dan farklı olarak CaceNet, her iki görüntüden omurga ağ tarafından çıkarılan öznitelik haritalarının yanı sıra görüntülerden elde edilen koşullu öznitelik haritalarını da kullanarak, uçtan uca bir eğitim gerçeklemektedir. Koşullu öznitelik haritalarının çıkarımında, görüntü çiftinde görüntü içi ve görüntüler arası farklılıkların yoğunlaştığı öznitelik bölgelerinin belirlenmesi, eşlenmesi ve bir benzerlik grafı üzerinden modellenmesine dayalı işlem adımları bulunmaktadır. Böylece insan beyninde olduğu gibi, sorgu görüntüsünün ilgi noktaları, karşılaştırıldığı görüntü de göz önüne alınarak belirlenir. Bu sayede giriş görüntülerinin hem genel hem de birbirlerine göre koşullu öznitelikleri kullanılarak oldukça ayrıntılı bir karşılaştırma yapılır.

Derin öğrenme ile kişi tanılamada öznitelik çıkarımında omurga ağ mimarisi önem taşımaktadır. Bu nedenle tez çalışmasında Baseline ve CaceNet mimarilerindeki ResNet-50 omurga ağlarında dinamik konvolüsyon kullanılmıştır. Bu amaçla kanal tümleştirme mekanizmalı dinamik konvolüsyon kullanılarak ILSVRC-2012 veri setinde sınıflandırıcı olarak eğitilmiş ResNet-50 omurga ağı (DY-ResNet-50) referans olarak alınmıştır. Statik konvolüsyonlu Baseline (ST-BL) kişi tanılama ağı omurga ağ mimarisindeki konvolüsyon katmanları dinamik konvolüsyon katmanları ile değiştirilerek DY-BL kişi tanılama ağı elde edilmiştir. Benzer değişiklikler statik konvolüsyonlu CaceNet (ST-Cace) kişi tanılama ağı omurga mimarisinde de yapılarak DY-Cace kişi tanılama ağı elde edilmiştir. DY-BL ve DY-Cace ağları farklı kişi tanılama veri setleri kullanılarak uçtan uca eğitilmiştir. ILSVRC-2012 veri setinde eğitilen DY-ResNet-50 modeli, sınıflandırıcı katmanı atılarak, DY-BL ve DY-Cace ağlarının eğitiminde ön-eğitilmiş model olarak kullanılmıştır. Karşılaştırma amacıyla ST-BL ve ST-Cace ağlarının da uçtan uca eğitimi gerçeklenmiştir. Kişi tanılama ağlarının öğrenme performansları eğitimde kullanılan farklı kişileri sınıflandırma performansı olarak, başarım oranı (success rate (SR)) metriği ile raporlanmıştır. Kişi tanılama modellerinin çıkarım performansını incelemek amacıyla, eğitim mimarilerinde bulunan sınıflandırıcı katmanları atılarak, çıkarım mimarileri elde edilmiştir. Kişi tanılama performansı ortalama hassasiyet (mean average precision (mAP)) ve top-k metrikleri ile raporlanmıştır. Eğitim ve çıkarım adımlarında, zorluk dereceleri birbirinden farklı olan Market-1501, DukeMTMC-reID, CUHK03 ve Occluded-DukeMTMC veri setleri kullanılmıştır.

Raporlanan sonuçlara göre, 80 döngü (epoch (ep)) eğitilmiş modellerle çıkarım sonucunda çoğu durumda DY-BL'de kişi tanılama performansının, ST-BL ile karşılaştırıldığında önemli ölçüde yüksek olduğu gözlemlenmiştir. Dinamik omurga ağ mimarisi kullanımı, Market-1501, DukeMTMC-reID, CUHK03 ve Occluded-DukeMTMC veri setleri ile çıkarımda mAP bazında sırasıyla %1,12, %2,31, %0,73 ve %2,24 artış sağlamıştır. top-1 bazında ise Market-1501, DukeMTMC-reID ve Occluded-DukeMTMC veri setlerinde sırasıyla %0,63, %1,93 ve %3,08 artış

görülürken CUHK03 veri setinde %0,57'lik bir düşüş görülmüştür. ST-BL ve DY-BL ağları arasında öğrenme performansları bakımından önemli bir farklılık görülmemiştir.

ST-Cace ve DY-Cace ağlarında ise 80 döngü eğitim sonunda öğrenme ve çıkarım performanslarının, kullanılan performans metrikleri bazında benzer olduğu görülmüştür. Çıkarım adımında ise CUHK03 ve Occluded-DukeMTMC veri setlerinde, aynı kimliğe ait sorgu ve galeri görüntülerinin öznitelikleri arasındaki uzaklıkların sorgu görüntülerinin büyük bir çoğunluğu için DY-Cace'de, ST-Cace'yle karşılaştırıldığında daha düşük olduğu raporlanmıştır. CUHK03 veri setinde 1.400 sorgu görüntüsünün 1.392 tanesinde, aynı kimlikten galerilerle ortalama eşleme uzaklıkları ST-Cace'yle karşılaştırıldığında DY-Cace'de daha düşüktür. Occluded-DukeMTMC'de ise 2.210 sorgu görüntüsünün 2.142 tanesinde, dinamik omurga kullanımı sayesinde eşleme uzaklıklarının düştüğü gözlemlenmiştir. Benzer bir durum Market-1501 veri seti ile çıkarımda da gözlemlense de DY-Cace'de, ST-Cace'ye kıyasla daha düşük uzaklıkla eşlenen sorgu görüntüsü sayısının tüm sorgu görüntülerine oranı, CUHK03 ve Occluded-DukeMTMC veri setlerinde gözlemlenen oranlarla karşılaştırıldığında daha azdır. Market-1501 veri setinde 3.368 sorgu görüntüsünün 2.714 tanesi, dinamik omurga ağ mimarisi kullanıldığında daha düşük ortalama uzaklıklarla eşlenmiştir. DukeMTMC-reID veri seti ile çıkarımda ise uzaklık bazında DY-Cace ve ST-Cace ağları arasında belirgin bir fark gözlemlenememiş olup 2.228 sorgu görüntüsünün 1.290 tanesinde aynı kimlikle eşleme uzaklıkları dinamik omurgalı mimaride statik omurgalı mimariye kıyasla daha düşüktür. Bu durum, CaceNet'te dinamik omurga ağ mimarisi kullanımının, CUHK03, Occluded-DukeMTMC ve Market-1501 ile çıkarımda eşleme güvenini arttırdığını gösterir. Ayrıca CUHK03 veri seti ile eğitim ve çıkarımda farklı döngülerde karşılaştırmalar yapıldığında hem eğitim sırasında hem de çıkarımda performansların, kullanılan metrikler bazında ilk döngülerde ST-Cace'yle karşılaştırıldığında DY-Cace'de daha yüksek olduğu görülmüştür. Fakat ilerleyen döngülerde her iki ağda da SR, mAP ve top-1 değerleri benzer seviyelere ulaşmıştır. Bu durum, Market-1501, DukeMTMC-reID ve Occluded-DukeMTMC veri setleri ile eğitim ve çıkarım adımlarında gözlemlenmemiş olup eğitim boyunca ve farklı döngülerde çıkarımlarda performanslar benzer seviyelerdedir. Dinamik omurga ağ mimarisi kullanımının, Baseline gibi ayırt ediciliği sınırlı öznitelikler kullanan basit ağ mimarilerinde çoğu durumda performansı önemli ölçüde arttırabileceği gözlemlenmiştir.

Tez kapsamında yapılan eğitimlerde ve çıkarım testlerinde aynı veri setinden görüntüler kullanılmıştır. Farklı veri setlerini birlikte kullanarak yapılacak dinamik eğitim ile performans arttırılabilir. Dinamik ağlar son dönemde yazılımsal ve donanımsal gerçeklemede önemli bir araştırma alanını oluşturmaktadır. Bu kapsamda konvolüsyon katmanlarının yanı sıra aktivasyon fonksiyonları gibi işlemlerin de girişe göre özelleştirilmesini sağlayan öğrenme modelleri ve mimarileri geliştirilmektedir. Tez kapsamında gerçeklenen ağlara bu mimariler eklenerek ağın giriş verisine adaptifliği daha da arttırılabilir.

DEEP LEARNING VIA DYNAMIC CONVOLUTION WITH CHANNEL FUSION MECHANISM

SUMMARY

In recent years deep learning networks are widely used in areas such as autonomous vehicles, natural language processing, and medical imaging. The most important factor of these networks in terms of performance is the feature maps that represent the big data correctly. Dynamic convolutional neural networks are proposed in the literature to increase performance which is strongly related to the more distinctive feature maps. In dynamic convolutional neural networks, kernels are depending on the input data and the parameters of the kernel are updated adaptively. Hence dynamic convolutional networks can extract more distinctive feature maps compared to the static version whose kernel parameters are updated in the training phase and kept constant in the inference phase. This thesis proposes using dynamic convolutional neural networks in the person re-identification application. Person re-identification, one of the most common applications of pattern recognition, aims to match different images of the same person from different cameras. The main challenges in person re-identification are illumination changes, viewpoint changes, similar clothing, and occlusions. These challenges are very common in real-world problems and it is very important to have distinctive features to overcome these issues. In the literature, some works try to match the samples by making comparisons in detail or by specifying the more distinctive features for comparison and suppressing less distinctive features. However, it is observed that using dynamic convolutional networks for person re-identification is not common in the literature. Therefore, the effects of dynamic convolution on the person re-identification networks are investigated in this thesis. First, the backbone network architecture of two different person re-identification networks is implemented using dynamic convolution and then compared with their static counterparts. Dynamic backbone architecture is implemented by using the channel-fusion dynamic convolution method from the literature. In this method, in each layer, the feature map in the layer input is projected into a lower dimensional space by using a static convolution filter that is learned in the training phase. In this projected space, channels are fused by using the channel fusion matrix which is obtained from the information of the layer input. Then, the number of channels of the feature map is increased to the number of output channels with the help of another static convolution filter. In this way, it is possible to achieve more distinctive features with a negligible increase in the computational cost regarding the static counterpart of the network. In this thesis, first, a simple network with less number of layers that is named Baseline in the literature is used as a person re-identification network. In addition, the effects of dynamic backbone architecture in the CaceNet are investigated. Baseline architecture is an architecture that does matching as a result of less detailed comparisons concerning features maps extracted by backbone architecture. On the

other hand, CaceNet is a complex architecture that does matching as a result of very detailed comparisons. In addition to the feature maps that are extracted from backbone architecture, CaceNet also uses relative feature maps of the image pairs. Relative feature maps are obtained from the important pixel pairs between both intra-images and inter-images and they specify the main focus regions of the images. Hence, similar to the human brain, interest points of query images are specified by considering the images that are compared with query images. Hence, using both general and conditional feature maps of the input images makes the comparison very detailed.

In the person re-identification with deep learning, backbone network architecture is very important in extracting the feature maps. In this thesis, the ResNet-50 network pre-trained in the ILSVRC-2012 dataset as a classifier is used in both backbones of the person re-identification networks of this work. Convolutional layers of the ResNet-50 architecture are changed with channel-fusion dynamic convolutional layers. The CaceNet which uses dynamic backbone network architecture is named as DY-Cace and the CaceNet which uses static backbone network architecture is named as ST-Cace. Similarly, the Baseline network is also named differently for static and dynamic backbones, ST-BL and DY-BL, respectively.

Cacenet architecture consists of three different stages. In the individual feature embedding stage, individual feature maps are extracted by using the backbone network. Then these individual features are fed into the visual clue alignment stage which is the second stage of the network. In the visual clue alignment stage, the important pixel pairs both intra-images and inter-images are selected from input image pairs. In the conditional feature embedding stage which is the third stage of the network, by using these important pixel pairs, the conditional feature maps. Therefore, the points of interest of the target object are created regarding the compared image, similar to the human brain. In this way, it makes a very detailed comparison using the global, local and conditional information of the images.

In Baseline architecture, first, to obtain the statistical features of the channels, global average pooling and global maximum pooling are applied to the feature map of the output of the ResNet-50's fifth layer. After that, the pooled features are concatenated. Then, this concatenated feature is transferred to a convolutional layer to have reduced dimension, and batch normalization is applied to the output of the convolutional layer. So Baseline architecture uses only the global information of feature maps which is extracted by the backbone architecture and makes a very basic comparison.

DY-BL and DY-Cace networks are trained end-to-end by using different person re-identification databases. The DY-ResNet 50 model that is trained ILSVRC-2012 dataset is used as a pre-trained model in the training of ST-BL and ST-Cace after removing its classifier layer. For comparison purposes, the end-to-end train of ST-BL and ST-Cace networks is also completed. In the training phase, the number of learnable parameters for ST-Cace and DY-Cace is 30,398,176 and 34,447,208 respectively. For ST-BL and DY-BL, these numbers are 26,655,296 and 30,704,328 respectively.

Market-1501, DukeMTMC-reID, CUHK03, and Occluded-DukeMTMC datasets are used for train and inference and the effects of feature maps that are obtained with dynamic convolution on the performance of object classification and person re-identification are reported in detail. The metric of classification performance is

the success rate (SR) and the metric of person re-identification is both mean average precision (mAP) and top-k.

The training and inference performances of ST-Cace and DY-Cace networks are compared after training for 80 epochs. Even though the results look similar in terms of performance metrics, it is observed that the distances between the features of the same identities are lower in the DY-Cace than ST-Cace for many query images during the inference for CUHK03 and Occluded-DukeMTMC datasets. A similar situation is also observed in the Market-1501 dataset. However, the ratio of matched queries with lower distances and all queries was lesser than CUHK03 and Occluded-DukeMTMC datasets. These results show that using dynamic backbone network architecture increases the confidence of matching at the inference phase. Additionally to these, a comparison of different epochs in the CUHK03 dataset is investigated and it is found that both in training and inference, classification and identification performances (SR, mAP, and top-1) of DY-Cace are higher than ST-Cace in the first epochs. However, in the following epochs, the value of these metrics become similar. Even though there isn't a big difference in the learning performances of ST-BL and DY-BL networks, it can be said that DY-BL achieved better inference results than ST-BL for 80 epoch-trained models. The usage of dynamic backbone network architecture increased the mAP by 1.12%, 2.31%, 0.73%, and 2.24% for Market-1501, DukeMTMC-reID, CUHK03, and Occluded-DukeMTMC datasets, respectively. For the top-1 metric, a 0.63% increase in Market-1501, a 1.93% increase in DukeMTMC, a 3.08% increase in Occluded-DukeMTMC, and finally 0.57% decrease in the CUHK33 dataset are obtained.

In the ST-CAce and DY-Cace networks, the performance comparison of training and inference after 80 epochs resulted very similar in terms of performance metrics. In the inference phase of CUHK03 and Occluded-DukeMTMC databases, the distance between feature maps of query and gallery images for the same identity is observed shorter for most of the query images in DY-Cace than ST-Cace. For example, in the CUHK03 dataset, 1392 of 1400 query images have a lower average matching distance with equivalent gallery images in DY-Cace than ST-Cace. In the Occuleded-DukeMTMC database, 2142 of 2210 query images has shorter matching distance due to the usage of dynamic backbone architecture. Even though similar behavior is observed in inference with the Market-1501 database, the number of query images that matched in a shorter distance in DY-Cace than ST-Cace is lesser than in CUHK03 and Occluded-DukeMTMC database. In the Market-1501 database, 2714 of 3368 query images are matched in a shorter average distance due to dynamic backbone architecture. In the inference with the DukeMTMC-reID dataset, the average distance between query images in DY-Cace and ST-Cace don't have any significant differences. 1290 of 2228 query images matched in a shorter distance in DY-Cace than ST-Cace. This observation shows that using dynamic backbone architecture in CaceNet increases the matching confidence in the inference with CUKHK04, Occluded-DukeMTMC, and Market-1501 databases. Besides, in the training and inference phases with the CUHK03 database, DY-Cace has better performance than ST-Cace in the first epochs in terms of performance metrics. However, in the proceeding epochs, both of the networks reached the same performance in the metrics of SR, mAP, and top-1. This behavior is not observed in the train and inference phases.

Market-1501, DukeMTMC-reID and Occluded-DukeMTMC database. However, it can be claimed that using dynamic backbone architecture increases the performance of simple networks such as “Baseline” whose feature maps are less distinctive and limited.

In the training and inference tests conducted within the scope of this thesis, images from the same database are used. The performance of cross-database tests does not result in good enough. Dynamic training by using different databases together can increase performance.

In recent years, dynamic networks became an important research area both in hardware and software implementation of neural networks. It is also possible to develop a learning model or architecture by changing the operations such as activation functions to be specialized concerning input data, similar to what is performed on convolutional layers in this thesis.

1. GİRİŞ

Tez çalışması kapsamında dinamik konvolüsyonlu derin öğrenme ağları kullanılarak kişi tanılama problemi üzerinde çalışılmıştır. Kişi tanılama (Re-ID) , bir kişiye ait sorgulanan görüntünün, aynı kişinin farklı kameralardan farklı zamanlarda alınan görüntüleri ile eşlenmesi olarak tanımlanabilir. Re-ID birçok güvenlik uygulamasında gerek duyulan temel bir işlemdir. Güvenilir bir Re-ID sisteminin, kamera bakış açısındaki değişikliler, görüntülenen ortamdaki ışıklılık değişimleri, izlenen kişinin bir başka kişi ya da nesne tarafından örtülmesi, ortamda izlenene benzer kıyafetli kişiler olması gibi zorluklar altında istenilen performansı sağlaması beklenmektedir. Bu nedenle sorgulanan görüntü ile olası eşleme görüntüleri arasındaki benzerliğin ayrıntılı bir şekilde modellenebilmesi önemlidir.

Literatürde görüntü sınıflandırma ve kişi tanılama amacıyla çok farklı yöntemler önerilmiş olmakla birlikte, son yıllarda derin öğrenme ile konvolüsyonel sinir ağları kullanılarak Re-ID gerçekleyen yöntemler yaygınlık kazanmıştır [20]–[22]. Kişi tanılama ağları, genel olarak her kimlik bir sınıf olmak üzere, sınıflandırıcı olarak eğitilir. Eğitimde kullanılan kayıp fonksiyonları, aynı kimliğe ait görüntüler için birbirine daha yakın, farklı kimliklere ait görüntüler için birbirine daha uzak özniteliklerin öğrenilmesini sağlayacak şekilde seçilir. Çıkarım aşamasında ise verilen bir sorgu görüntüsü, öznitelikleri en yakın galeri görüntülerine eşlenir.

Çoğu uygulamada sorgulanan görüntünün genel özniteliklerinin yanında daha ayrıntılı özniteliklerin de aramaya eklenmesi gerekir. Bu amaçla sorgu görüntüsündeki her bir bölgenin ya da her bir vücut parçasının öznitelikleri, olası eşlenebilir görüntülerin toplandığı veri kümesindeki (galeri) kişilerden çıkarılan bölgesel öznitelikler ile karşılaştırılmaktadır [19]–[22]. [19]'da insan duruş veri seti (human pose dataset) ile eğitilen bir bölge öneri ağı (region proposal network (RPN)) ile insan vücudundaki ana bölgeler belirlenir. Sonrasında giriş görüntüsünün ve belirlenen ana bölgelerin öznitelik vektörleri elde edilir ve ağ sonunda bu vektörler birleştirilerek çıkış

öznitelik vektörü elde edilir. Bu yöntemde ağın performansı, önemli derecede RPN performansına bağlıdır. [20]'de sorgu görüntüsü için omurga ağ çıkışında elde edilen öznitelik haritaları eşit alt bölgelere bölünür ve her bir alt bölgeye karşı düşen öznitelikler ile galeri görüntülerinden benzer şekilde elde edilen öznitelikler arasında eşleme yapılır. Bu yöntemde alt bölgelerin sorgulanan görüntüye göre küçük ya da büyük seçilmesi, kişinin bütünsel olarak eşlenememesine yol açabileceğinden, kritik rol oynamaktadır. [21]'de çok dallı bir yapıyla hem genel hem de yerel öznitelikler çıkartılarak birleştirilir ve çıkış öznitelik haritası elde edilir. Bu durumda sorgulanan görüntünün hem genel hem yerel bilgilerinden yararlanılmış olsa da genel ve yerel bilgiler arasındaki ilişkinin nasıl kurulacağı önem kazanmaktadır. [22]'de kullanılan kademeli yapıda ilk olarak öznitelik haritaları [20]'deki gibi alt bölgeler için hesaplanır, ardından her bir bölge en yakın komşu bölge ile birleştirilerek her bir kademede daha geniş bölgeler oluşturulur. Bu şekilde oluşturulan öznitelik benzerlik ağacı üzerinden sorgulama ve eşleme yapılır. Bu sayede genel ve yerel bilgilerin yanında farklı ölçekli bilgilerden de yararlanılmış olur. Önerilen yöntemin işlemsel yükü fazladır ve duruş değişimlerine karşı dayanıklılığı düşüktür.

Bu çalışmada kullanılan derin öğrenmeye dayalı Re-ID mimarilerinden ilki Baseline olarak adlandırılmaktadır [26]. Baseline (ST-BL), ResNet-50 omurga ağı [3] ile çıkarılan özniteliklerden Re-ID özniteliklerini öğrenen az katmanlı bir derin ağ mimarisidir. Kullanılan ikinci Re-ID ağı CaceNet [5,10], yukarıda bahsedilen mimariler gibi görüntülerin genel ve yerel özniteliklerini kullanmanın yanı sıra koşullu öznitelikleri de kullanmaktadır. CaceNet (ST-Cace), bahsedilen mimarilerden farklı olarak, ayırt edici bölgeleri önceden belirlenmiş kurallarla değil sorgu görüntüsü ile olası eşlenecek görüntü arasındaki koşullu öznitelikleri ağ içinde bulunan bir "ilgililik dikkat modülü (correspondence attention module)" ile seçer. Bu sayede kişi görüntülerini karşılaştırırken, görüntülerin ayrı ayrı özniteliklerini karşılaştırmanın yanında insan beyninin çalışma mantığıyla paralel olarak ilk görüntüdeki odak noktalarını, ikinci görüntünün içeriğine göre değiştirir. Bu sayede hem genel hem yerel hem de koşullu görsel ipuçlarını verimli bir şekilde kullanarak detaylı karşılaştırmalar sonucunda eşlemeler yapar.

Yüksek doğruluklu eşleme yapan kişi tanılama sistemlerinin tasarımında derin öğrenmede kullanılacak ayırt edici özniteliklerin çıkarılması çok önemlidir. Daha ayırt edici öznitelikler elde etmek amacıyla literatürde dinamik konvolüsyonlu derin öğrenme ağları önerilmiştir. Giriş verisine bağlı olarak ağ elemanlarında değişiklikler yapan dinamik ağların bir alt kümesi olan dinamik konvolüsyonlu derin öğrenme ağları, giriş verilerinden dinamik şekilde bilgiler çıkarır ve bu bilgilerden yararlanarak ilgili veriye uygulanacak konvolüsyon kernellerinin parametrelerini giriş verisine göre özelleştirir [1,6,7]. Dinamik konvolüsyonlu ağlar, her bir giriş verisi için özelleştirilmiş konvolüsyon kernelleri kullanması sayesinde benzer işlemsel karmaşıklığa sahip statik konvolüsyonlu ağlarla karşılaştırıldığında giriş verisinin özniteliklerini daha ayırt edici biçimde çıkarır. Böylece muadili olan statik CNNlerden daha yüksek performans gösterebilir [2].

Konvolüsyon parametrelerinin dışında diğer ağ elemanları veya ağın mimarisi de farklı amaçlarla giriş verisine adapte olacak şekilde özelleştirilebilir. [8]'de bir aktivasyon fonksiyonu olan "doğrultulmuş lineer birim (rectified linear unit - ReLU)" fonksiyonunun eğimleri, giriş verisinden elde edilen genel bilgiler kullanılarak adaptif biçimde ayarlanır. Böylece işlem yükünde önemli bir artış olmadan ağın temsil gücü arttırılır. Anlamsal bölütleme (semantic segmentation) yapan [9]'da ise çıkarım aşamasında kolay veriler için gereksiz işlemler yapmaktan kaçınmak amacıyla giriş verisinin zorluğuna göre ağ derinliği değiştirilir. Mimaride bulunan birden çok çıkışın her birinde pikseller için tahminler yapılır ve piksel tahminlerinin güven puanı (confidence score) hesaplanır. Yüksek tahmin güvenilirliğine sahip piksellerin yüzdesi belirli bir sınırın üzerindeyse veri ağdan çıkarılır, değilse daha derin katmanlara gönderilir. Böylece çıkarım aşamasında yapılacak işlem sayısı, giriş verisine göre dinamik şekilde ayarlanır ve performansta bir düşüş olmadan işlem yükü azaltılabilir. Bütün bu avantajları ve uygulama yöntemlerinin sınırsızlığı nedeniyle giriş verisine göre adaptif şekilde işlem yapan dinamik ağlar, son dönemde sıkça kullanılmaktadır.

Tez kapsamında, dinamik konvolüsyonun kişi tanılama uygulamasında kullanılması önerilmektedir. Bu amaçla kişi tanılama ağlarının omurga mimarilerinde Bölüm 2.2'de ayrıntılı olarak anlatılan kanal tümleştirme mekanizmalı dinamik konvolüsyon [2] kullanılmıştır. Kanal tümleştirme mekanizmalı dinamik konvolüsyon, Bölüm 2.1'de

anlatılan kernel ve kanal tümleştirme ile dinamik konvolüsyona [1,6] benzese de, bazı avantajları bulunmaktadır. Kernel ve kanal tümleştirme ile dinamik konvolüsyonda, her bir katmanda eğitim aşamasında öğrenilen K adet konvolüsyon kerneli bulunur. Bu K adet kernelin her biri, giriş verilerinin genel öznitelikleri kullanılarak öğrenilen K adet ağırlıkla ağırlıklandırılarak toplanır ve katman kerneli elde edilir. Bu sayede her bir giriş verisine, veriye göre özelleştirilmiş konvolüsyon kernelleri uygulanır ve çıkarılan özniteliklerin temsil gücü arttırılır. Ancak K adet kernel kullanılması, ağın parametre sayısının statik konvolüsyonel katmanlar kullanımına göre K kat artması sonucunu doğurmaktadır. Bunun yanı sıra kernel tümleştirmede kullanılan ve girişe bağlı olarak öğrenilen ağırlıkları üreten dinamik dikkat modelinin ve konvolüsyon kernellerinin ortak optimizasyonu zordur.

Tez kapsamında kullanılan kanal tümleştirmeli dinamik konvolüsyon yönteminde ise ana katman kerneli, "dinamik konvolüsyon ayrıştırma" yöntemi kullanılarak bileşenleri cinsinden ifade edilir ve katman girişine bu bileşenler uygulanır. Öncelikle katman girişi, statik bir konvolüsyon kerneli ile daha düşük boyutlu bir uzaya izdüşürülür ve düşük boyutlu uzayda girişe bağlı adaptif bir kanal tümleştirme matrisi ile kanallar tümleştirilir. Sonrasında elde edilen öznitelik haritasının kanal sayısı, başka bir statik konvolüsyon kerneli kullanılarak çıkış kanal sayısına yükseltilir. Kanal tümleştirmeli dinamik konvolüsyon, kernel ve kanal tümleştirme yöntemi ile yapılan dinamik konvolüsyonla karşılaştırıldığında, hem daha az öğrenilebilir parametre kullanır hem de ağın optimizasyonu daha kolaydır.

Tez çalışmasında, yukarıda bahsedilen ST-BL ve ST-Cace ağlarının ResNet-50 omurga ağ mimarisi (ST-ResNet-50) kanal tümleştirme mekanizmalı dinamik konvolüsyon (DY-ResNet-50) omurga ağı ile değiştirilerek, sırasıyla, DY-BL ve DY-Cace kişi tanılama ağları gerçeklenmiş ve hem statik omurgalı hem de dinamik omurgalı ağların, zorluk dereceleri farklı veri setlerinde uçtan uca 80 döngü eğitimleri yapılmıştır. Bu amaçla [2]'de gerçeklenen ILSVRC-2012 veri setinde [11] sınıflandırıcı olarak eğitilmiş ResNet-50 modeli sınıflandırıcı katmanları atılarak, dinamik omurgalı Re-ID mimarilerinin eğitiminde ön-eğitilmiş model olarak kullanılmıştır. Benzer şekilde statik omurgalı Re-ID ağlarında da ILSVRC-2012 veri setinde ön-eğitilmiş ResNet-50 ön-eğitilmiş model olarak kullanılmıştır.

Re-ID ağlarının öğrenme performansları, eğitimde farklı kimlikli kişilerin sınıflandırma performansı olarak ele alınmış ve raporlamada başarım oranı (success rate (SR)) metriği kullanılmıştır. Modellerin çıkarım performanslarını raporlamak amacıyla, eğitim mimarilerinden sınıflandırıcı katmanları atılarak çıkarım mimarileri oluşturulmuştur. Re-ID performansı raporlamada ortalama hassasiyet (mean average precision (mAP)) ve top-k metrikleri kullanılmıştır. Eğitim ve çıkarım aşamalarında Market-1501 [12], DukeMTMC-reID [13], CUHK03 [14] ve Occluded-DukeMTMC [15] veri setleri kullanılmıştır. Yapılan eğitim ve çıkarımlar "Google Colaboratory" ortamında yapılmış olup yazılımda Python programlama dili kullanılmıştır.

Elde edilen sonuçlara göre, 80 döngü eğitilmiş modellerle çıkarım sonucunda çoğu durumda DY-BL'de kişi tanılama performansının, ST-BL'ye göre daha yüksek olduğu raporlanmıştır. Dinamik omurga ağ mimarisi kullanımı, Market-1501, DukeMTMC-reID, CUHK03 ve Occluded-DukeMTMC veri setleri ile çıkarımda mAP bazında sırasıyla %1,12, %2,31, %0.73 ve %2,24 artış sağlamıştır. top-1 bazında ise Market-1501, DukeMTMC-reID ve Occluded-DukeMTMC'de sırasıyla %0,63, %1,93 ve %3,08 artış raporlanırken CUHK03'te %0,57'lik bir düşüş raporlanmıştır. ST-BL ve DY-BL ağlarında öğrenme performansları bakımından önemli bir farklılık görülmemiştir.

ST-Cace ve DY-Cace ağlarında ise 80 döngü eğitim sonunda öğrenme ve çıkarım performanslarının, kullanılan metrikler bazında benzer olduğu görülmüştür. Çıkarım adımında ise CUHK03 ve Occluded-DukeMTMC veri setlerinde aynı kimliğe ait sorgu ve galeri görüntülerinin öznitelikleri arasındaki uzaklıkların, sorgu görüntülerinin büyük bir kısmı için ST-Cace ile karşılaştırıldığında DY-Cace'de daha düşük olduğu görülmüştür. CUHK03 ve Occluded-DukeMTMC veri setlerinde, sorgu görüntülerinin sırasıyla %99,43 ve %96,92'sinin ST-Cace ile karşılaştırıldığında DY-Cace'de aynı kimlikten galeri görüntülerine daha düşük ortalama uzaklıklarla eşlendiği görülmüştür. Benzer bir durum Market-1501 veri seti ile çıkarımda da gözlemlense de ST-Cace'ye kıyasla DY-Cace'de, daha düşük uzaklıklarla eşlenen sorgu görüntüsü sayısının toplam sorgu görüntüsü sayısına oranı, CUHK03 ve Occluded-DukeMTMC veri setlerindeki oranlarla karşılaştırıldığında daha azdır. Market-1501 veri setinde bu

oran %80,58'dir. DukeMTMC-reID veri seti ile çıkarımda ise DY-Cace ve ST-Cace ağları arasında eşleme uzaklığı bazında önemli bir farklılık görülmemiş olup ortalama eşleme uzaklığı düşen sorgu görüntüsü sayısının tüm sorgu görüntülerine oranı %57,90'dır. Bu durum, CaceNet'te dinamik omurga ağ mimarisi kullanımının, CUHK03,Occluded-DukeMTMC ve Market-1501 veri setleri ile çıkarımda eşleme güvenini arttırdığını gösterir.

Ayrıca CUHK03 veri seti ile eğitim ve çıkarım aşamalarında farklı döngülerde karşılaştırmalar yapıldığında hem eğitim sırasında hem de çıkarımda performansların, metrikler bazında ilk döngülerde ST-Cace'yle karşılaştırıldığında DY-Cace'de daha yüksek olduğu raporlanmıştır. İlerleyen döngülerde ise her iki ağda da SR, mAP ve top-1 değerleri benzer seviyelere ulaşmıştır. Market-1501, DukeMTMC-reID ve Occluded-DukeMTMC veri setleri ile eğitim ve çıkarımda ise benzer bir durum gözlemlenmemiş olup eğitim boyunca ve farklı döngülerde çıkarımlarda performansların benzer olduğu görülmüştür.

Sonuç olarak, Baseline gibi ayırt ediciliği sınırlı öznitelikler kullanan basit ağ mimarilerinde dinamik omurga ağ mimarisi kullanımının, çoğu durumda performansı önemli arttırabileceği gözlemlenmiştir. CaceNet gibi ayırt ediciliği yüksek özniteliklerle detaylı karşılaştırmalar yapan ağlarda ise dinamik omurga ağ mimarisi kullanımının performansa katkısının sınırlı olduğu görülmüştür.

Tez içeriğinde Bölüm 2'de tez kapsamında kullanılan kanal tümleştirme mekanizmalı dinamik konvolüsyonun teorik alt yapısı ve kullanılan dinamik omurga ağ mimarisinin detayları verilmektedir. Bölüm 3'te dinamik omurga ağ mimarisi ile gerçeklenen kişi tanılama ağları anlatılmaktadır. Bölüm 4'te dinamik omurga ağ mimarisi ile gerçeklenen kişi tanılama ağlarında gözlemlenen performanslar, statik omurga kullanan kişi tanılama ağlarında gözlemlenen performanslarla karşılaştırmalı şekilde raporlanmaktadır. Son olarak Bölüm 5'te ise sonuçlar ve tartışma sunulmaktadır.

2. DİNAMİK KONVOLÜSYONLU DERİN ÖĞRENME AĞLARI

Statik konvolüsyonlu derin öğrenme ağları, eğitim aşamasında giriş verisine bağlı olarak konvolüsyon kernellerini öğrenir ve çıkarım aşamasında öğrenilen ağ parametrelerinde güncelleme yapmaz. Son dönemde gerçeklenen dinamik konvolüsyonlu ağ mimarilerinde önerilen temel yaklaşım ise aynı anda birden fazla kernel öğrenerek daha ayırdedici özniteliklerin çıkarılmasıdır. Eğitim aşamasında birden fazla kernel parametresi öğrenilirken bunların tümleştirilmesinde kullanılacak ağırlıklandırma parametreleri de öğrenilir. Çıkarım aşamasında ağırlıklandırma parametreleri giriş verisine göre güncellenerek dinamik bir öğrenme gerçeklenir.

Literatürde kernel ve kanal tümleştirme gerçekleyen ya da sadece kanal tümleştirme gerçekleyen farklı konvolüsyon modelleri önerilmiştir. Bölüm 2.1'de kernel ve kanal tümleştirme ile dinamik konvolüsyon formülasyonu verildikten sonra Bölüm 2.2'de tez kapsamında kullanılan kanal tümleştirme ile dinamik konvolüsyon ayrıntılı açıklanmaktadır.

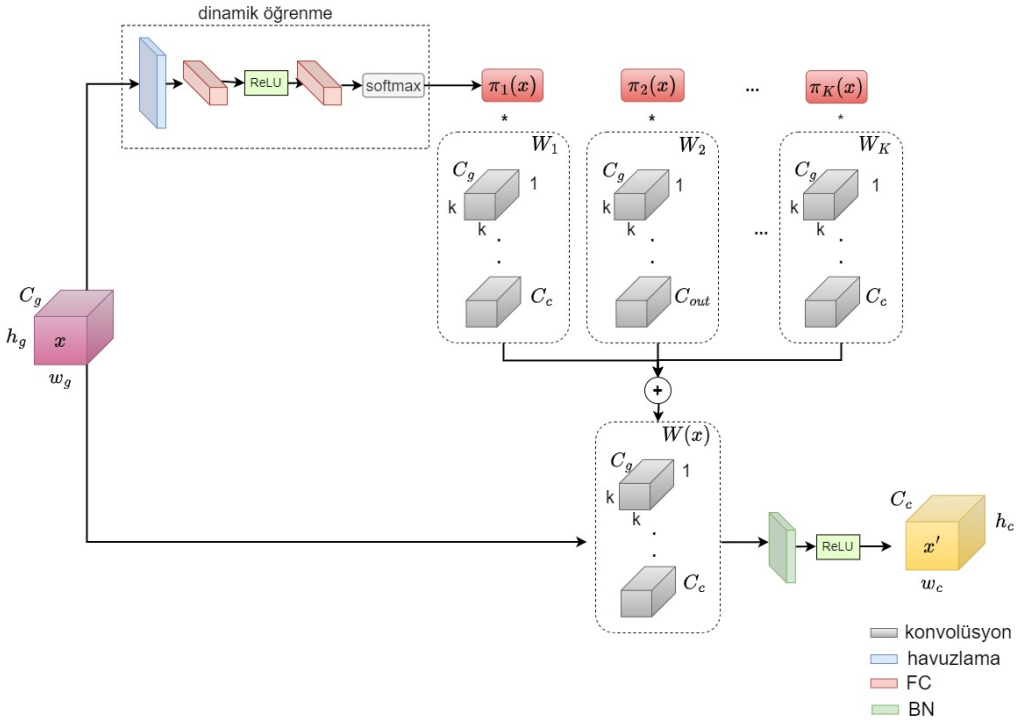
2.1 Kernel ve Kanal Tümleştirme ile Dinamik Konvolüsyon

Bir katmanda birden çok kernelin girişe göre ağırlıklandırılarak kullanılması fikrine dayanan dinamik konvolüsyon yapısında [1,6], her bir katman için K tane statik konvolüsyon kerneli öğrenilir. K kernelin herbiri ile çıkarılan özniteliklerden elde edilecek sonuç öznitelik haritası, tümünün ağırlıklı toplamı şeklinde modellenir. Ağırlıklandırmada kullanılan "dinamik dikkat (attention) katsayıları" da eğitim sırasında öğrenilir. Bu modelleme, sonuç öznitelik haritasının, Denklem 2.1'de görülen K kernelin ağırlıklı toplamı olan tek bir kernel ile giriş verisinin konvolüsyonu ile elde edileceği şeklinde yorumlanabilir.

$$\mathbf{W}(\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x}) \mathbf{W}_k \tag{2.1}$$

Denklem 2.1'de görülen $\mathbf{W}(\mathbf{x})$, katman girişine uygulandığı varsayılan ana kernel, $\pi_k(\mathbf{x})$'ler katmandaki her bir kerneli ağırlıklandıran ve çıkarım aşamasında güncellenerek her bir kernelinin girişe göre özelleştirilmesini sağlayan dinamik dikkat katsayıları, $\mathbf{W}_k$'ler ise eğitim aşamasında öğrenilen, çıkarım aşamasında sabit kalan statik konvolüsyon kernelleridir.

Tanımlanan dinamik konvolüsyonda, statik kerneller ve onları tümleştiren dikkat modeli ($\pi_k(.)$) eğitim aşamasında öğrenilir. Öğrenilen dikkat modeli ile giriş verisine bağlı dinamik dikkat katsayıları elde edilir ve statik kerneller dinamik dikkat katsayılarıyla ağırlıklandırılarak toplanır. Böylece farklı kerneller, her bir giriş için konvolüsyon işlemine farklı seviyede etki eder. Şekil 2.1'de tanımlanan dinamik konvolüsyonun gerçeklenmesinde kullanılan ağ mimarisi verilmiştir.



Şekil 2.1 : Kernel ve kanal tümleştirme gerçekleyen dinamik konvolüsyon katmanı mimarisi ([1]'den uyarlanmıştır).

Kernel ve kanal tümleştirme yaklaşımıyla gerçeklenen dinamik konvolüsyonun işlem yükü çok fazladır. Her katmanda K tane konvolüsyon kerneli bulunduğundan eğitilebilir parametre sayısı statik konvolüsyon ile karşılaştırıldığında K kat daha

fazladır. Öte yandan eğitim sırasında hem dinamik dikkat modelinin ($\pi_k(.)$) hem de statik konvolüsyon kernellerinin ($\mathbf{W}_k$) ortak optimizasyonu zordur.

Parametre uzayını küçültmek amacıyla [6]'da dinamik dikkat katsayılarının Sigmoid katmanı çıkışında elde edilmesi önerilmiştir. Bu durum, dikkat katsayılarını $0 \leq \pi_k(\mathbf{x}) \leq 1$ olacak şekilde sınırlar. Fakat bu sınırlamaya rağmen $\mathbf{W}(\mathbf{x})$'in parametre uzayı çok geniştir ve bu durum dikkat modelinin öğrenilmesini zorlaştırır. Parametre uzayını küçültmek amacıyla [1]'de dikkat katsayılarının Softmax fonksiyonu çıkışında elde edilmesi önerilmiştir. Böylece Softmax fonksiyonunun doğası gereği $0 \leq \pi_k(\mathbf{x}) \leq 1$ sınırlamasına ek olarak $\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1$ sınırlaması da getirilmiş olur. Bu sınırlama, kernel uzayını daraltarak dikkat modelinin öğrenilmesini kolaylaştırır. Fakat Softmax fonksiyonunun doğası gereği bazı dikkat katsayıları çok küçük olabilir. Bu da eğitim aşamasında küçük ağırlıklarla ağırlıklandırılan statik kernellerin öğrenilmesini engelleyebilir. Bu yüzden ek sınırlamalara ihtiyaç duyulur. Örneğin [1]'de ilk döngülerde Softmax fonksiyonu büyük sıcaklık (temperature) değerleriyle kullanılmıştır (Denklem 2.2) ve sıcaklık değerleri kademeli olarak azaltılarak kullanılan Softmax fonksiyonu, orijinal Softmax fonksiyonuna indirgenir (sıcaklık değeri ilk 10 döngüde lineer olarak 30'dan 1'e düşürülür). Böylece dikkat katsayılarının, eğitimin ilk aşamalarında birbirine daha yakın olması sağlanır.

$$\pi_k(\mathbf{x}) = \frac{exp(z_k/\tau)}{\sum_j exp(z_k/\tau)} \tag{2.2}$$

Denklem 2.1'de görülen dinamik konvolüsyonda kullanılan her bir statik kernel $\mathbf{W}_k$, Denklem 2.3'teki gibi bir ortalama kernel ve statik kernelin ortalamadan sapma miktarını belirten bir artık kernelin toplamı olarak tanımlanabilir [2].

$$\mathbf{W}_k = \mathbf{W}_0 + \Delta\mathbf{W}_k, \quad k \in 1, ..., K \tag{2.3}$$

Burada ortalama kernel $\mathbf{W}_0 = \frac{1}{K}\sum_{k=1}^{K} \mathbf{W}_k$, k. statik kernelin ortalamadan sapma miktarı olan k. artık kernel ise $\Delta\mathbf{W}_k = \mathbf{W}_k - \mathbf{W}_0$ olarak tanımlanır. Böylece Denklem 2.3 kullanılarak Denklem 2.1, Denklem 2.4 şeklinde tekrar yazılabilir [2].

$$\mathbf{W}(\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x})\mathbf{W}_0 + \sum_{k=1}^{K} \pi_k(\mathbf{x})\Delta\mathbf{W}_k \tag{2.4}$$

Statik kernellerin artık kernelleri $\Delta\mathbf{W}_k$'ler, tekil değer ayrıştırma (SVD) [18] ile ayrıştırılarak birden çok kernelin, giriş değerlerine bağlı oluşturulan dinamik dikkat katsayılarıyla ağırlıklandırılarak toplanması işleminin farklı bir matematiksel gösterimi elde edilebilir.

Denklem 2.4'teki artık kerneller, SVD kullanılarak $\Delta\mathbf{W}_k = \mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T$ şeklinde ayrıştırılabilir ve Denklem 2.4, Denklem 2.5 şeklinde ifade edilebilir [2].

$$\mathbf{W}(\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x})\mathbf{W}_k = \sum_{k=1}^{K} \pi_k(\mathbf{x})\mathbf{W}_0 + \sum_{k=1}^{K} \pi_k(x)\mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T \tag{2.5}$$

Dinamik konvolüsyon katmanı giriş ve çıkış kanal sayılarının eşit olduğu ($C_g = C_c = C$) ve katman ana kernel boyutunun 1×1 olduğu varsayılırsa, Denklem 2.5'te görülen $\mathbf{U}_k$ matrisi, sütunları $\Delta\mathbf{W}_k$ matrisinin sol tekil değerleri olan C×C boyutlu ortonormal bir matristir. $\mathbf{S}_k$ matrisi, köşegen elemanları $\Delta\mathbf{W}_k$ matrisinin tekil değerleri olan C×C boyutlu bir köşegen matristir. $\mathbf{V}_k$ ise sütunları $\Delta\mathbf{W}_k$ matrisinin sağ tekil değerleri olan C×C boyutlu ortanormal bir matristir.

$\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1$ sınırlaması kullanılarak Denklem 2.5'deki toplamın ilk terimi Denklem 2.6'da görüldüğü gibi açık şekilde yazılabilir.

$$\sum_{k=1}^{K} \pi_k(\mathbf{x})\mathbf{W}_0 = \pi_1(\mathbf{x})\mathbf{W}_0 + \pi_2(\mathbf{x})\mathbf{W}_0 + ... + \pi_K(\mathbf{x})\mathbf{W}_0 \tag{2.6}$$

$w_{i,j}$, $\mathbf{W}_0$ matrisinin i. satır j. sütun elemanı olmak üzere, $\mathbf{W}_0$ Denklem 2.7 ile gösterilir.

$$\mathbf{W}_0 = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,C} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ w_{C,1} & w_{C,2} & \cdots & w_{C,C} \end{bmatrix} \tag{2.7}$$

$\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1$ koşulu altında $\mathbf{W}_0$ matrisinin herhangi bir elemanının dinamik dikkat katsayılarıyla çarpımı Denklem 2.8'deki gibi hesaplanabilir.

$$\pi_1(\mathbf{x})w_{i,j} + ... + \pi_K(\mathbf{x})w_{i,j} = w_{i,j}.(\pi_1(\mathbf{x}) + ... + \pi_K(\mathbf{x})) = w_{i,j} \tag{2.8}$$

Denklem 2.8'den görüldüğü üzere, $\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1$ koşulu altında dinamik dikkat katsayılarıyla çarpım, $\mathbf{W}_0$ matrisinin herhangi bir elemanında değişime neden olmaz. Böylece Denklem 2.5'deki ilk toplam terimi $\mathbf{W}_0$'a indirgenmiş olur (Denklem 2.9) [2].

$$\sum_{k=1}^{K} \pi_k(\mathbf{x})\mathbf{W}_0 = \mathbf{W}_0 \tag{2.9}$$

Denklem 2.5'deki ikinci toplam terimi (artık kısım) ise Denklem 2.10'daki gibi açılabilir.

$$\sum_{k=1}^{K} \pi_k(\mathbf{x})\mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T = \pi_1(\mathbf{x})\mathbf{U}_1\mathbf{S}_1\mathbf{V}_1^T + ... + \pi_K(\mathbf{x})\mathbf{U}_K\mathbf{S}_K\mathbf{V}_K^T \tag{2.10}$$

Denklem 2.10'daki k. konvolüsyon kerneli için C×C matris ($\mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T$) ve skaler $\pi_k(\mathbf{x})$'in çarpımı, $\mathbf{I}_C$ C×C birim matris olmak üzere matris çarpımı şeklinde düzenlenebilir (Denklem 2.11).

$$\pi_k(\mathbf{x})\mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T = \mathbf{U}_k\pi_k(\mathbf{x})\mathbf{I}_C\mathbf{S}_k\mathbf{V}_k^T \tag{2.11}$$

Böylece Denklem 2.10, matris çarpımı şeklinde yazılarak Denklem 2.12 elde edilir [2].

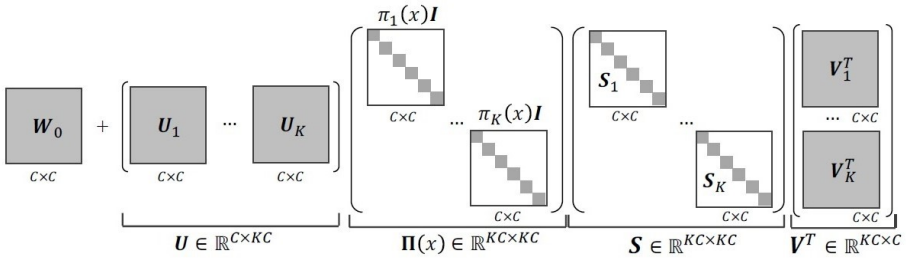
$$\sum_{k=1}^{K} \pi_k(\mathbf{x})\mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T = \sum_{k=1}^{K} \mathbf{U}_k\pi_k(\mathbf{x})\mathbf{I}\mathbf{S}_k\mathbf{V}_k^T = \mathbf{U}\mathbf{\Pi}(\mathbf{x})\mathbf{S}\mathbf{V}^T \tag{2.12}$$

Burada $\mathbf{U} = [\mathbf{U}_1, ..., \mathbf{U}_K]$, $\mathbf{\Pi}(\mathbf{x}) = diag(\pi_1(\mathbf{x}), ..., \pi_K(\mathbf{x}))$, $\mathbf{S} = diag(\mathbf{S}_1, ..., \mathbf{S}_K)$ ve $\mathbf{V} = [\mathbf{V}_1, ..., \mathbf{V}_K]$'dir.

Böylece Denklem 2.9 ve Denklem 2.12 kullanılarak $\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1$ koşulu altında Denklem 2.5, Denklem 2.13 şeklinde yeniden yazılabilir [2].

$$\mathbf{W}(\mathbf{x}) = \mathbf{W}_0 + \mathbf{U}\mathbf{\Pi}(\mathbf{x})\mathbf{S}\mathbf{V}^T = \mathbf{W}_0 + \sum_{i=1}^{KC} \pi_{\lceil i/C \rceil}(\mathbf{x})\mathbf{u}_i s_{i,i}\mathbf{v}_i^T \tag{2.13}$$

Burada $\mathbf{u}_i$, $\mathbf{U} \in \mathbb{R}^{C \times KC}$ matrisinin i. sütun vektörü, $s_{i,i}$ $\mathbf{S} \in \mathbb{R}^{KC \times KC}$ matrisinin i. köşegen elemanı ve $\mathbf{v}_i^T$ ise $\mathbf{V}^T \in \mathbb{R}^{KC \times C}$ matrisinin i. satır vektörüdür. Burada CxC sırasıyla çıkış ve giriş kanal sayılarını göstermektedir. 2.13'deki işlemin matris gösterimi Şekil 2.2'de görülebilir.



Şekil 2.2 : Kernel ve kanal tümleştirmeli dinamik konvolüsyonda ana kernelin matris gösterimi [2].

Denklem 2.13'de eşitliğin sol tarafından görüldüğü üzere, dinamik katman girişe uygulandığında giriş ($\mathbf{x} \in \mathbb{R}^C$) önce "gizli uzay (latent space)" olarak adlandırılan daha yüksek boyutlu bir uzaya izdüşürülür ($\mathbf{SV}^T\mathbf{x} \in \mathbb{R}^{KC}$) ve kanal sayısı C'den KC'ye çıkarılır. Sonrasında izdüşürülen yüksek boyutlu uzayda K tane kanal grubuna (KC tane kanala) dinamik dikkat katsayıları ($\pi_k(\mathbf{x})$) uygulanarak kanal grupları ağırlıklandırılır. Son olarak $\mathbf{\Pi}(\mathbf{x})\mathbf{SV}^T\mathbf{x}$'ye $\mathbf{U} \in \mathbb{R}^{KC \times C}$ uygulanarak çıkış kanal sayısı, giriş kanal sayısı ile aynı olacak şekilde tekrar C'ye düşürülür.

Fakat kanal gruplarının dinamik dikkat katsayıları ile ağırlıklandırılması, girişi yüksek boyutlu bir uzaya izdüşürerek işlemlerin bu uzayda yapılmasını gerektirdiğinden statik konvolüsyonla karşılaştırıldığında parametre sayısında büyük bir artışa neden olur. Ayrıca Denklem 2.13'te eşitliğin sağ tarafından görüldüğü üzere statik baz vektörleri $\mathbf{u}_i$ ve $\mathbf{v}_i$'ler, her bir i değeri için farklıdır ve $\pi_{\lceil i/C \rceil}(\mathbf{x})$'ler farklı i değerleri için aynı değeri alabilir. Bu durum optimizasyonu zorlaştırır. Çünkü küçük dikkat katsayıları, ilgili statik baz vektörleri $\mathbf{u}_i$ ve $\mathbf{v}_i$'lerin öğrenilmesini engelleyebilir. Özellikle eğitimin ilk aşamalarında bu durum önemli bir sorundur. Bu problemlerin hafifletilmesi amacıyla [2]'de, bu yaklaşımda yapılan işleme eşdeğer bir işlemin yeni bir matris ayrıştırma yöntemi ile yapılması önerilmektedir (Bölüm 2.2).

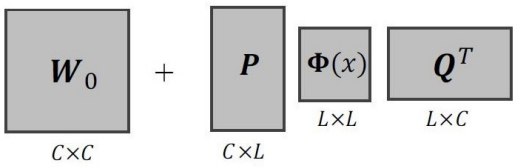
2.2 Kanal Tümleştirme Yaklaşımı ile Dinamik Konvolüsyon

Kernel ve kanal tümleştirme ile dinamik konvolüsyon, ağın temsil gücünü arttırmasına rağmen, statik konvolüsyonla karşılaştırıldığında parametre sayısında büyük bir artışa ve optimizasyon zorluğuna neden olur. Optimizasyonu kolaylaştırmak amacıyla getirilen Bölüm 2.1'de bahsedilen sınırlamaları ortadan kaldırmak ve parametre sayısındaki artışı hafifletmek için [2]'de dinamik konvolüsyonun, kanal gruplarının dinamik dikkat katsayılarıyla ağırlıklandırılarak gerçeklenmesi yerine, "dinamik kanal tümleştirme (dynamic channel fusion)" ile gerçeklenmesi önerilmiştir.

Bu amaçla [2]'de Denklem 2.3'teki artık kernel ($\Delta\mathbf{W}_k$), SVD yerine "dinamik konvolüsyon ayrıştırma (DCD)" ile ayrıştırılmış ve Bölüm 2.1'de anlatılan kernel ve kanal tümleştirme yaklaşımıyla karşılaştırıldığında girişin izdüşürüldüğü uzayın boyutu düşürülerek ağın parametre sayısı azaltılmıştır. DCD kullanılarak artık kernel $\Delta\mathbf{W}_k = \mathbf{P}\boldsymbol{\Phi}(\mathbf{x})\mathbf{Q}^T$ olarak tanımlanır. Böylece Denklem 2.4'teki ana kernel $\mathbf{W}(\mathbf{x})$, $\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1$ sınırlaması altında Denklem 2.14 şeklinde ifade edilebilir [2].

$$\mathbf{W}(\mathbf{x}) = \mathbf{W}_0 + \mathbf{P}\boldsymbol{\Phi}(\mathbf{x})\mathbf{Q}^T = \mathbf{W}_0 + \sum_{i=1}^{L}\sum_{j=1}^{L} \mathbf{p}_i \phi_{i,j}(\mathbf{x}) \mathbf{q}_j^T \tag{2.14}$$

Burada $\mathbf{W}_0 \in \mathbb{R}^{C\times C}$ ortalama kernel, $\mathbf{p}_i$ statik matris $\mathbf{P} \in \mathbb{R}^{C\times L}$'nin i. sütun vektörü, $\phi_{i,j}(\mathbf{x})$ $\boldsymbol{\Phi}(\mathbf{x}) \in \mathbb{R}^{L\times L}$'nin i. satır j. sütun elemanı ve $\mathbf{q}_j^T$ ise statik matris $\mathbf{Q}^T \in \mathbb{R}^{L\times C}$'nin j. satır vektörüdür. Kanal tümleştirme mekanizmalı dinamik konvolüsyonda ana kernelin matris gösterimi Şekil 2.3'te görülebilir.



Şekil 2.3 : Kanal tümleştirme mekanizmalı dinamik konvolüsyonda ana kernelin matris gösterimi ($L \ll C$) [2].

Denklem 2.14'te eşitliğin sol tarafının 2. teriminden görüldüğü gibi, $C \ll L$ olmak üzere kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanı girişe

uygulandığında giriş ($\mathbf{x} \in \mathbb{R}^C$), önce daha düşük boyutlu bir gizli uzaya izdüşürülür ($\mathbf{Q}^T\mathbf{x} \in \mathbb{R}^L$) ve kanal sayısı C'den L'ye düşürülür. Sonrasında gizli uzaydaki L tane kanal, girişe bağlı dinamik kanal tümleştirme matrisi $\mathbf{\Phi}(\mathbf{x})$ ile tümleştirilerek ağırlıklandırılır. Son olarak $\mathbf{\Phi}(\mathbf{x})\mathbf{Q}^T\mathbf{x}$'e $\mathbf{P}$ uygulanarak çıkış kanal sayısı, giriş kanal sayısı ile aynı olacak şekilde L'den tekrar C'ye yükseltilir. Burada L, C'den çok daha küçüktür. Bu durum, kernel ve kanal tümleştirme yaklaşımıyla karşılaştırıldığında parametre sayısında önemli bir düşüş sağlar. [2]'de gizli uzay boyutu L, $L < C^2$ olarak sınırlandırılır ve varsayılan değeri $\lfloor \frac{C}{2^{L\log_2\sqrt{C}}} \rfloor$'dir.

Ayrıca kanal tümleştirme mekanizmalı dinamik konvolüsyonda dinamik matris $\mathbf{\Phi}(\mathbf{x})$, kernel ve kanal tümleştirme yaklaşımıyla dinamik konvolüsyonda kullanılan köşegen dinamik matris $\mathbf{\Pi}(\mathbf{x})$'in aksine seyrek (sparse) bir matris değildir. Burada dinamik kanal tümleştirme matrisi $\mathbf{\Phi}(\mathbf{x})$'in her elemanı girişe bağlı bir sayıdır. Bu yüzden Denklem 2.14'te eşitliğin sağ tarafından görüldüğü üzere $\mathbf{p}_i$ ve $\mathbf{q}_i$ vektörleri birden çok $\phi_{i,j}$ değeriyle ilişkilidir. Örneğin $\mathbf{p}_i$ vektörü $\phi_{i,1}$, $\phi_{i,2}$, ... , $\phi_{i,L}$ ile, $\mathbf{q}_j^T$ vektörü ise $\phi_{1,j}$, $\phi_{2,j}$, ... , $\phi_{L,j}$ değerleriyle ilişkilidir. Bu durum Bölüm 2.1'de bahsedilen küçük dinamik dikkat katsayıları nedeniyle statik matrislerin öğrenilmesinin engellenmesi sorununu ortadan kaldırır ve optimizasyonu kolaylaştırır. Böylece [1]'deki durumun aksine eğitimin ilk aşamalarında Softmax katmanında büyük sıcaklık değerleri kullanılmasına gerek kalmaz.

2.2.1 Kanal bazlı dikkat mekanizması ile formülasyonun genelleştirilmesi

Bu bölüme kadar işlemler $\sum_{k=1}^{K} \pi_k(\mathbf{x}) = 1$ sınırlaması kullanılarak yapılmış ve Denklem 2.4'daki ilk toplam terimi, $\sum_{k=1}^{K} \pi_k(\mathbf{x})\mathbf{W}_0 = \mathbf{W}_0$ şeklinde basitleştirilmiştir. Bu sınırlama ortadan kaldırılırsa Denklem 2.4'daki ilk toplam terimi de giriş değerlerine bağlı hale gelir. Bu durumda ana kernel Denklem 2.15 şeklinde tanımlanır [2].

$$\mathbf{W}(\mathbf{x}) = \mathbf{\Lambda}(\mathbf{x})\mathbf{W}_0 + \mathbf{P}\mathbf{\Phi}(\mathbf{x})\mathbf{Q}^T \tag{2.15}$$

Burada $\mathbf{\Lambda}(\mathbf{x}) \in \mathbb{R}^{C\text{x}C}$ köşegen bir matristir ve her bir köşegen elemanı $\lambda_{i,i}(\mathbf{x})$ girişin bir fonksiyonudur. $\mathbf{\Lambda}(\mathbf{x})$, her bir kanalın girişe bağlı değerlerle ağırlıklandırılmasını

sağlar ve bu sayede ağın temsil gücü daha da artar. $\mathbf{I}_{C\times C}$ $C\times C$ birim matris olmak üzere Denklem 2.14, Denklem 2.15'nın $\mathbf{\Lambda}(\mathbf{x}) = \mathbf{I}_{C\times C}$ için özel bir formudur.

Kanal bazlı dikkat matrisi ($\mathbf{\Lambda}(\mathbf{x})$) ve dinamik kanal tümleştirme matrisi ($\mathbf{\Phi}(\mathbf{x})$), dinamik öğrenme kolu ile elde edilir. Dinamik öğrenme kolunda ilk olarak bir ortalama havuzlama katmanı ile giriş kanallarının genel öznitelikleri çıkarılır ve bir FC katmanı ile boyut $1/r$ oranında düşürülür. Sonrasında paralel iki farklı FC katmanı ile $\mathbf{\Lambda}(\mathbf{x}) \in \mathbb{R}^{C\times C}$ köşegen matrisinin elemanları için C değer, $\mathbf{\Phi}(\mathbf{x}) \in \mathbb{R}^{L\times L}$ tam matrisinin (full matrix) elemanları içinse L^2 değer olmak üzere çıkışlar oluşturulur. Dinamik öğrenme koluyla elde edilen girişe bağlı $\mathbf{\Lambda}(\mathbf{x})$ ve $\mathbf{\Phi}(\mathbf{x})$ matrisleri ve $\mathbf{W}_0$, $\mathbf{P}$ ve $\mathbf{Q}$ statik kernelleri ile katman çıkışı x' elde edilir.

Sonuç olarak kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanlarında, her bir giriş verisi için girişe göre özelleştirilmiş parametreler kullanılması sayesinde ağın temsil gücü artar. Bu sayede, kanal tümleştirmeli dinamik konvolüsyon katmanları kullanan ağlarda, statik konvolüsyon katmanlı eşlenikleri ile karşılaştırıldığında daha yüksek performanslar raporlanabilir. Ayrıca kanal tümleştirme mekanizmalı dinamik konvolüsyon, katman kernellerinin girişe göre ağırlıklandırılması işlemini kernel ve kanal tümleştirme yaklaşımıyla dinamik konvolüsyondan daha farklı bir perspektifte uygular ve kernel ve kanal tümleştirme yaklaşımıyla karşılaştırıldığında daha az parametre kullanması ve optimizasyon kolaylığıyla önemli bir fark yaratır. Bütün bu avantajları nedeniyle bu çalışmada kanal tümleştirme mekanizmalı dinamik konvolüsyon kullanılmıştır. Literatürde açık bir araştırma alanı olduğu görülen dinamik konvolüsyonla kişi tanılama konusuna katkı sağlamak amacıyla tez kapsamında kanal tümleştirme mekanizmalı dinamik konvolüsyonun, farklı kişi tanılama mimarilerinin omurgalarında kullanıldığında performansa etkileri incelenmiştir.

2.2.2 Dinamik omurga ağ mimarisi ile ayırt edici öznitelik çıkarımı

Tez çalışmasında omurga ağ mimarisi olarak literatürde bulunan ResNet-50 mimarisi [3] kullanılmıştır. Bu bölümde sırasıyla statik konvolüsyon katmanları ile gerçeklenen orijinal ResNet-50 mimarisi (ST-ResNet-50) ve ayırt edici öznitelikler çıkarılması amacıyla [2]'de önerilen kanal tümleştirme mekanizmalı dinamik konvolüsyon

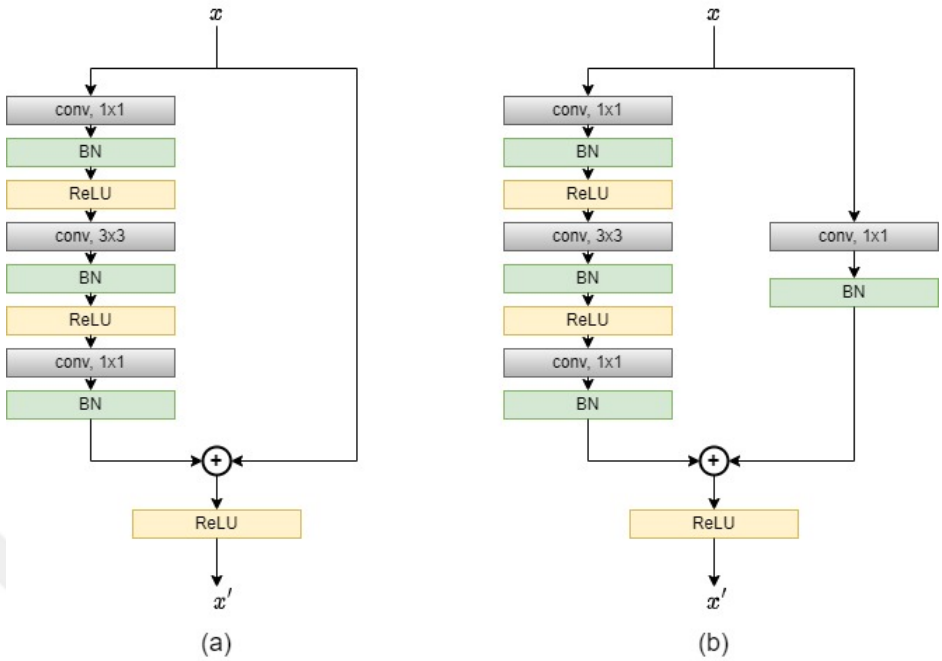
katmanları ile gerçeklenmiş ResNet-50 mimarisi (DY-ResNet-50) anlatılmıştır. Her iki ResNet-50 mimarisi de ILSVRC-2012 [11] veri seti ile 1000 sınıf için sınıflandırıcı olarak eğitilmiş şekilde kullanılmıştır.

ST-ResNet-50 ağının da içinde bulunduğu artık ağlar (residual network (ResNet)), derin ağlarda sıkça görülen gradyanların kaybolması problemini hafifletmek amacıyla önerilmiştir [3]. Gradyanların kaybolması problemi, ağ derinliği arttıkça geri yayılım aşamasında gradyanların değerlerinin çok küçülmesi nedeniyle özellikle ilk katmanların ağırlıklarının güncellenememesi ve ağın eğitilememesi olarak tanımlanabilir.

Artık ağlar, grandyanların kaybolması problemini hafifletmek için atlama bağlantıları (skip connections) kullanır. Artık ağlarda temel bloklar 2 ya da 3 konvolüsyon katmanı içerir ve bu blokların çıktısı, son konvolüsyon katmanı ve blok girişinin toplanması ile elde edilir. Başka bir deyişle atlama bağlantısı kullanmayan mimarilerin aksine artık ağlarda blok çıkışı, kullanılan aktivasyon fonksiyonu F olmak üzere $H(x) = F(wx+b)$ yerine $H(x) = F(wx+b)+x$ olur. Böylece gradyanların ilk katmanlara doğru çok küçülmesi büyük ölçüde engellenerek gradyanların kaybolması problemi hafifletilir.

ST-ResNet-50 ve daha derin artık ağ mimarilerinde her bir temel blok "bottleneck (BON)" olarak adlandırılır ve Şekil 2.4'te görüldüğü gibi konvolüsyon katmanları ve atlama bağlantıları içerir. Her bir BON bloğunda parametre sayısını azaltmak amacıyla girişe öncelikle 1x1 lik bir kernel uygulanarak kanal sayısı düşürülür. Sonrasında düşük kanallı veriye 3x3 lük bir kernel uygulanır ve eleman bazında toplama yapabilmek amacıyla başka bir 1x1 lik kernel ile kanal sayısı, blok girişinin kanal sayısına yükseltilir. Son olarak katman çıkışı, bir atlama bağlantısı ile girişle toplanır ve blok çıkışı elde edilir (Şekil 2.4(a)). Blok girişini direkt olarak blok çıkışına aktaran atlama bağlantıları birim bağlantı (identity connection) olarak adlandırılır. Blok çıkışındaki kanal sayısının, blok girişindeki kanal sayısına eşit olmadığı bloklarda ise blok sonunda toplama işlemini gerçekleştirebilmek amacıyla atlama bağlantısında 1x1 lik bir kernel kullanılarak girişten gelen verinin kanal sayısı, katmanların çıkışındaki verinin kanal sayısına eşitlenir (Şekil 2.4(b)). Bu şekilde 1x1lik kernel ile kanal

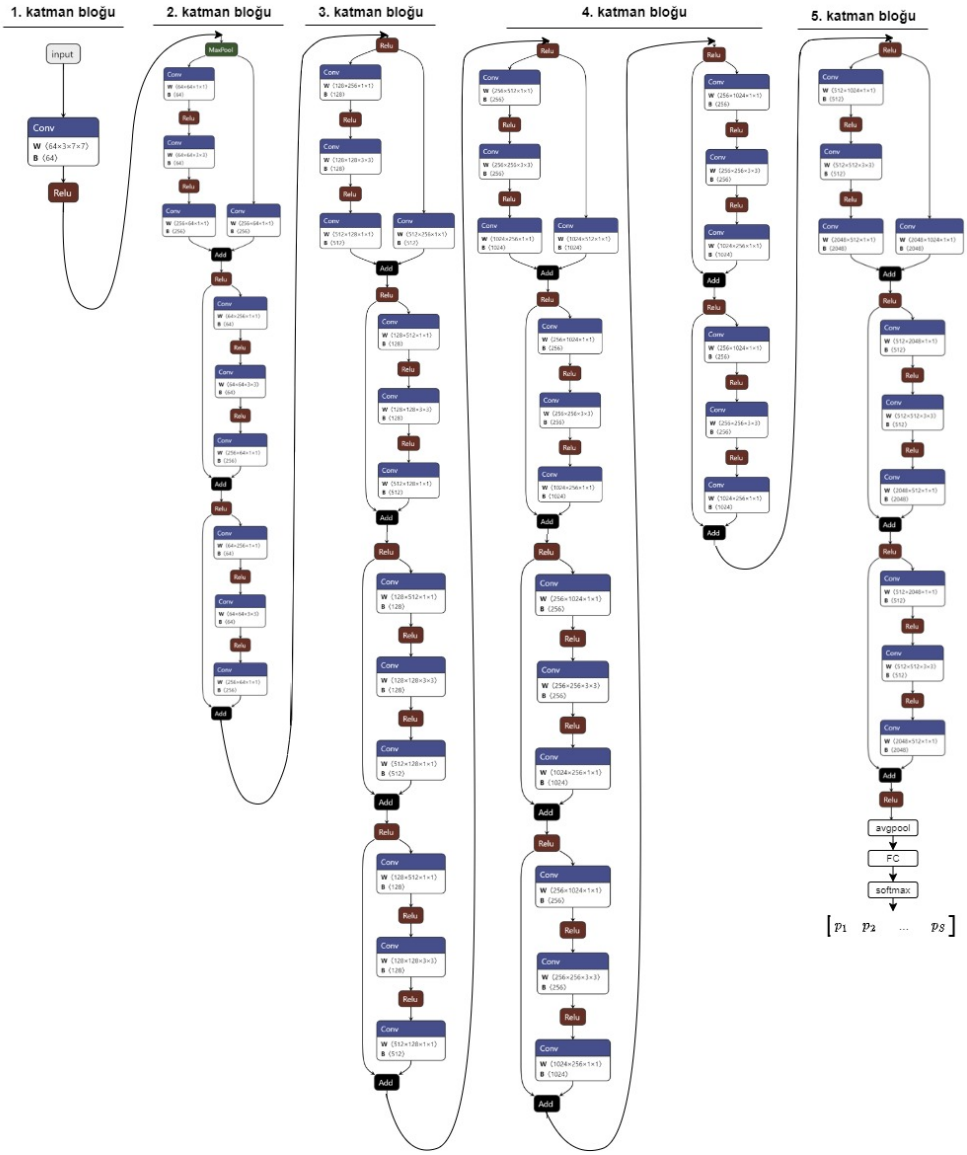
sayısını ayarlayan atlama bağlantıları ise izdüşüm bağlantısı (projection connection) olarak adlandırılır.



Şekil 2.4 : BON temel bloğu. (a) birim bağlantı (b) izdüşüm bağlantısı ([3]'ten uyarlanmıştır).

ST-ResNet-50, toplam 50 katmandan oluşan, 5 ana katman bloğu içeren bir artık ağ mimarisidir. ST-ResNet-50'nin ilk katman bloğunda herhangi bir atlama bağlantısı bulunmaz. 2., 3., 4. ve 5. katman blokları ise sırasıyla 3, 4, 6 ve 3 adet, her biri bir atlama bağlantısı bulunduran BON bloğundan oluşur ve her bir konvolüsyon katmanından sonra "batch" normalizasyonu (BN) ve ReLU aktivasyon fonksiyonu kullanılır. Bu katman bloklarının çıkışları sırasıyla 256, 512, 1024 ve 2048 kanallı öznitelik haritalarıdır.

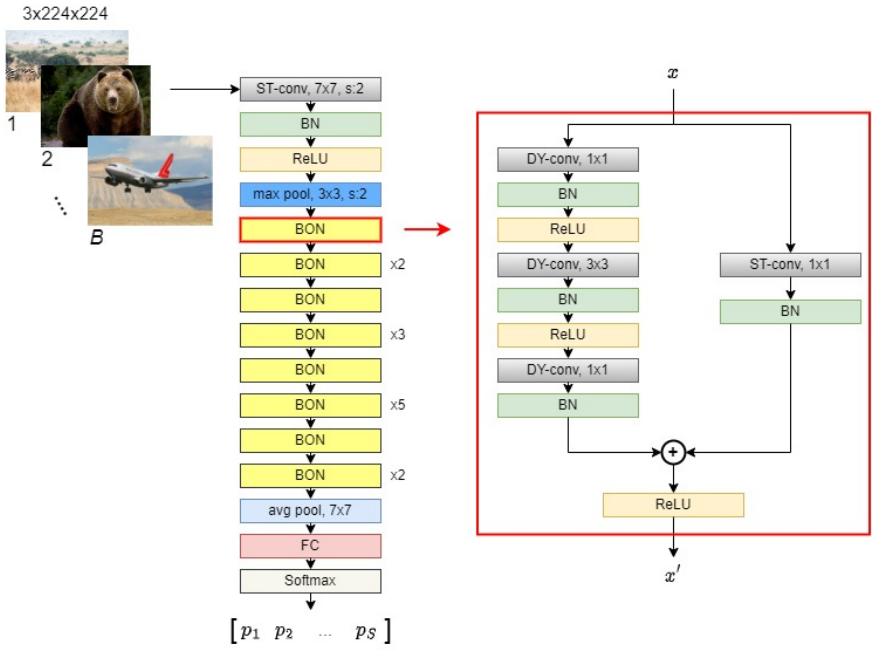
ST-ResNet-50 ile sınıflandırıcı eğitiminde, giriş verilerinin ST-ResNet-50 ile çıkarılan 2048 kanallı öznitelik haritaları, bir havuzlama ve ardından gelen bir FC katmanına gönderilir. FC yi izleyen SsoftMax katmanı çıkışında her bir giriş verisi için, verinin mevcut sınıflara ait olma olasılıkları hesaplanır ($[p_1\ p_2\ ...\ p_S]$) (ILSVRC-2012 veri seti için 1000 sınıf). Bu olasılıklar ve verilerin gerçek referans değerleri (ground truth (GT)) kullanılarak sınıflandırma kaybı hesaplanır ve ağ parametreleri güncellenir. ST-ResNet-50 ile tasarlanmış sınıflandırıcı ağ mimarisi Şekil 2.5'te verilmiştir.



Şekil 2.5 : ST-ResNet-50 sınıflandırıcı ağ mimarisi ([3]'ten uyarlanmıştır).

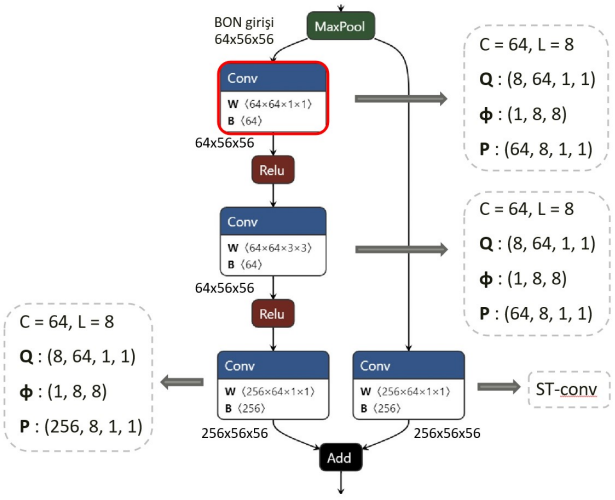
[2]'de daha ayırt edici öznitelikler çıkarılması amacıyla ST-ResNet-50 mimarisinin 2., 3., 4. ve 5. katman bloklarını oluşturan BONların, izdüşüm bağlantıları dışındaki konvolüsyon katmanları, Bölüm 2.2'te anlatılan kanal tümleştirme mekanizmalı dinamik konvolüsyon ile gerçeklenmiştir (DY-ResNet-50). Dinamik konvolüsyon katmanlarının giriş ve çıkış kanal sayıları C_g ve C_c, ST-ResNet-50 mimarisinde karşı düşen statik konvolüsyon katmanlarının giriş ve çıkış kanal sayıları ile aynı olup kanal tümleştirmenin uygulandığı gizli uzay boyutu $L = \lfloor \sqrt{C_g} \rfloor$'dir.

DY-ResNet-50 ile sınıflandırıcı ağ mimarisi ve DY-ResNet-50'nin 2. katman bloğu, 1. BONunun içeriği Şekil 2.6'da verilmiştir.



Şekil 2.6 : DY-ResNet-50 sınıflandırıcı ağ mimarisi.

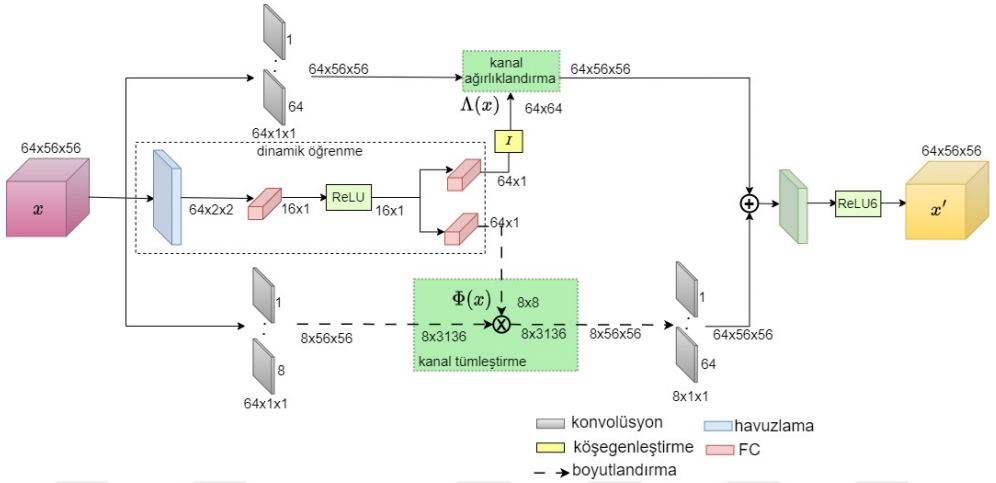
ST-ResNet-50'nin 2. katman bloğu, 1. BONu ve DY-ResNet-50'de ilgili katmanlarda bulunan statik kernellerin (**Q**, **P**) ve kanal tümleştirme matrislerinin ($\mathbf{\Phi}(\mathbf{x})$) boyutları Şekil 2.7'de görülmektedir.



Şekil 2.7 : ST-ResNet-50 ve DY-ResNet-50'nin 2. katman bloğu 1. BONunda kullanılan kernel boyutları.

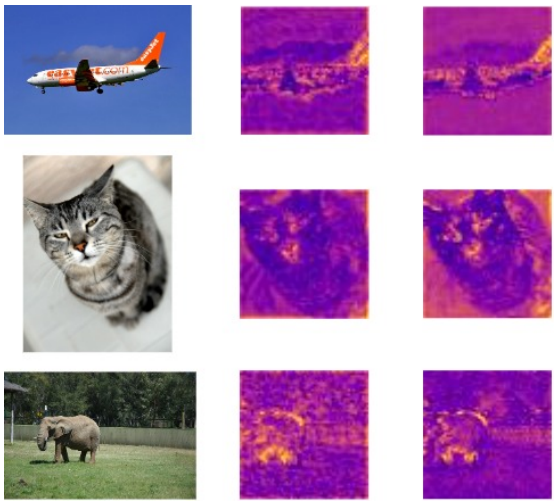
Şekil 2.7'de görüldüğü gibi, ST-ResNet-50'de her katmanda bir kernel kullanılırken DY-ResNet-50'de her katmanda birden çok kernel kullanılarak giriş verisine adaptif şekilde konvolüsyon işlemi yapılır.

Şekil 2.7'de kırmızı sınırlar ile belirtilen katmana karşı düşen, DY-ResNet-50'nin 2. katman bloğu, 1. BONunun ilk kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanı mimarisi Şekil 2.8'de görülmektedir.



Şekil 2.8 : DY-ResNet-50'nin 2. katman bloğu, 1. BONunun 1. kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanı mimarisi ([2]'den uyarlanmıştır).

Statik ve dinamik omurga ağ mimarileri ile gerçeklenen ve ILSVRC-2012 veri setinde sınıflandırıcı olarak eğitilmiş ResNet-50 modellerinin (ST-ResNet-50 ve DY-ResNet-50), ILSVRC-2012 veri setinden seçilen bazı görüntüler için 2. katman bloğu çıkışındaki öznitelik haritaları Şekil 2.9'da görülmektedir. Öznitelik haritaları, 2. katman bloğu çıkışındaki 256 kanalın ortalaması alınarak elde edilmiştir.



Şekil 2.9 : Sınıflandırıcı olarak eğitilmiş ST-ResNet-50 ve DY-ResNet-50 modellerinin ILSVRC-2012 veri setinden seçilen görüntüler için 2. katman bloğu çıkışlarındaki öznitelik haritaları. (a) giriş görüntüsü, (b) ST-ResNet-50 ile elde edilen öznitelik haritaları, (c) DY-ResNet-50 ile elde edilen öznitelik haritaları.

Şekil 2.9(a) giriş görüntüleri, (b) ST-ResNet-50'nin, (c) DY-ResNet-50'nin 2. katman bloğunun çıkışları olmak üzere öznitelik haritaları karşılaştırıldığında dinamik ağdan çıkan özniteliklerin öğrenilen nesneye daha iyi lokalize olduğu görülmektedir.

Dinamik konvolüsyon katmanları ile konvolüsyon kernellerinin, giriş verisinden elde edilen bilgiler kullanılarak ilgili veriye göre özelleştirilmesi sayesinde dinamik omurga ağ mimarisi ile, statik konvolüsyon katmanları kullanan omurga ağ mimarisine göre daha ayırt edici öznitelikler çıkarılabilir.

3. DİNAMİK KANAL TÜMLEŞTİRME İLE KİŞİ TANILAMA

Kişi tanılama, bir kişiye ait sorgu (query) görüntüsünün farklı kameralardan, farklı zamanlarda elde edilen aynı kişiye ait görüntülerle eşlenmesini amaçlar. Kişi tanılama ağları, genel olarak her kimlik bir sınıf olmak üzere, sınıflandırıcı olarak eğitilir. Eğitimde kullanılan kayıp fonksiyonları, aynı kimliğe ait görüntüler için birbirine daha yakın, farklı kimliklere ait görüntüler için birbirine daha uzak özniteliklerin öğrenilmesini sağlayacak şekilde seçilir. Çıkarım aşamasında ise verilen bir sorgu görüntüsü, öznitelikleri en yakın galeri görüntülerine eşlenir (Şekil 3.1).

Şekil 3.1 : Re-ID ağlarında amaç, sorgu görüntülerini ilgili galeri görüntülerine eşlemektir [4].

Tez kapsamında, görüntülerin sadece genel bilgilerini kullanarak eşleme yapan az katmanlı basit bir Re-ID mimarisi ve görüntülerin hem genel hem de koşullu bilgilerini kullanarak eşleme yapan daha karmaşık bir Re-ID mimarisi kullanılmıştır. Re-ID ağlarında, giriş verilerine göre özelleştirilmiş ağ parametreleri kullanılarak

ağın temsil gücünün arttırılmasının, Re-ID gibi ayrıntılı bilgilerin çok önemli olduğu bir alanda performans üzerinde olumlu etkileri olabileceği düşünülmüştür. Literatür incelendiğindeyse, dinamik konvolüsyonla kişi tanılama çalışmalarının, bilindiği kadarıyla, olmadığı görülmüştür. Bu nedenle, iki Re-ID mimarisinin temel öznitelik çıkarımını sağlayan bileşenleri olan omurga ağ katmanları, Bölüm 2.2.2'de anlatılan kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanları ile gerçeklenen omurga ağ mimarisi ile değiştirilmiş ve omurga ağ mimarisinde kanal tümleştirme mekanizmalı dinamik konvolüsyon kullanımının performans üzerindeki etkileri incelenmiştir.

Bölüm 3.1'de tezde kullanılan kişi tanılama mimarileri anlatılmaktadır. Gerçeklenen eğitim mimarileri için kullanılan kayıp fonksiyonları verilerek karşı düşen çıkarım mimarileri ile birlikte açıklanmaktadır.

3.1 Kullanılan Kişi Tanılama Mimarileri

Bu çalışmada, dinamik omurga ağ mimarisi kullanımının Re-ID ağlarında performansa etkilerini ayrıntılı şekilde gözlemleyebilmek amacıyla literatürde bulunan biri basit, diğeri kompleks iki farklı Re-ID mimarisi kullanılmıştır. Bölüm 3.1.1'de anlatılan az katmanlı, basit bir Re-ID mimarisi olan Baseline (ST-BL) [26], giriş görüntülerinin sadece genel bilgilerini kullanarak karşılaştırmalar yapar. Bölüm 3.1.2'de anlatılan CaceNet mimarisi (ST-Cace) [5,10] ise giriş görüntülerinin genel, yerel ve koşullu bilgilerinden yararlanarak detaylı karşılaştırmalar sonucunda eşlemeler yapan güncel bir Re-ID yöntemidir. Kullanılan iki Re-ID mimarisinde de omurga olarak Bölüm 2.2.2'de anlatılan ST-ResNet-50 mimarisi [3] kullanılmaktadır ve bu çalışmada iki Re-ID ağında da omurga mimarisi, kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanları ile gerçeklenen DY-ResNet-50 ile değiştirilerek statik ve dinamik omurgalı kişi tanılama mimarilerinin farklı veri setleriyle eğitimleri yapılmıştır. Sınıflandırıcı olarak tasarlanan ST-ResNet-50 ve DY-ResNet-50 ağ mimarilerinin, Re-ID uygulamasında kullanılması amacıyla son katmanları atılmış ve her iki Re-ID ağında da omurga ağları ILSVRC-2012 veri setinde [11] ön-eğitilmiş şekilde kullanılmıştır.

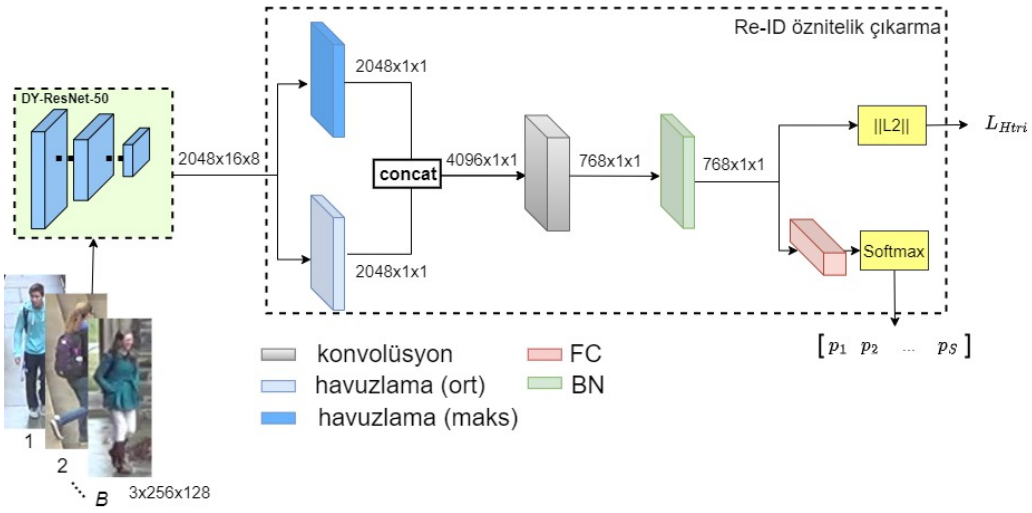
3.1.1 Az katmanlı kişi tanılama ağı eğitim mimarisi

Re-ID ağlarında dinamik omurga mimarisi kullanımının performansa etkilerini gözlemleyerek karşılaştırmalar yapabilmek amacıyla öncelikle [26]'da Baseline olarak adlandırılan (ST-BL) az katmanlı, basit bir Re-ID ağının omurgasında kanal tümleştirme mekanizmalı dinamik konvolüsyonla (Bölüm 2.2) gerçeklenen DY-ResNet-50 ağı (Bölüm 2.2.2) kullanılmıştır. Bu bağlamda, ST-BL'de omurga mimarisi olarak kullanılan ST-ResNet-50'nin ilk katmanı ve izdüşüm bağlantıları dışındaki bütün konvolüsyon katmanları kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanları ile değiştirilmiştir.

ST-BL, 2 farklı kayıp fonksiyonu kullanılarak eğitilen bir Re-ID mimarisidir. Bu mimaride, önce ST-ResNet-50 omurga ağı ile giriş verisinin öznitelik haritası çıkarılır. Sonra öznitelik haritasına, her bir kanalın istatistiksel özelliklerini elde etmek amacıyla global ortalama havuzlama ve global maksimum havuzlama uygulanarak çıkışlar birleştirilir (concatenate). Birleştirilen çıkış, boyut düşürmek amacıyla bir konvolüsyon katmanına aktarılır ve konvolüsyon katmanı çıkışına "batch" normalizasyonu (BN) uygulanır. Eğitim sırasında her bir giriş verisi için elde edilen 768 boyutlu öznitelik vektörleri ile veriler arası L2 uzaklıkları elde edilerek bu uzaklıklarla zor üçlü-kayıp (L_{Htri}) (Denklem 3.6) hesaplanır. 768 boyutlu vektörlerin FC katmanı ve Softmax fonksiyonundan geçirilmesi ile elde edilen giriş verilerinin sınıflara (Re-ID özelinde kimlikler) ait olma olasılıklarıyla ise etiket yumuşatmalı karşıt-ilinti kaybı (L_{LS-CE}) (Denklem 3.4) hesaplanır. DY-BL de ise DY-ResNet-50 omurga ağı ile çıkarılan özniteliklere aynı işlemler uygulanır. DY-BL eğitim mimarisi Şekil 3.2'de görülmektedir.

ST-BL ve DY-BL ağlarının kaybı, Denklem 3.1'deki gibi etiket yumuşatmalı karşıt-ilinti kaybı ve zor üçlü-kayıp değerleri toplanarak hesaplanır ve toplam kayıp üzerinden ağ ağırlıkları güncellenerek omurga ağı ve Re-ID modülü birleşik şekilde eğitilir.

$$L_{BL} = L_{LS-CE} + L_{Htri} \tag{3.1}$$



Şekil 3.2 : DY-BL eğitim mimarisi.

Etiket yumuşatmalı karşıt-ilinti kaybı (cross entropy with label smooth) [23] (L_{LS-CE}), derin ağlarda sıkça karşımıza çıkan aşırı öğrenme (over-fitting) ve aşırı güven (over-confidence) problemlerini azaltmak amacıyla karşıt-ilinti kaybının bir düzenleme faktörü (regularization factor) ile düzenlenmesi ile oluşturulmuş bir kayıp fonksiyonudur. Aşırı öğrenme problemi; ağın, eğitim örneklerini öğrenmek yerine ezberlemesi sonucu ortaya çıkan ve çıkarım aşamasındaki performansın eğitim aşamasına göre çok düşük olmasına neden olan bir problemdir. Aşırı güven ise sınıflandırıcının, giriş görüntülerini çok yüksek olasılıklarla sınıflara ataması sonucu tahmin edilen olasılıkların, ortalama doğruluk değerine göre çok yüksek olması problemidir. Karşıt-ilinti kaybı Denklem 3.2'de verilmiştir.

$$L_{CE} = -\frac{1}{S}\sum_{i=1}^{S} y_i log(p_i) \tag{3.2}$$

S sınıf sayısı, y_i i. sınıfa ait gerçek referans değer (ground truth (GT)) (GT sınıf için 1 diğer sınıflar için 0), p_i ise sınıflandırıcı çıkışında Softmax fonksiyonu ($p_i = exp(z_i)/\sum_{n=1}^{S} exp(z_n)$) ile hesaplanan i. sınıfa ait olasılık değeridir. Denklem 3.2'deki logaritma fonksiyonu sayesinde, GT sınıf için hesaplanan olasılık değeri arttıkça L_{CE} hızla düşer. Fakat karşıt-ilinti kayıp fonksiyonunda GT dağılımının 0-1'ler şeklinde olması (one-hot vector), modeli her iterasyonda Softmax fonksiyonu çıkışında GT sınıflarına 1, diğer sınıflara 0 olasılık atamak için zorlayacaktır. Bu

da modelin Softmax girişindeki vektör elemanlarını 1 çıkışı için $+\infty$'a, 0 çıkışı için $-\infty$'a götürmeye çalışması demektir. Bu da bazı durumlarda aşırı öğrenme ve aşırı güven problemlerine neden olabilir. Bu yüzden karşıt-ilinti kaybında kullanılan etiket vektörleri (Denklem 3.2'deki $y_i$ler), bir etiket yumuşatma faktörü ile yumuşatılarak (Denklem 3.3) aşırı öğrenme ve aşırı güven problemleri hafifletilmiştir [23].

$$y_{smooth} = (1-\varepsilon)y_{one-hot} + \varepsilon/N \tag{3.3}$$

Etiket yumuşatmalı karşıt-ilinti kaybı Denklem 3.4'te verilmiştir.

$$L_{LS-CE} = -\frac{1}{S}\sum_{i=1}^{S} y_{smooth_i} log(p_i) \tag{3.4}$$

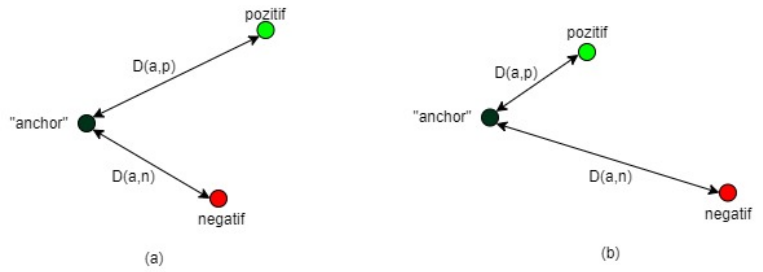
ε etiket yumuşatma faktörü (ST-BL, DY-BL, ST-Cace ve DY-Cace mimarileri için $\varepsilon = 0,1$) olmak üzere etiket vektörü artık 1 ve 0'lardan oluşmamaktadır. Bu da modelin, Softmax girişindeki vektör elemanlarını $+\infty$ ve $-\infty$'a götürmeye çalışmasını önleyerek aşırı öğrenmeyi azaltmakla beraber Softmax çıkışındaki olasılık değerlerinin birbirine daha yakın olmasını sağlayarak aşırı güveni de azaltır. Böylece modelin adaptifliği artar.

Üçlü-kayıp (Triplet loss), ilk olarak 2015 yılında [24]'te önerilen ve üçlü görüntü gruplarının öznitelikleri arasındaki uzaklıklarla hesaplanan bir kayıp fonksiyonudur. Bu kaybın kullanımındaki amaç ağı, aynı kimliğe ait öznitelikler daha yakın, farklı kimliğe ait öznitelikler daha uzak olacak şekilde öznitelikler çıkarmaya zorlamaktır. Bir üçlü grup "anchor", pozitif örnek ve negatif örnek olarak adlandırılan 3 adet görüntü içerir. Pozitif örnek, "anchor" görüntüsü ile aynı kimlikten başka bir görüntüdür ve eğitim sonunda pozitif örneğin özniteliklerinin "anchor"ın öznitelikleri ile yakın olması istenir. Negatif örnek ise "anchor"dan farklı kimlikten bir görüntüdür ve eğitim sonunda negatif örneğin öznitelikleri ile "anchor"ın özniteliklerinin birbirinden uzak olması istenir. Genel olarak bir üçlü eğitim örneği grubu için üçlü-kayıp Denklem 3.5'teki gibi hesaplanabilir.

$$L_{tri} = max(D(a,p) - D(a,n) + m, 0) \tag{3.5}$$

Burada $D(a,p)$ "anchor" ve pozitif örneğin öznitelikleri arasındaki uzaklık, $D(a,n)$ "anchor" ve negatif örneğin öznitelikleri arasındaki uzaklık, m ise marjin hiperparametresidir.

Şekil 3.3'te görüldüğü gibi eğitim boyunca kaybı küçültmek için $D(a,p)$ azalmaya, $D(a,n)$ artmaya zorlanır ve böylece eğitim sonunda aynı kimliğe sahip görüntülerin öznitelikleri birbirine yaklaşırken farklı kimlikten görüntülerin öznitelikleri birbirinden uzaklaşır.



Şekil 3.3 : (a) eğitim öncesi öznitelik vektörleri uzaklıkları (b) eğitim sonrası öznitelik vektörleri uzaklıkları ([3]'ten uyarlanmıştır).

Üçlü-kaybın önerilmesinden bu yana birçok farklı çalışmada üçlü grupların seçilmesi konusunda birçok farklı yaklaşım uygulanmıştır. Bu tezde [25]'te önerilen ve literatürde "zor üçlü-kayıp (hard triplet loss)" olarak bilinen yaklaşım kullanılmıştır. Bu yaklaşımda her bir "batch" için rastgele P sınıftan, her bir sınıftan rastgele R tane veri olacak şekilde eğitim verileri seçilir. "Batch"teki her bir görüntü sırayla "anchor" görüntü olarak seçilir bu görüntünün öznitelik vektörü ve "batch" içindeki diğer eğitim örneklerinin öznitelik vektörleri arasındaki uzaklıklar hesaplanır. "anchor" görüntü ile aynı kimlikten en yüksek uzaklığa sahip görüntü zor-pozitif (hard-positive), farklı kimlikten en düşük uzaklığa sahip görüntü zor-negatif (hard-negative) olarak seçilir ve üçlü grup bu 3 görüntü ile elde edilerek Denklem 3.6'deki gibi kayıp hesaplanır [25].

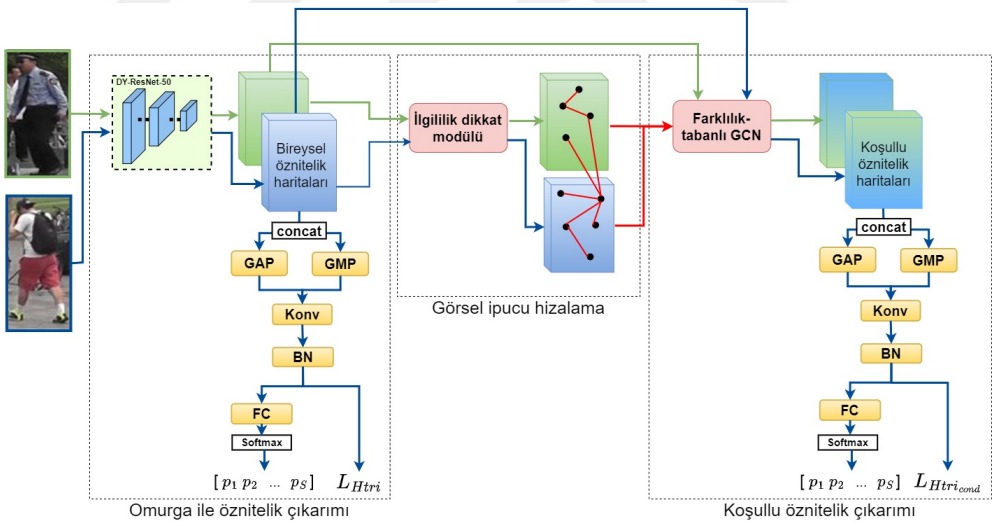
$$L_{Htri} = \sum_{i=1}^{P} \sum_{a=1}^{R} [m + \max_{p=1,\ldots,R} D(f_\theta(x_a^i), f_\theta(x_p^i)) - \min_{\substack{\text{j = 1,...,P} \\ \text{n = 1,...,R} \\ \text{j} \neq \text{i}}} D(f_\theta(x_a^i), f_\theta(x_n^j))] \tag{3.6}$$

Burada P "batch" içindeki sınıf (kimlik) sayısını, R her bir sınıftan "batch" içinde bulunan veri sayısını, m marjin hiperparametresini (ST-BL, DY-BL, ST-Cace ve DY-Cace mimarileri için $m = 0,5$), $f_\theta(x)$ x görüntüsünün öznitelik vektörünü, x_j^i ise

"batch"teki i. kimliğin j. görüntüsünü temsil etmektedir. Bu yaklaşım sayesinde olabildiğince kullanışlı üçlü gruplar seçilerek bu gruplardan olabildiğince fazla bilgi elde edilir ve ağ ağırlıklarının optimizasyonu daha verimli bir şekilde yapılır.

3.1.2 Graf tabanlı kişi tanılama ağı eğitim mimarisi

Re-ID ağlarında dinamik omurga mimarisi kullanımının etkilerini daha detaylı bir şekilde gözlemlemek amacıyla ST-BL ve DY-BL ağlarına ek olarak giriş verilerinin genel, yerel ve koşullu özniteliklerini kullanarak detaylı karşılaştırmalar sonucunda eşlemeler yapan CaceNet [5,10] (ST-Cace) ağının da omurga ağ mimarisindeki statik konvolüsyon katmanları, kanal tümleştirmeli dinamik konvolüsyon katmanlarıyla değiştirilmiştir ve DY-Cace ağı tasarlanmıştır. Omurgasında ST-ResNet-50 bulunduran ST-Cace ve omurgasında DY-ResNet-50 bulunduran DY-Cace mimarileri, 3 farklı kayıp fonksiyonu ile eğitilir ve omurga ile öznitelik çıkarımı, görsel ipucu hizalama (visual clue alignment) ve koşullu öznitelik çıkarımı olmak üzere 3 aşamadan oluşur. DY-Cace eğitim mimarisi Şekil 3.4'te görülmektedir.



Şekil 3.4 : DY-Cace eğitim mimarisi ([5]'ten uyarlanmıştır).

ST-Cace'de omurga ile öznitelik çıkarımı aşamasında öncelikle ağ girişindeki bir görüntü çiftinin ST-ResNet-50 omurga ağı (Bölüm 2.2.2) ile öznitelik haritaları çıkarılır. Sonrasında ST-BL ve DY-BL ağlarında olduğu gibi omurga ağı çıkışındaki her bir öznitelik haritasından 768 boyutlu öznitelik vektörleri elde edilir. Bu öznitelik vektörleri arasındaki uzaklıklar kullanılarak zor üçlü-kayıp (L_{Htri}) (Denklem 3.6),

bu vektörlerin FC katmanı ve Softmax fonksiyonundan geçirilmesi ile elde edilen sınıf olasılıklarıyla da etiket yumuşatmalı karşıt-ilinti kaybı (L_{LS-CE}) (Denklem 3.4) hesaplanır. DY-Cace'de ise DY-ResNet-50 omurga ağı (Bölüm 2.2.2) ile öznitelik haritaları çıkarıldıktan sonra bu haritalara aynı işlemler uygulanır. Şekil 3.2 ve Şekil 3.4'ten görülebileceği gibi DY-Cace'nin 1. aşaması, DY-BL mimarisine denktir. Aynı şekilde ST-Cace'nin 1. aşaması da DY-BL ile tamamen aynıdır.

ST-Cace ve DY-Cace mimarilerinde 2. ve 3. aşamalar aynıdır. ST-Cace ve DY-Cace mimarilerinin 2. aşaması olan görsel ipucu hizalama aşamasında, omurga çıkışındaki öznitelik haritaları kullanılarak ilgililik dikkat modülü ile hem iki öznitelik haritası arasındaki hem de her bir öznitelik haritasının kendi içindeki önemli görsel ipucu çiftleri seçilir. Böylece genel bilginin yanında detaylı bir karşılaştırma da yapabilmek için en ayırt edici bölgeler belirlenir. Bir giriş görüntüsü çifti için omurga çıkışındaki öznitelik haritaları $\mathbf{x}^{(u)}$ ve $\mathbf{x}^{(v)}$ olmak üzere omurga ile çıkarılan öznitelik haritalarının kendi içindeki ve iki öznitelik haritası arasındaki piksel çiftlerinin önemi sırasıyla Denklem 3.7 ve Denklem 3.8 ile hesaplanır.

$$\mathbf{S}^{(u)} = \hat{\mathbf{x}}^{(u)} \mathbf{W} \hat{\mathbf{x}}^{(u)T} \tag{3.7}$$

$$\mathbf{S}'^{(u,v)} = \hat{\mathbf{x}}^{(u)} \mathbf{W}' \hat{\mathbf{x}}^{(v)T} \tag{3.8}$$

Burada $\hat{\mathbf{x}}^{(u)}$ ve $\hat{\mathbf{x}}^{(v)}$, $HW \times C$ boyutlu matrislerdir ve omurga çıkışındaki $H \times W \times C$ boyutlu öznitelik haritaları $\mathbf{x}^{(u)}$ ve $\mathbf{x}^{(v)}$'nin tekrar boyutlandırılması ile elde edilir. $\mathbf{W}$ matrisi her bir öznitelik kanalına bir ağırlık atayan öğrenilebilir bir köşegen parametre matrisi, $\mathbf{S}^{(u)}$ ve $\mathbf{S}'^{(u,v)} \in \mathbb{R}^{HW \times HW}$ ise sırasıyla görüntü içi (intra-image) ve görüntüler arası (inter-image) olmak üzere her bir elemanı bir piksel çiftinin önem (importance) değerini barındıran önem matrisleridir. Son olarak önem matrisleri birleştirilir ve aktivasyon fonksiyonundan geçirilerek sadece pozitif önem değerine sahip piksel çiftleri tutulur (Denklem 3.9).

$$\mathbf{A}^{(u,v)} = ReLU \begin{bmatrix} \mathbf{S}^{(u)} & \mathbf{S}'^{(u,v)} \\ \mathbf{S}'^{(v,u)} & \mathbf{S}^{(v)} \end{bmatrix} \tag{3.9}$$

Seçilen önemli piksel çiftleri, komşuluk matrisi (adjacency matrix) $\mathbf{A}^{(u,v)}$ olan yönsüz bir graf formuna getirilir ve bu graf, koşullu öznitelik çıkarımında kullanılmak üzere 3. aşamasıya gönderilir.

ST-Cace ve DY-Cace'nin 3. aşaması olan koşullu öznitelik çıkarımı aşamasında ise farklılık-tabanlı bir graf konvolüsyon ağı (GCN) ile görüntü çiftlerinin birbirlerine göre koşullu öznitelik haritaları çıkarılır. Koşullu öznitelik çıkarımı amacıyla oluşturulan grafta düğüm öznitelikleri, omurga ağı çıkışındaki öznitelik haritalarının piksel değerleri, grafın komşuluk matrisi ise görsel ipucu hizalama aşamasında, ilgililik dikkat modülü ile tahmin edilen $\mathbf{A}^{(u,v)}$ matrisidir. Bu durumda hesaplanan graf konvolüsyonu Denklem 3.10'da verilmiştir.

$$g_\theta * x_i^{(u)} = \theta(\mathbf{I}_N - \mathbf{D}^{-1/2}\mathbf{A}^{(u,v)}\mathbf{D}^{-1/2})x_i^{(u)} \tag{3.10}$$

Burada θ öğrenilebilir bir parametre, $x_i^{(u)}$ düğüm özniteliği ($\mathbf{x}^{(\mathbf{u})}$ matrisinin i. piksel değeri), N grafın düğüm sayısı olmak üzere $\mathbf{I}_N$ N×N boyutlu birim matris, $\mathbf{A}^{(u,v)}$ komşuluk matrisi ve $\mathbf{D}$ köşegen elemanları düğümlerin bağlantı sayıları olan köşegen bir mertebe matrisidir (degree matrix). Bu konvolüsyon sonucunda ağın giriş görüntüleri $\mathbf{I}^{(u)}$ ve $\mathbf{I}^{(v)}$ olmak üzere $\mathbf{I}^{(u)}$'nun $\mathbf{I}^{(v)}$'ye göre ayarlanmış koşullu öznitelik haritası $f_{cond}(\mathbf{I}^{(u)}|\mathbf{I}^{(v)})$ ve $\mathbf{I}^{(v)}$'nin $\mathbf{I}^{(u)}$'ya göre ayarlanmış koşullu öznitelik haritası $f_{cond}(\mathbf{I}^{(v)}|\mathbf{I}^{(u)})$ elde edilir. Son olarak omurga ile öznitelik çıkarımı aşamasında omurgadan çıkan özniteliklere uygulanan adımlar izlenerek koşullu öznitelik haritalarından koşullu öznitelik vektörleri elde edilir ve bu koşullu öznitelik vektörleri ile zor üçlü-kayıp (Denklem 3.6), bu vektörlerin FC katmanı ve Softmax fonksiyonundan geçirilmesi ile elde edilen sınıf olasılıklarıyla da karıştırmalı karşıt-ilinti kaybı (mix-up cross entropy loss) (Denklem 3.12) hesaplanır.

ST-Cace ve DY-Cace ağlarının kaybı, 1. ve 3. aşamada elde edilen kayıp değerleri toplanarak hesaplanır ve birleşik bir eğitim yapılır. 3. aşamadaki kayıplara ek olarak 1. aşamada da kayıp hesaplanmasının nedeni, omurga ağını, temsil gücü daha yüksek öznitelikler çıkarmaya zorlamaktır. ST-Cace ve DY-Cace ağlarının kayıp fonksiyonu Denklem 3.11'de görülmektedir.

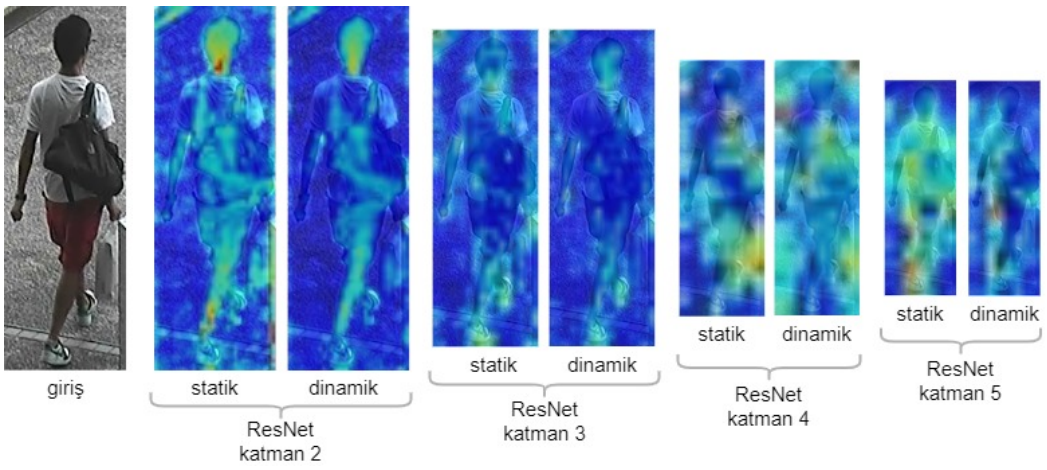
$$L_{Cace} = L_{LS-CE} + L_{Htri} + L_{mixup_0} + L_{mixup_1} + L_{Htri_{cond}} \tag{3.11}$$

ST-Cace ve DY-Cace ağlarının 3. aşamalarında elde edilen koşullu öznitelik vektörleri, her ne kadar çıkarıldığı görüntüye ait olsa da koşul olarak alınan görüntüye de bağlıdır. Bu nedenle öznitelik vektörünün çıkarıldığı görüntünün etiketinin yanı sıra koşul olarak alınan görüntünün etiketi de önemlidir. 3. aşamada, koşullu öznitelik haritaları çıkarılan görüntü çiftlerinde, kullanılan her iki görüntünün de etiketini kayıp hesaplamada göz önünde bulundurmak amacıyla, ağırlıklandırılmış karşıt-ilinti kayıplarının toplamı şeklinde tanımlanan karıştırmalı karşıt-ilinti kaybı (mix-up cross entropy loss) kullanılır. Eğitim verileri, her bir "batch" için rastgele P sınıftan, her bir sınıftan rastgele R tane veri olacak şekilde örneklenirse karıştırmalı karşıt-ilinti kaybı Denklem 3.12'deki gibi hesaplanır [5,10].

$$\begin{aligned} L_{mix-up} = \sum_{u=1}^{P}\sum_{v=1}^{R} & \alpha L_{CE}(y^{(u)}, f_{cond}(\mathbf{I}^{(u)}|\mathbf{I}^{(v)})) \\ & +(1-\alpha)L_{CE}(y^{(v)}, f_{cond}(\mathbf{I}^{(u)}|\mathbf{I}^{(v)}))) \end{aligned} \tag{3.12}$$

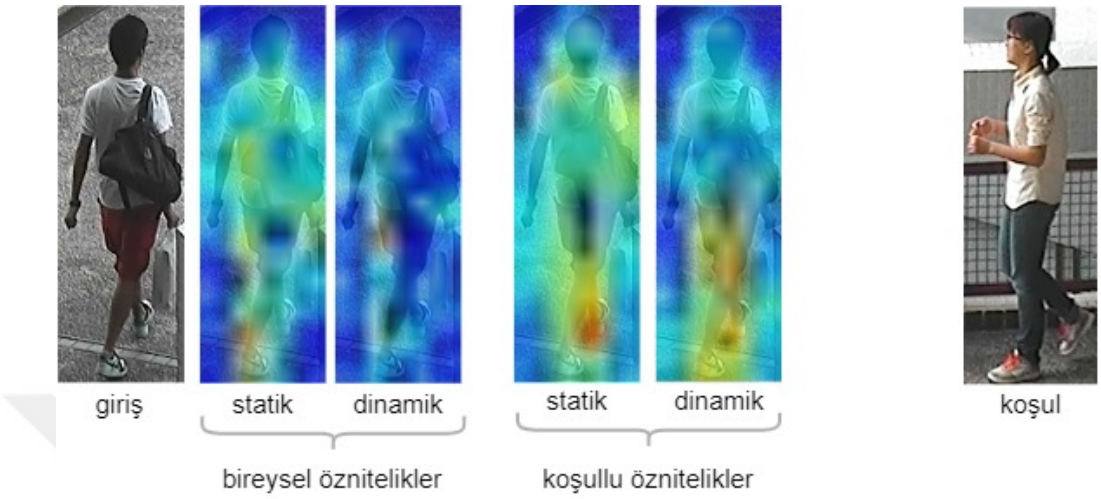
Burada L_{CE} karşıt-ilinti kaybı (Denklem 3.2), α ise karıştırma katsayısıdır (ST-Cace ve DY-Cace mimarileri için $\alpha = 0,9$).

CUHK03 veri setinden [14] seçilen bir görüntü için ST-Cace ve DY-Cace ağlarının omurga katman bloklarının çıkışlarında elde edilen kanal ortalamalı öznitelik haritaları Şekil 3.5'te verilmiştir.



Şekil 3.5 : CUHK03'ten seçilen bir görüntü için ST-Cace ve DY-Cace ağlarının omurga katman blokları çıkışlarında elde edilen öznitelik haritaları.

Aynı giriş görüntüsü için ST-Cace ve DY-Cace ağlarında elde edilen omurga ile çıkarılan ve koşullu öznitelik haritaları ise kanal ortalamalı şekilde Şekil 3.6'da görülmektedir.

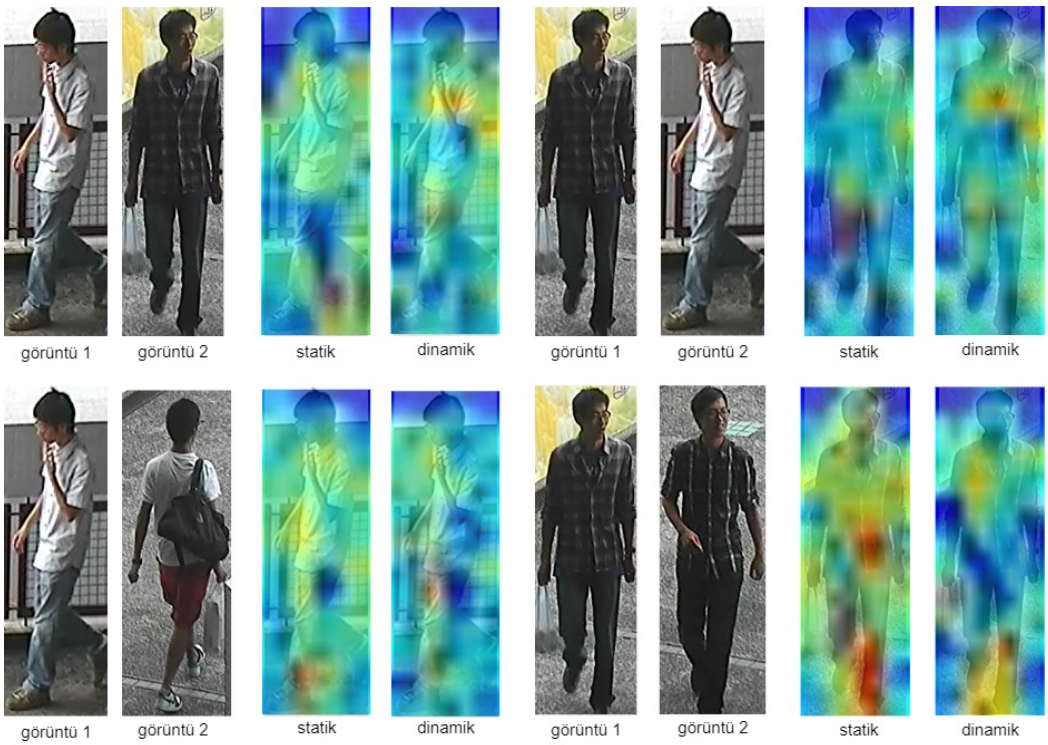


Şekil 3.6 : CUHK03'ten seçilen bir görüntü için ST-Cace ve DY-Cace ağlarında elde edilen omurga ile çıkarılan ve koşullu öznitelik haritaları.

Şekil 3.6'dan görüldüğü gibi koşullu öznitelikler, karşılaştırmada odaklanılacak bölgeleri belirler. Böylece bir görüntünün önemli noktaları, karşılaştırıldığı görüntü de göz önüne alınarak belirlenir.

CUHK03'ten seçilen farklı görüntüler için ST-Cace ve DY-Cace ağlarında elde edilen kanal ortalamalı koşullu öznitelik haritaları, 1. görüntü giriş görüntüsü, 2. görüntü koşul görüntüsü olmak üzere ise Şekil 3.7'de verilmiştir.

ST-Cace ve DY-Cace ağlarında görüntülerin odak noktaları, karşılaştırılan görüntüye göre adaptif şekilde değişir. Bu nedenle Şekil 3.7'den görüldüğü gibi bir görüntünün farklı görüntülere göre koşullu öznitelikleri birbirinden farklıdır. Ayrıca koşullu özniteliklerin çıkarımında, hem görüntü içi hem de görüntüler arası önemli piksel çiftlerinin kullanılmasından dolayı iki görüntünün birbirine göre koşullu öznitelikleri de farklıdır. Bu sayede ST-Cace ve DY-Cace ağları, giriş görüntülerinin hem genel hem de birbirlerine göre koşullu öznitelikleri kullanılarak oldukça ayrıntılı bir karşılaştırma yapılır.



Şekil 3.7 : CUHK03'ten seçilen görüntüler için ST-Cace ve DY-Cace ağlarında elde edilen koşullu öznitelik haritaları.

3.1.3 Kişi tanılama ağları çıkarım mimarisi

ST-BL ve DY-BL ağlarında çıkarım adımında, BN çıkışındaki 768 boyutlu öznitelik vektörleri kullanılarak sorgu görüntüleri ve galeri görüntüleri arasındaki kosinüs uzaklıkları hesaplanır ve her bir sorgu görüntüsü, öznitelik vektörünün en yakın olduğu galeri görüntüleri ile eşlenir. Herhangi iki vektörün yakınlığını hesaplamak için kullanılan kosinüs benzerliği S_{cos} ve ST-BL ve DY-BL ağlarında sorgu ve galeri görüntülerinin öznitelikleri arasındaki uzaklıkları hesaplamak için kullanılan kosinüs uzaklığı, sırasıyla Denklem 3.13 ve Denklem 3.14'te verilmiştir.

$$S_{cos} = cos(\theta) = \frac{\mathbf{A}.\mathbf{B}}{||\mathbf{A}||.||\mathbf{B}||} \tag{3.13}$$

$$D_{cos} = 1 - S_{cos} = 1 - \frac{\mathbf{A}.\mathbf{B}}{||\mathbf{A}||.||\mathbf{B}||} \tag{3.14}$$

ST-Cace'nin çıkarım adımında ise [5]'te belirtildiği üzere öncelikle bütün sorgu ve galeri görüntülerinin omurga ağı tarafından öznitelikler çıkarılır. Sonrasında her bir sorgu görüntüsü için ilgili galeri görüntüleri, özniteliklerin benzerliklerine göre sıralanır ve sorgu görüntüsü ve sorgu görüntüsüne en benzer M adet görüntünün öznitelik haritaları, 2. ve 3. aşamalara gönderilerek sorgu görüntülerinin, sadece en benzer M adet galeri görüntüsüne göre koşullu öznitelikleri çıkarılır. Son olarak sorgu ve galeri görüntülerinin koşullu özniteliklerinin benzerlikleri baz alınarak tekrar sıralama yapılır ve sorgu görüntüleri, en yakın galeri görüntülerine eşlenir. [5]'te M değeri 100 olarak belirlenmiştir. Fakat [5]'te sorgu görüntülerinin sadece 1. aşamada top-1 benzerliği en düşük olan %5'lik bir bölümü için koşullu öznitelik çıkarıp, yüksek top-1 benzerliğine sahip sorgu görüntülerini, sadece omurgayla çıkarılan öznitelikleri kullanarak galeri görüntüleri ile eşlemenin, işlem maliyeti bakımından daha verimli olduğu belirtilmiştir.

Bu tez çalışmasında ise ST-Cace ve DY-Cace çıkarım adımında bütün eşlemeler için sadece omurga ile çıkarılan öznitelik haritaları kullanılmıştır. Çıkarımda 1. aşamadaki BN çıkışındaki 768 boyutlu öznitelik vektörleri kullanılarak sorgu görüntüleri ve galeri görüntüleri arasındaki kosinüs uzaklığı (Denklem 3.14) hesaplanır ve sorgu görüntüleri, en yakın oldukları galeri görüntüleri ile eşlenir. Başka bir deyişle tez kapsamında ST-BL, DY-BL, ST-Cace ve DY-Cace ağlarının çıkarım adımları tamamen aynıdır.

Bölüm 4'te kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanlarıyla gerçeklenen omurga ağ mimarisinin, bu bölümde anlatılan Re-ID ağlarında performansa etkileri raporlanmıştır.

4. PERFORMANS TESTLERİ VE SONUÇLAR

Bu bölümde statik ve dinamik omurga ağ mimarileri ile gerçeklenen kişi tanılama ağlarında eğitim ve çıkarım aşamasında gözlemlenen performanslar incelenmiş ve kişi tanılama ağlarında dinamik omurga ağ mimarisi kullanımının performansa etkilerini gözlemleyebilmek amacıyla dinamik omurgalı kişi tanılama ağları, karşı düşen statik omurgalı ağlar ile karşılaştırılmıştır.

Eğitimler Bölüm 4.1'de anlatılan Market-1501 [12], DukeMTMC-reID [13], CUHK03 [14] ve Occluded-DukeMTMC [15] veri setlerinde 80 döngü boyunca yapılmış olup çıkarımlarda gözlemlenen performanslar da aynı veri setleri üzerinde raporlanmıştır. Eğitim ve çıkarımlar "Google Colaboratory" ortamında yapılmış olup yazılımda Python programlama dili kullanılmıştır. Baseline ve CaceNet kodları için [26] sitesindeki kodlardan yararlanılmıştır.

Baseline [26] ve CaceNet [5,10] ağlarının her ikisinde de omurga mimarisi olarak Bölüm 2.2.2'de anlatılan statik ve kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanlarıyla gerçeklenen ResNet-50 ağları kullanılmış (ST-ResNet-50 ve DY-ResNet-50) ve bütün Re-ID eğitimlerinde omurga ağı, ILSVRC-2012 veri seti [11] ile sınıflandırıcı olarak ön-eğitilmiş şekilde kullanılmıştır. ILSVRC-2012, 1000 farklı sınıftan nesneler içeren geniş kapsamlı bir veri setidir. Eğitim setinde 1.2 milyon, doğrulama setinde 50.000 ve test setinde 150.000 görüntü bulunur. Eğitim ve doğrulama setinin görüntüleri ve etiketleri açık kullanıma sunulurken test setinin sadece görüntüleri kullanılabilmektedir.

Sınıflandırıcı olarak eğitilmiş statik ResNet-50 modeli "torch" kütüphanesinden, dinamik ResNet-50 modeli ise [2]'nin açık kullanıma sunulan GitHub projesinden [27] alınarak kullanılmıştır. Statik ve dinamik konvolüsyon katmanları ile gerçeklenen ResNet-50 modellerinin ILSVRC-2012 doğrulama setinde sınıflandırma performansları Çizelge 4.1'de görülmektedir.

Çizelge 4.1 : ST-ResNet-50 ve DY-ResNet-50 ağlarının ILSVRC-2012 doğrulama setinde sınıflandırma performansları (top-k(%)).

Mimari	top-1	top-5
ST-ResNet-50	76,13	92,86
DY-ResNet-50	78,22	93,90

Çizelge 4.1'den görüldüğü gibi sınıflandırıcı olarak eğitilmiş dinamik model, sınıflandırıcı olarak eğitilen statik modelle karşılaştırıldığında top-1 metriğinde %2,09, top-5 metriğinde ise %1,04 artış sağlamıştır.

Bölüm 4.1'de ön-eğitilmiş statik ve dinamik omurga mimarileri kullanılarak Re-ID uygulaması için eğitilen ağların eğitim ve çıkarımlarında kullanılan veri setleri tanıtılmış, Bölüm 4.2'de kullanılan performans raporlama metrikleri anlatılmış, Bölüm 4.4 ve Bölüm 4.5'te ise Re-ID ağlarının sırasıyla eğitim ve çıkarım performansları incelenmiştir.

4.1 Kullanılan Veri Setleri

Re-ID eğitimi ve çıkarımında, veri sayıları ve zorlukları değişken 4 farklı Re-ID veri seti kullanılmıştır. Kullanılan veri setlerinde ışıklılık değişimi, benzer kıyafetli kişiler, kısmi örtüşme gibi eşlemeyi zorlaştırıcı etkiler bulunmaktadır.

Market-1501 veri seti [12]: Market-1501 veri seti kişi tanılama amacıyla oluşturularak 2015 yılında kullanıma sunulan, biri düşük çözünürlüklü (720×576), beşi yüksek çözünürlüklü (1280×1080) olmak üzere toplam 6 kameradan toplanmış görüntülerden oluşan, 1.501 farklı kimlikli kişiden toplam 32.668 görüntü içeren bir veri setidir. 751 kimlikten 12.936 görüntü eğitim seti, 750 kimlikten 3.368 görüntü sorgu görüntüsü, 15.913 görüntü galeri görüntüsü olmak üzere toplam 19.281 görüntü test seti olacak şekilde ayrılmıştır. Bu veri setinde farklı kameralar arasında görüş alanları kesişmesi bulunmakla birlikte her kişi en az 2 kamerada, en fazla 6 kamerada görülebilmektedir. Çıkarım aşamasında kullanılan 3.368 sorgu görüntüsü, test setinde her bir kameradan görülen her bir kimlik için bir görüntü sorgu görüntüsü olarak alınarak elde edilmiştir. Market-1501 veri setinden örnek görüntüler Şekil 4.1'de görülmektedir.

Şekil 4.1 : Market veri setinden görüntü örnekleri.

DukeMTMC-reID veri seti [13]: DukeMTMC-reID veri seti, DukeMTMC veri setinin kişi tanılama amacı ile oluşturulan bir alt kümesidir ve 8 farklı kameradan kaydedilmiş görüntüler içerir. 702 kimlikten 16.522 eğitim görüntüsü, 702 kimlikten de 2.228'i sorgu, 17.661'i galeri görüntüleri olmak üzere toplam 19.889 test görüntüsü içerir. DukeMTMC-reID veri setinde de Market-1501'deki gibi test setinde her bir kameradan görülen her bir kimlik için bir görüntü sorgu görüntüsü olarak alınmış, geri kalan görüntülerle ise galeri seti oluşturulmuştur. DukeMTMC-reID veri setinden örnek görüntüler Şekil 4.2'de görülmektedir.

Şekil 4.2 : DukeMTMC-reID veri setinden görüntü örnekleri.

CUHK03 veri seti [14]: CUHK03, 6 farklı kameradan kaydedilen, 1.467 kimliğe ait, bir kısmı insan eliyle etiketlenen bir kısmı dedektörle tespit edilmiş görüntüler bulunduran bir veri setidir. Eğitim setinde 767 kimliğe ait, 7.368 tanesi insan eliyle etiketlenmiş, 7.365 dedektör ile tespit edilmiş toplam 14.733 görüntü bulunur. Test setinde ise 700 kimlikten 1.400 elle etiketlenmiş, 1.400 dedektörle tespit edilmiş toplam 2.800 sorgu görüntüsü ve 5.328 elle etiketlenmiş, 5.332 dedektörle tespit edilmiş toplam 10.660 galeri görüntüsü bulunur. Dedektör ile tespit edilmiş görüntü setinde hedef kişilerin başka bir nesne tarafından kısmi kapatılması, yanlış hizalamalar, vücut parçası eksiklikleri bulunur. İnsan eliyle etiketlenen görüntü setinde ise hedef kişilerin başka bir nesne tarafından kısmi kapatılması durumu olmakla birlikte

sınırlandırıcı kutular (bounding box) düzgün hizalanmıştır. Her iki görüntü setinde de hava şartları, güneş ışığı yönü ve gölgeler nedeniyle ışıklılık değişimleri vardır. Bu da veri setinin zorluğunu arttırmaktadır. Tez kapsamında kişi tanılama eğitim ve çıkarımlarında sadece insan eliyle etiketlenmiş görüntüler kullanılmaktadır. Diğer veri setlerindeki gibi CUHK03 veri setinde de her bir kimlik için kişinin görüldüğü her kameradan bir görüntü sorgu görüntüsü olarak kullanılmış, geri kalan görüntüler ise galeri görüntüleri olarak kullanılmıştır. CUHK03 veri setinden görüntü örnekleri Şekil 4.3'te görülmektedir.

Şekil 4.3 : CUHK03 veri setinden görüntü örnekleri.

Occluded-DukeMTMC veri seti [15]: DukeMTMC-reID veri setindeki görüntülerin büyük bir kısmının farklı şekilde yeniden gruplara ayrılması ile elde edilen Occluded-DukeMTMC veri seti, eğitim setinde 702 kimlikten 15.618 görüntü içerir. Test setinde ise 519 kimlikten 2.210 adet sorgu görüntüsü, 1.110 kimlikten 17.661 adet galeri görüntüsü bulunur. Eğitim, sorgu ve galeri görüntülerinin sırasıyla %9, %100 ve %10'unun herhangi bir insan ya da nesne tarafından kısmen kapatılmış olması nedeniyle Occluded-DukeMTMC veri seti diğer veri setleriyle karşılaştırıldığında oldukça zordur. Occluded-DukeMTMC veri setinden örnek görüntüler Şekil 4.4'te görülmektedir.

Şekil 4.4 : Occluded-DukeMTMC veri setinden görüntü örnekleri.

4.2 Performans Raporlama Metrikleri

Eğitim aşamasında bir sınıflandırıcı problemi olarak ele alınan Re-ID problemi, çıkarım aşamasında sorgu görüntülerinin farklı kameralardan aynı kimliğe ait galeri görüntüleri ile eşlenmesi prensibine dayanır. Bu yüzden tez kapsamında performanslar, nesne sınıflandırma ve kişi tanılama problemleri üzerinden raporlanmıştır. Sınıflandırıcı performansı için SR (başarım oranı (success rate)) metriği, kişi tanılama performansı için ise mAP (ortalama hassasiyet (mean average precision)) ve literatürde "rank-k" olarak da bilinen top-k metrikleri kullanılmıştır.

Sınıflandırıcı performansı için kullanılan SR, basitçe Denklem 4.1'deki gibi hesaplanabilir.

$$\text{SR} = \frac{\text{\# doğru tahminler}}{\text{\# tüm tahminler}} \tag{4.1}$$

Tez kapsamında, eğitim aşamasında ST-BL ve DY-BL ağlarında mimari sonunda, ST-Cace ve DY-Cace ağlarında 1. aşama sonunda Softmax fonksiyonu ile hesaplanan sınıf olasılıkları ve GT kimlikler kullanılarak hesaplanan SR, basit şekilde en yüksek sınıf olasılığının GT sınıfa ait olduğu eğitim verisi sayısının tüm eğitim verilerinin sayısına oranı olarak tanımlanabilir.

Bölüm 3.1.2'te belirtildiği gibi Baseline ve CaceNet ağlarında çıkarım işlemi tamamen aynı şekilde yapılmaktadır. CaceNet mimarisinde çıkarım işleminin maliyetini azaltmak amacıyla çıkarım aşamasında sadece omurgayla çıkarılan öznitelik haritaları kullanılarak karşılaştırmalar yapılmıştır.

Çıkarım aşamasında öncelikle sorgu görüntüleri ve galeri görüntülerinin öznitelik vektörleri çıkarılır. Her bir sorgu görüntüsü için, sorgu görüntüsüne ait öznitelik vektörü ve bütün galeri görüntülerinin öznitelik vektörleri arasındaki kosinüs uzaklığı (Denklem 3.14) hesaplanarak bu değerler küçükten büyüğe sıralanır. Sonrasında her bir sorgu görüntüsü için, sorgu görüntüsü ile aynı kimlik ve kameradan olan galeri görüntüleri atılarak ilgili sorgu görüntüsüne ait galeri kümesi oluşturulur ve uzaklık

sıralamasında, ilgili sorgu görüntüsü ile aynı kimliğe sahip galeri görüntülerine ait sıra baz alınarak mAP ve top-k değerleri hesaplanır.

Her bir sorgu görüntüsü için hesaplanan ortalama kesinlik (AP) metriğinin denklemi, G_q ilgili sorgu görüntüsü ile aynı kimlik ve kameradan olan galeri görüntüleri atılarak oluşturulan sorgu görüntüsüne ait galeri kümesindeki galeri görüntüsü sayısı, G_{qTP} ilgili sorgu görüntüsünün galeri kümesinde bulunan aynı kimlikli görüntü sayısı ve M_i ilgili sorgu görüntüsü için $i.$ doğru eşlemeye kadar eşlenen toplam görüntü sayısı olmak üzere Denklem 4.2'de verilmiştir.

$$AP_q = \frac{\sum_{i=1}^{G_q}(i/M_i)}{G_{qTP}} \tag{4.2}$$

Tüm sorgu görüntüleri üzerinden hesaplanan, basit şekilde tüm sorgu görüntülerinin APlerinin ortalama değeri olarak tanımlanabilecek mAP metriğinin denklemi ise Q veri setindeki toplam sorgu görüntüsü sayısı ve AP_{q_j} $j.$ sorgu görüntüsüne ait AP değeri olmak üzere Denklem 4.3'te verilmiştir.

$$mAP = \frac{1}{Q}.\sum_{j=1}^{Q} AP_{q_j} \tag{4.3}$$

top-k ise Q_k $k.$ eşlemeye kadar herhangi bir doğru eşlemesi olan sorgu görüntüsü sayısı olmak üzere Denklem 4.4'te verilmiştir.

$$top-k = \frac{Q_k}{Q} \tag{4.4}$$

4.3 Ağ Eğitim Detayları

ST-BL, DY-BL, ST-Cace ve DY-Cace ağlarının eğitiminde Bölüm 4.1'de tanıtılan Market-1501, DukeMTMC-reID, CUHK03 ve Occluded-DukeMTMC veri setleri kullanılmıştır. Her bir ağ için eğitimler 80 döngü (epoch (ep)) boyunca baştan sona (end-to-end) yapılmış olup iyileştirici (optimizer) olarak Rasgele Gradyan İnişi (Stochastic Gradient Descent (SGD)) kullanılmıştır. SGDde momentum katsayısı 0.9, ağırlık azaltma (weight decay) katsayısı ise 5.10^{-4} olarak kullanılmıştır.

ST-BL ve DY-BL için "batch" boyutu 128, ST-Cace ve DY-Cace için 16 olarak kullanılmış ve her mimari için her bir "batch", "batch"te bulunan her bir kimlikten en az 4 örnek içerecek şekilde eğitim veri setlerinden örneklenmiştir. Öğrenme oranı (learning rate (LR)), ilk 5 döngü boyunca ağı ısıtmak (warm up) amacıyla 0'dan LRnin başlangıç değerine lineer şekilde arttırılmış, eğitimin devamında ise başlangıç değerinden kosinüs yumuşatma (cosine annealing (CA)) metoduyla [28] eğitim sonunda 0 olacak şekilde düşürülmüştür. ST-BL ve DY-BL için LR başlangıç değeri 5.10^{-2}, ST-Cace ve DY-Cace içinse $6.25.10^{-3}$'tür. CA metodu ile LR düşümü formülasyonu Denklem 4.5'te görülmektedir.

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + cos(\frac{T_{cur}}{T}\pi)) \tag{4.5}$$

Burada η_t güncel durumdaki LR değeri, η_{max} ve η_{min} LR değerinin alabileceği en büyük ve en küçük değerler (yani sırasıyla $6.25.10^{-3}$ ya da 5.10^{-2} ve 0), T_{cur} LRnin güncellendiği ana kadarki döngü sayısı, T ise eğitimin sürdürüleceği döngü sayısıdır. CA metodu, eğitim başında değil de bu çalışmada kullanılan ağlarda olduğu gibi ağ bir ısıtma sürecinden geçirildikten sonra kullanılmaya başlanmak istenirse bu durumda T_0 LR düşümü için CA kullanılmaya başlanacak döngü sayısı olmak üzere LR düşümü formülasyonu Denklem 4.6'daki gibi yazılabilir.

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + cos(\frac{T_{cur} - T_0}{T - T_0}\pi)) \tag{4.6}$$

Bölüm 3.1.1'de anlatıldığı gibi ST-BL ve DY-BL'de mimari sonunda elde edilen L_{LS-CE} (Denklem 3.4) ve L_{Htri} (Denklem 3.6) kayıplarının toplamı ağların kayıp fonksiyonudur (Denklem 3.1). ST-Cace ve DY-Cace'de ise Bölüm 3.1.2'te anlatıldığı gibi 1. aşaması sonunda öznitelik vektörleri kullanılarak elde edilen L_{LS-CE} (Denklem 3.4) ve L_{Htri} (Denklem 3.6) kayıpları ile 3. aşama sonunda koşullu öznitelik vektörleriyle elde edilen L_{mixup_0}, L_{mixup_1} (Denklem 3.12) ve $L_{Htri_{cond}}$ (Denklem 3.6) kayıplarının toplamı ağların kayıp fonksiyonudur (Denklem 3.11). ST-BL için öğrenilebilir parametre sayısı 26.655.296 iken DY-BL için 30.704.328'dir. ST-Cace ve DY-Cace'de ise öğrenilebilir parametre sayıları sırasıyla 30.398.176 ve 34.447.208'dir.

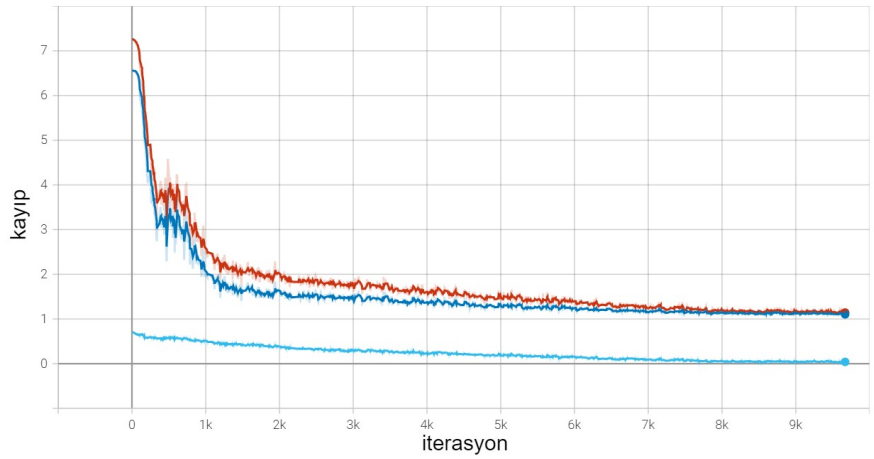
4.4 Ağların Öğrenme Performansı

Bu bölümde statik ve dinamik omurgalı Re-ID mimarilerinde eğitim aşamasında gözlemlenen performanslar incelenmiştir. Eğitimler Market-1501, DukeMTMC-reID, CUHK03 ve Occluded-DukeMTMC veri setlerinde 80 döngü boyunca yapılmış olup ST-BL ve DY-BL ağlarının öğrenme performansları DukeMTMC-reID veri setinde, ST-Cace, DY-Cace ağlarının öğrenme performansları ise CUHK03 veri setinde ayrıntılı raporlanmıştır.

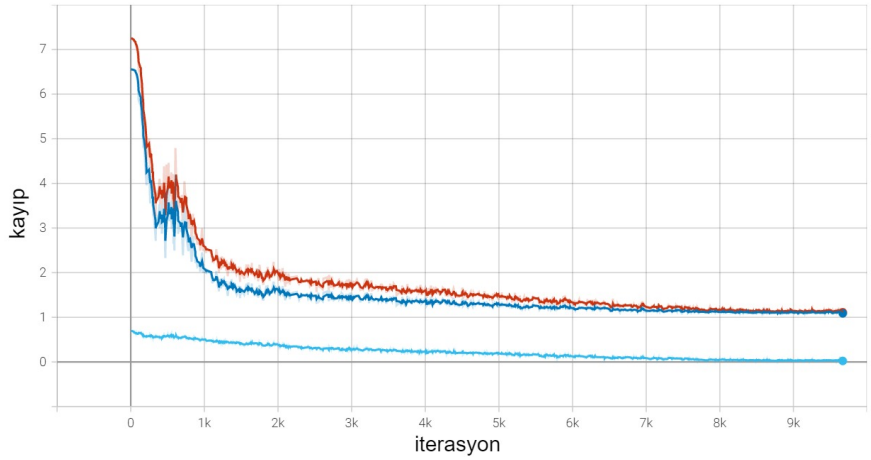
4.4.1 Nesne sınıflandırma performansı

Re-ID ağlarının eğitim kalitesini ölçmek için eğitim aşamasındaki sınıflandırma performansını incelemek önemlidir. Bu amaçla Bölüm 4.1'de anlatılan veri setleriyle ayrı ayrı eğitilen ST-BL, DY-BL, ST-Cace ve DY-Cace mimarilerinin nesne sınıflandırma performansları incelenmiş ve ST-BL ve DY-BL ağlarının DukeMTMC-reID veri seti ile, ST-Cace ve DY-Cace ağlarının CUHK03 veri seti ile eğitimlerindeki nesne sınıflandırma performansları ayrıntılı raporlanmıştır. ST-Cace, DY-Cace'de nesne sınıflandırma performans metriği SR, 1. aşamadaki sınıf olasılıkları baz alınarak hesaplanmıştır.

DukeMTMC-reID veri seti ile 80 döngü (9.669 iterasyon) ST-BL ve DY-BL eğitiminde gözlemlenen kayıp grafikleri yatay eksen iterasyon sayısı, düşey eksen kayıp değeri olmak üzere sırasıyla Şekil 4.5 ve Şekil 4.6'da verilmiştir.



Şekil 4.5 : ST-BL'nin DukeMTMC-reID ile eğitimi boyunca gözlemlenen kayıplar (kırmızı: L_{BL}, lacivert: L_{LS-CE}, mavi: L_{Htri}).



Şekil 4.6 : DY-BL'nin DukeMTMC-reID ile eğitimi boyunca gözlemlenen kayıplar (kırmızı: L_{BL}, lacivert: L_{LS-CE}, mavi: L_{Htri}).

Eğitim sonucunda L_{BL}, ST-BL'de 1,14 iken DY-BL'de 1,11'dir.

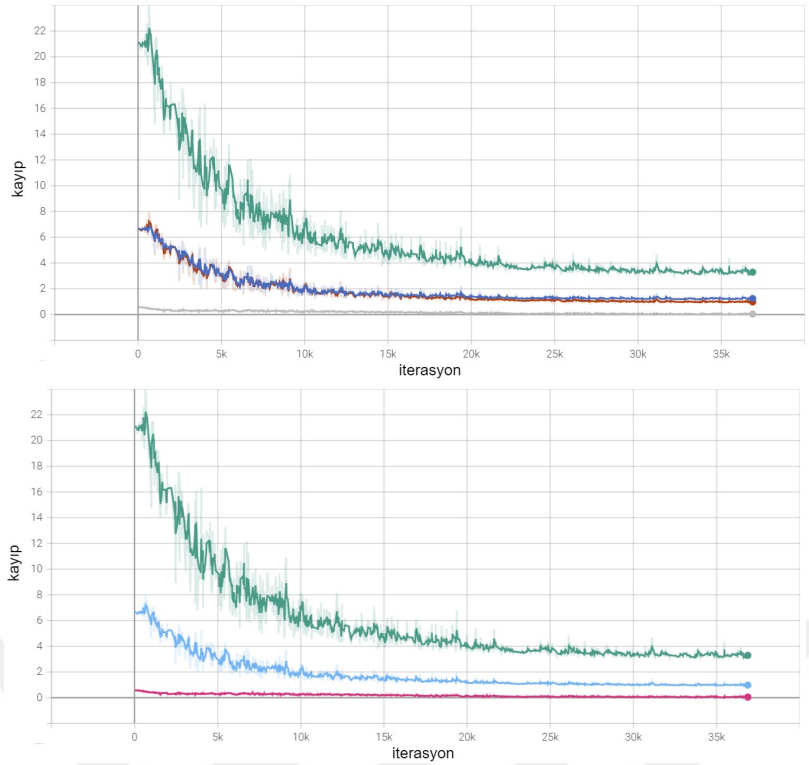
ST-BL ve DY-BL ağlarının DukeMTMC-reID veri seti ile eğitiminde 10 döngüde bir gözlemlenen SRler Çizelge 4.2'de görülmektedir.

Çizelge 4.2 : ST-BL ve DY-BL'nin DukeMTMC-reID veri seti ile eğitimi boyunca gözlemlenen SR(%).

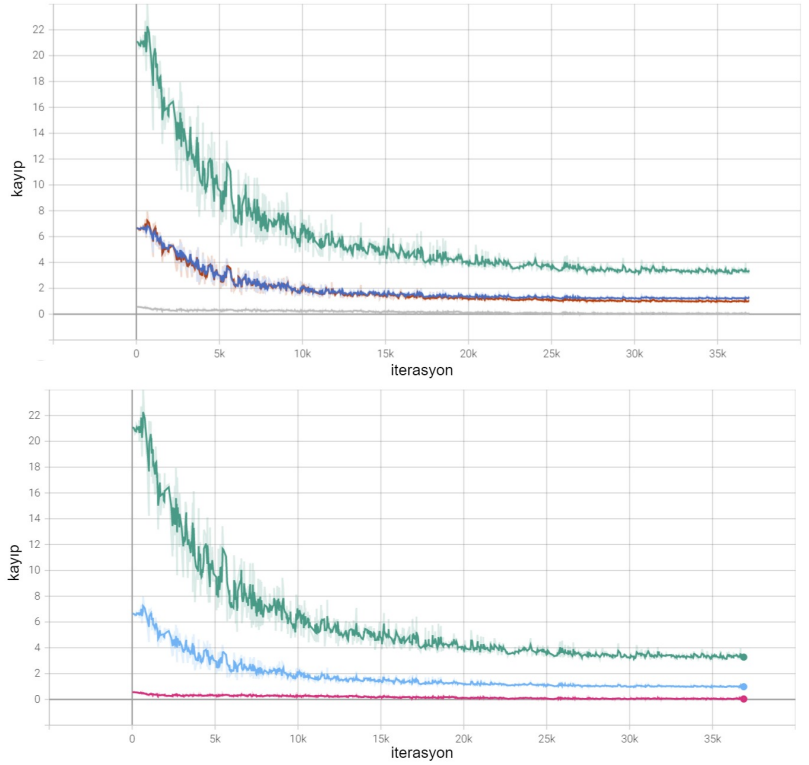
ep	ST-BL	DY-BL
80	99,72	99,78
70	99,65	99,74
60	99,36	99,53
50	98,85	98,92
40	97,20	97,82
30	95,93	96,11
20	92,99	93,71
10	85,45	85,49

Şekil 4.5 ve Şekil 4.6 incelendiğinde kayıplarda önemli bir fark görülmemekle birlikte Çizelge 4.2'den görülebileceği gibi DY-BL'de nesne sınıflandırma performansı, ST-BL'ye kıyasla eğitim boyunca bir miktar daha yüksektir.

CUHK03 veri seti ile 80 döngü (36.880 iterasyon) ST-Cace ve DY-Cace eğitiminde gözlemlenen kayıplarların grafikleri ise yatay eksen iterasyon sayısı, düşey eksen kayıp değeri olmak üzere sırasıyla Şekil 4.7 ve Şekil 4.8'de verilmiştir. Kayıp eğrilerinin örtüşmesini engellemek amacıyla gözlemlenen kayıp eğrileri iki grafikte verilmiştir.



Şekil 4.7 : ST-Cace'nin CUHK03 ile eğitimi boyunca gözlemlenen kayıplar (yeşil: L_{Cace}, lacivert: L_{LS-CE}, kırmızı: L_{mixup_0}, mavi: L_{mixup_1}, pembe: $L_{Htri_{cond}}$, gri: L_{Htri}).



Şekil 4.8 : DY-Cace CUHK03 ile eğitimi boyunca gözlemlenen kayıplar (yeşil: L_{Cace}, lacivert: L_{LS-CE}, kırmızı: L_{mixup_0}, mavi: L_{mixup_1}, pembe: $L_{Htri_{cond}}$, gri: L_{Htri}).

Eğitim sonucunda L_{Cace}, ST-Cace'de 3,31 iken DY-Cace'de 3,1'dir. Kayıp grafikleri incelendiğinde her iki ağın da yakınsadığı görülse de ST-Cace ve DY-Cace arasında kayıp değerleri bakımından önemli bir fark görülmemektedir.

ST-Cace ve DY-Cace ağlarının CUHK03 veri seti ile eğitiminde 10 döngüde bir hesaplanan SRler Çizelge 4.3'te görülmektedir.

Çizelge 4.3 : ST-Cace ve DY-Cace'nin CUHK03 veri seti ile eğitimi boyunca gözlemlenen SR(%).

ep	ST-Cace	DY-Cace
80	99,46	99,44
70	99,45	99,32
60	99,22	99,07
50	98,33	98,29
40	96,33	96,09
30	91,49	92,28
20	79,30	81,30
10	42,61	43,17

Kayıp grafiklerinde (Şekil 4.7 ve Şekil 4.8) belirgin bir fark gözlemlenmemiş olsa da Çizelge 4.3 incelendiğinde ilk döngülerde DY-Cace ağının sınıflandırma performansı, ST-Cace ile karşılaştırıldığında daha yüksektir. Eğitimin sonlarına doğru ise bu fark kapanır ve her iki ağ da benzer sınıflandırma performansına erişir.

4.4.2 Eğitim boyunca elde edilen özniteliklerin ayırt ediciliğinin incelenmesi

Kişi tanılama ağlarının eğitim kalitesini ölçmek için sınıflandırıcı performansına ek olarak eğitim boyunca elde edilen özniteliklerin Re-ID görevinde ayırt ediciliğinin incelenmesi de önemlidir. Bu amaçla eğitim boyunca eğitim verilerinin omurga ağı tarafından çıkarılan öznitelik haritaları, çıkarım rutininde sorgu ve galeri öznitelikleri olarak kullanılarak 10 döngüde bir Re-ID performansları hesaplanmıştır.

ST-BL ve DY-BL'nin DukeMTMC-reID veri seti ile eğitiminde gözlemlenen Re-ID performansları Çizelge 4.4'te verilmiştir.

Çizelge 4.2 ve Çizelge 4.4'ten görüldüğü gibi Baseline ağında dinamik omurga ağ mimarisi kullanımı, DukeMTMC-reID eğitiminde nesne sınıflandırma ve ReID performanslarını yükseltmiştir.

Çizelge 4.4 : ST-BL ve DY-BL'nin DukeMTMC-reID veri seti ile eğitimi boyunca gözlemlenen Re-ID performansları (mAP(%)/top-1(%)).

ep	ST-BL	DY-BL
80	99,94 / 99,87	99,98 / 99,98
70	99,91 / 99,83	99,97 / 99,94
60	99,73 / 99,65	99,86 / 99,79
50	98,34 / 98,55	98,96 / 99,12
40	94,43 / 96,08	94,24 / 96,25
30	87,24 / 92,26	89,15 / 93,94
20	76,79 / 87,88	81,65 / 89,04
10	65,83 / 79,98	66,52 / 79,88

ST-BL ve DY-BL'nin, kullanılan diğer veri setleri ile eğitimleri boyunca gözlemlenen Re-ID performansları Çizelge 4.5'te görülmektedir.

Çizelge 4.5 : ST-BL ve DY-BL'nin kullanılan diğer veri setleri ile eğitimleri boyunca gözlemlenen Re-ID performansları (mAP(%)/top-1(%)).

	Market-1501		CUHK03		Occluded-Duke	
ep	ST-BL	DY-BL	ST-BL	DY-BL	ST-BL	DY-BL
80	99,88 / 99,88	99,86 / 99,84	99,97 / 99,96	99,95 / 99,93	99,95 / 99,92	99,98 / 99,96
70	99,84 / 99,83	99,82 / 99,81	99,97 / 99,96	99,95 / 99,93	99,94 / 99,90	99,98 / 99,96
60	99,62 / 99,71	99,61 / 99,68	99,91 / 99,95	99,84 / 99,89	99,66 / 99,57	99,87 / 99,83
50	98,93 / 99,36	98,81 / 99,17	99,41 / 99,58	99,28 / 99,43	98,90 / 99,12	99,09 / 99,28
40	96,83 / 98,38	97,03 / 98,49	95,92 / 96,63	96,64 / 97,15	94,42 / 95,91	95,29 / 96,48
30	94,13 / 97,13	94,24 / 97,29	89,12 / 90,72	88,56 / 90,26	89,20 / 93,13	90,58 / 94,05
20	88,66 / 94,52	86,82 / 93,87	75,33 / 77,44	70,14 / 72,29	81,54 / 89,36	85,15 / 91,50
10	72,46 / 87,65	74,15 / 87,65	22,34 / 22,26	25,02 / 26,51	71,25 / 83,34	69,25 / 82,01

Çizelge 4.5 incelendiğinde diğer veri setleri ile eğitimlerde, Baseline ağında dinamik omurga kullanımının Occluded-DukeMTMC veri setinde eğitimde ReID performansını arttırdığı görülmekte, Market-1501 ve CUHK03'te ise önemli bir fark görülmemektedir.

ST-Cace ve DY-Cace ağlarının CUHK03 veri seti ile eğitiminde gözlemlenen Re-ID performansları ise Çizelge 4.6'da verilmiştir.

Çizelge 4.6 : ST-Cace ve DY-Cace'nin CUHK03 veri seti ile eğitimi boyunca gözlemlenen Re-ID performansları (mAP(%)/top-1(%)).

ep	ST-Cace	DY-Cace
80	99,98 / 99,97	99,98 / 99,96
70	99,98 / 99,97	99,98 / 99,96
60	99,97 / 99,97	99,97 / 99,96
50	99,91 / 99,95	99,84 / 99,82
40	99,05 / 99,25	99,06 / 99,25
30	93,45 / 94,13	95,30 / 96,13
20	84,78 / 86,09	86,53 / 88,28
10	56,37 / 57,79	62,05 / 63,03

Kullanılan diğer veri setleri ile ST-Cace ve DY-Cace eğitimleri boyunca gözlemlenen Re-ID performansları Çizelge 4.7'de görülmektedir.

Çizelge 4.7 : ST-Cace ve DY-Cace'nin kullanılan diğer veri setleri ile eğitimleri boyunca gözlemlenen Re-ID performansları (mAP(%)/top-1(%)).

	Market-1501		DukeMTMC-reID		Occluded-Duke	
ep	ST-Cace	DY-Cace	ST-Cace	DY-Cace	ST-Cace	DY-Cace
80	99,68 / 99,77	99,68 / 99,82	99,84 / 99,87	99,65 / 99,84	99,69 / 99,78	99,41 / 99,50
70	99,60 / 99,68	99,57 / 99,73	99,72 / 99,81	99,57 / 99,81	99,55 / 99,69	99,18 / 99,35
60	99,34 / 99,53	99,36 / 99,62	99,05 / 99,56	99,13 / 99,46	99,22 / 99,24	97,75 / 98,56
50	98,93 / 99,30	98,86 / 99,37	98,01 / 98,61	96,91 / 98,26	97,93 / 98,28	97,34 / 98,26
40	98,11 / 99,00	97,86 / 98,91	95,03 / 96,88	94,33 / 96,85	96,00 / 97,45	94,48 / 97,11
30	96,58 / 98,17	96,35 / 98,37	90,80 / 95,25	90,58 / 94,86	92,13 / 95,10	91,23 / 95,33
20	92,59 / 96,57	93,50 / 97,24	83,18 / 90,96	83,87 / 91,39	87,64 / 92,57	86,79 / 93,37
10	83,47 / 92,85	82,38 / 92,01	76,98 / 86,64	75,36 / 85,23	78,42 / 87,29	79,30 / 88,15

Çizelge 4.6'da görüldüğü gibi CUHK03 eğitim verisi ile elde edilen Re-ID performansları, Çizelge 4.3'te verilen nesne sınıflandırma performanslarında

olduğu gibi dinamik omurga ağ mimarisi sayesinde eğitimin ilk aşamalarında ST-Cace'yle karşılaştırıldığında DY-Cace'de daha yüksektir. Eğitim sonunda ise nesne sınıflandırma performansına benzer şekilde her iki ağda da benzer Re-ID performansına erişilir.

Çizelge 4.7 incelendiğinde ise CUHK03 eğitiminde görülen ST-Cace'yle karşılaştırıldığında DY-Cace'de Re-ID performansının eğitimin ilk aşamalarında daha yüksek olması durumu, diğer veri setleri ile eğitimlerde gözlemlenmemiştir.

4.5 Ağların Çıkarım Performansı

Bu bölümde statik ve dinamik omurgalı mimarilerin çıkarım aşamasındaki performansları incelenmiştir. Bölüm 4.4'e benzer şekilde bu bölümde de ST-BL ve DY-BL ağlarının çıkarım performansları DukeMTMC-reID veri setinde, ST-Cace ve DY-Cace ağlarının çıkarım performansları ise CUHK03 veri setinde detaylı raporlanmıştır.

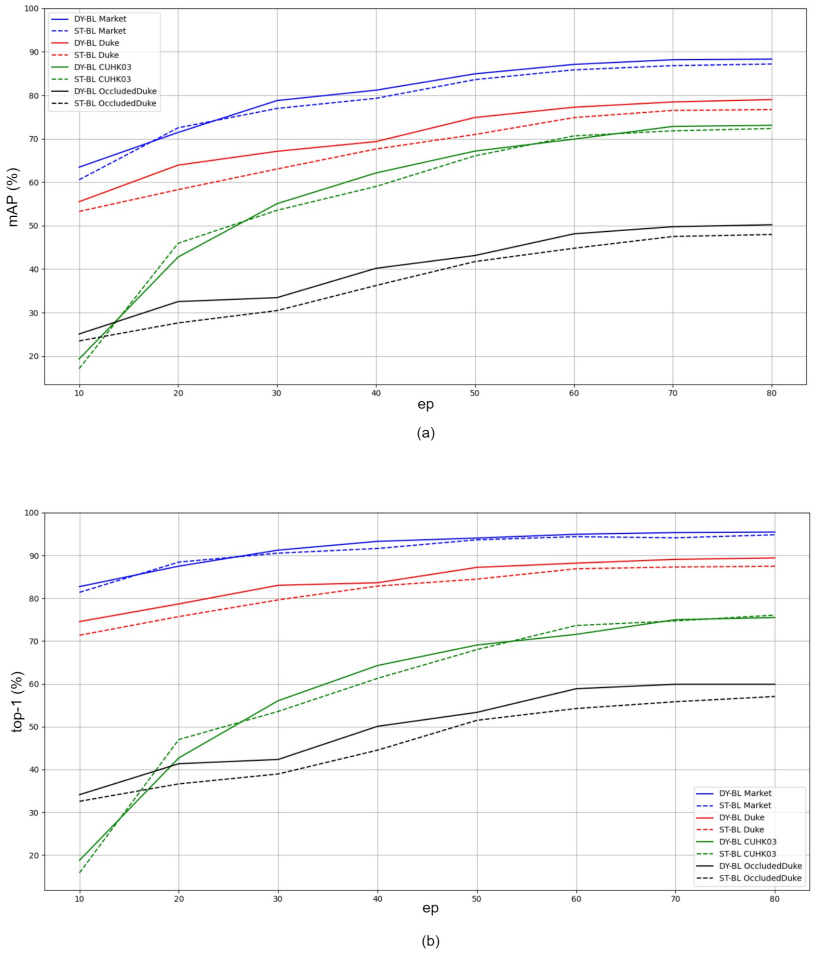
DukeMTMC-reID eğitim seti ile eğitilen ST-BL ve DY-BL ağlarının DukeMTMC-reID test setinde 10 döngüde bir hesaplanan çıkarım performansları Çizelge 4.8'de görülmektedir.

Çizelge 4.8 : DukeMTMC-reID veri seti ile eğitilen ST-BL ve DY-BL'nin DukeMTMC-reID veri setinde çıkarım aşamasında gözlemlenen Re-ID performansları (mAP(%)/top-1(%)).

ep	ST-BL	DY-BL
80	76,70 / 87,48	79,01 / 89,41
70	76,51 / 87,30	78,46 / 89,09
60	74,86 / 86,89	77,25 / 88,20
50	70,97 / 84,47	74,88 / 87,21
40	67,63 / 82,85	69,34 / 83,62
30	63,03 / 79,62	67,10 / 83,03
20	58,28 / 75,72	63,92 / 78,68
10	53,26 / 71,36	55,51 / 74,55

Çizelge 4.8'den görüldüğü gibi DukeMTMC-reID veri seti ile çıkarım aşamasında DY-BL'de gözlemlenen Re-ID performansları, ST-BL'ye kıyasla hem mAP hem de top-1 metriği için daha yüksektir. Buradan dinamik omurga ağ mimarisi ile çıkarılan ayırt edici özniteliklerin sağladığı performans artışı gözlemlenebilmektedir.

Kullanılan tüm veri setleri ile ST-BL ve DY-BL'nin çıkarım adımlarında gözlemlenen Re-ID performansları Şekil 4.9'da verilmiştir.



Şekil 4.9 : Kullanılan her bir veri seti için ST-BL ve DY-BL çıkarım aşamasında gözlemlenen Re-ID performansları (a) mAP(%) (b) top-1(%).

Şekil 4.9'dan görüldüğü gibi Baseline ağında dinamik omurga kullanımı, Re-ID performansını çoğu durumda yükseltmiştir. ST-BL ile karşılaştırıldığında DY-BL'de Market-1501, DukeMTMC-reID, CUHK03 ve Occluded-DukeMTMC veri setleri için sırasıyla %1,12, %2,31, %0,73 ve %2,24 daha yüksek mAP değerleri raporlanmıştır. top-1 bazında ise DY-BL'de Market-1501, DukeMTMC-reID ve Occluded-DukeMTMC ile sırasıyla %0,63, %1,93 ve %3,08 artış görülürken CUHK03'te %0,57'lik bir düşüş görülmüştür.

CUHK03 eğitim seti ile eğitilen ST-Cace ve DY-Cace ağlarının CUHK03 test setinde çıkarım performansları ise Çizelge 4.9'da verilmiştir.

Çizelge 4.9 : CUHK03 veri seti ile eğitilen ST-Cace ve DY-Cace'nin CUHK03 veri setinde çıkarım aşamasında gözlemlenen Re-ID performansları (mAP(%)/top-1(%)).

ep	ST-Cace	DY-Cace
80	80,03 / 82,43	80,05 / 82,07
70	79,47 / 80,79	79,61 / 80,86
60	78,77 / 80,50	78,57 / 80,50
50	76,17 / 77,93	76,82 / 79,64
40	70,92 / 73,07	71,45 / 73,79
30	61,35 / 63,43	64,24 / 65,64
20	55,91 / 57,00	58,10 / 60,07
10	39,04 / 38,93	43,75 / 44,43

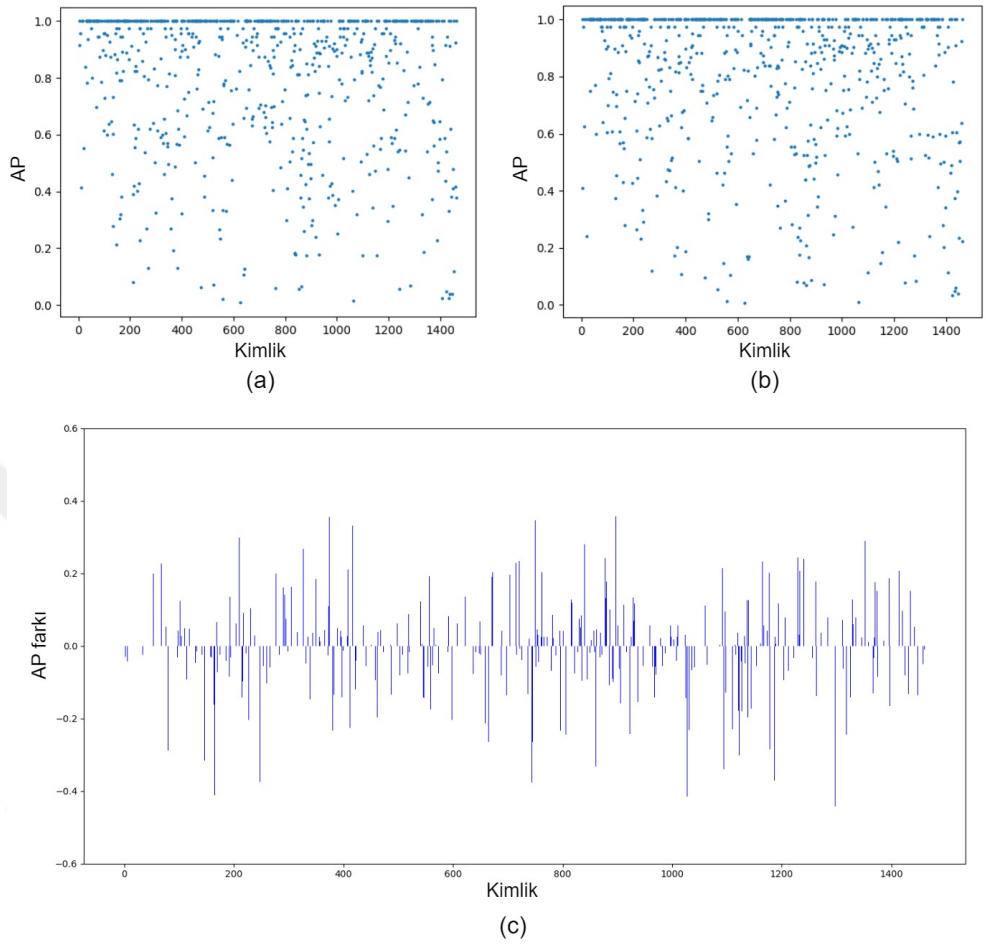
Çizelge 4.9'dan görüldüğü gibi ST-Cace ve DY-Cace ağlarının CUHK03 veri seti ile eğitiminde öğrenme performansına benzer şekilde (Çizelge 4.3 ve Çizelge 4.6) CUHK03 veri seti ile çıkarım aşamasında da eğitimin ilk aşamalarında DY-Cace'de gözlemlenen Re-ID çıkarım performansları, ST-Cace'ye kıyasla daha yüksektir. Döngü sayısı arttıkça iki ağda da gözlemlenen Re-ID çıkarım performansları benzer seviyelere ulaşmıştır.

Fakat 80 döngü sonunda ST-Cace ve DY-Cace ağlarında benzer mAP ve top-1 değerleri gözlemlenmesi, dinamik omurga ağ mimarisinin kişi tanılamada olumlu ya da olumsuz etkileri olmadığını söyleyebilmek için yeterli değildir. Daha detaylı gözlem yapmak amacıyla kimlik bazında ortalama APler (Denklem 4.2) ve sorgu görüntülerinin öznitelik vektörleri ile aynı kimliğe sahip galeri görüntülerinin öznitelik vektörleri arasındaki kosinüs uzaklıkları da (Denklem 3.14) incelenmiştir.

ST-Cace vs DY-Cace'ye ait kimlik bazında ortalama AP grafikleri yatay eksen kimlik numarası, düşey eksen ilgili kimliğe ait ortalama AP olmak üzere sırasıyla Şekil 4.10(a) ve Şekil 4.10(b)'de görülmektedir. 4.10(c)'de ise her bir kimlik için DY-Cace çıkarımında hesaplanan APler ve ST-Cace çıkarımında hesaplanan APlerin farkları görülmektedir.

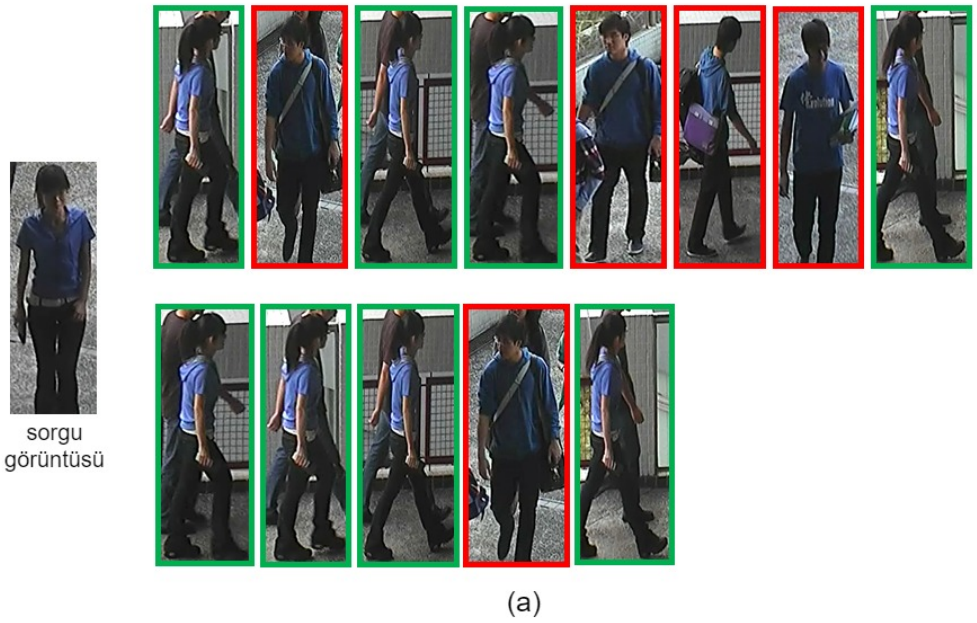
Şekil 4.10 incelendiğinde bazı kimlikler için hesaplanan APlerin ST-Cace ve DY-Cace ağlarında oldukça farklı olduğu görülmüştür. DY-Cace çıkarımında ST-Cace çıkarımına kıyasla daha yüksek ve daha düşük APlere sahip iki örnek sorgu görüntüsü ve eşlendiği galeri görüntüleri Şekil 4.11'de verilmiştir. Şekil 4.11(a)'daki 28. kimliğe

ait sorgu görüntüsü için ST-Cace'de AP yaklaşık %73 iken DY-Cace'de %95'tir. Şekil 4.11(b)'deki 349. kimliğe ait sorgu görüntüsünde ise ST-Cace'de AP yaklaşık %92 iken DY-Cace'de yaklaşık %80'dir.

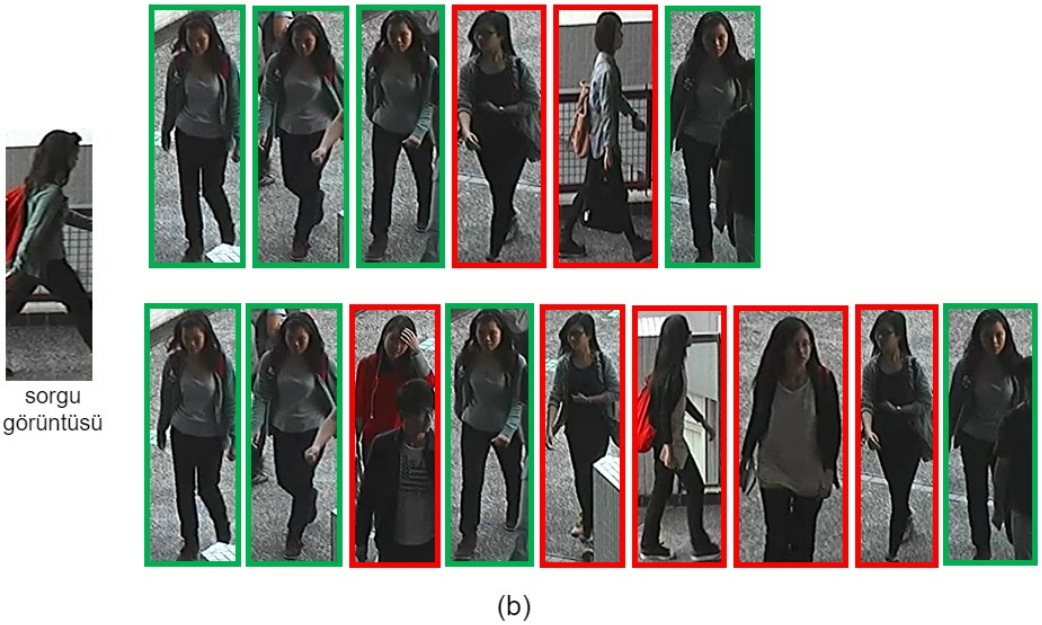


Şekil 4.10 : CUHK03 ile eğitilen ST-Cace ve DY-Cace'de çıkarım aşamasında kimlik bazında APler ve AP farkları (a) ST-Cace AP (b) DY-Cace AP (c) AP farkları (DY-Cace - ST-Cace).

ST-Cace ve DY-Cace'de yapılan eşlemelerde sorgu görüntülerinin özniteliklerinin, aynı kimliğe sahip galeri görüntülerinin özniteliklerine uzaklıkları ise yatay eksen sorgu görüntüsü indisi, düşey eksen ilgili sorgu görüntüsü ve bu görüntü ile aynı kimliğe sahip galeri görüntüleri arasındaki ortalama uzaklık olmak üzere sırasıyla Şekil 4.12(a) ve Şekil 4.12(b)'de verilmiştir. Şekil 4.12(a) ve Şekil 4.12(b)'de görüldüğü gibi ST-Cace ile karşılaştırıldığında DY-Cace, aynı kimlikli sorgu ve galeri görüntüleri için daha yakın öznitelik vektörleri çıkarmıştır. Farkın daha iyi gözlemlenebilmesi için uzaklık değerlerinin farkları Şekil 4.12(c)'de verilmiştir.



(a)

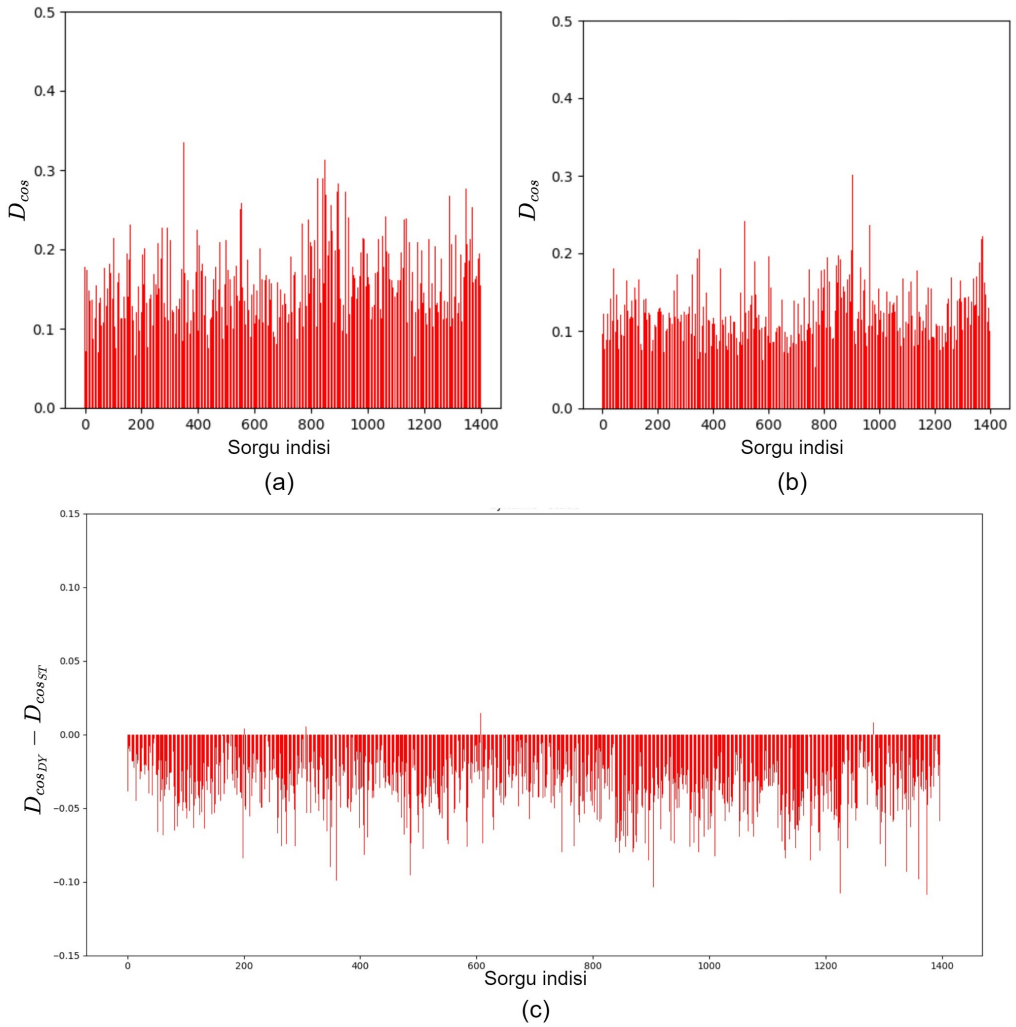


(b)

Şekil 4.11 : CUHK03'ten seçilen iki sorgu görüntüsü için DY-Cace ve ST-Cace çıkarımında yapılan eşlemeler. (a) 28. kimlik (b) 349. kimlik. Her bir sorgu görüntüsü için 1. satır ST-Cace, 2. satır DY-Cace ağında yapılan sıralamayı belirtmektedir. Yeşil kutular sorgu görüntüsü ile aynı, kırmızı kutular farklı kimliğe sahip görüntüleri ifade etmektedir.

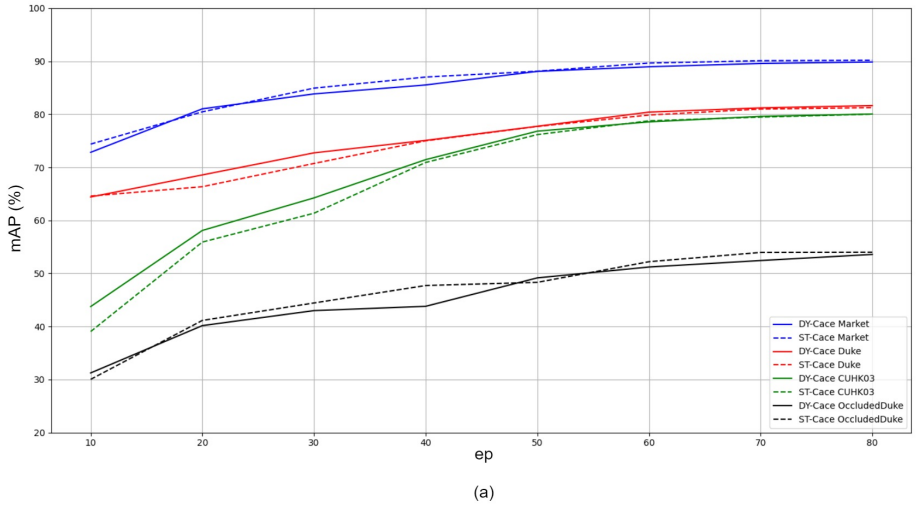
Şekil 4.12'den görüldüğü gibi ağlarda gözlemlenen mAP ve top-1 değerleri benzer olsa da ST-Cace ile karşılaştırıldığında DY-Cace, aynı kimlikli sorgu ve galeri görüntülerini, sorgu görüntülerinin %99,43'ünde daha düşük uzaklık değerleriyle eşlemiştir.

Occluded-DukeMTMC ve Market-1501'de de benzer bir durum gözlemlenmiş olup bu oran sırasıyla %96,92 ve %80,58'dir. DukeMTMC-reID'de ise sorgugörüntülerinin %57,90'ı DY-Cace'de daha ortalama uzaklıklarla eşlenmiştir. Bu da CaceNet'te dinamik omurga ağ mimarisi kullanımının, CUHK03, Occluded-DukeMTMC ve Market-1501 veri setleri ile çıkarımda eşleme güvenini arttırdığını gösterir.

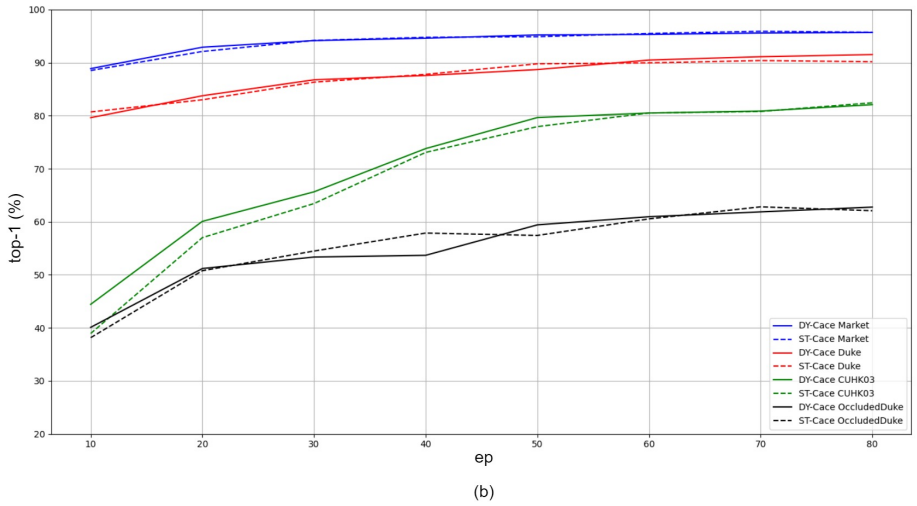


Şekil 4.12 : DY-Cace ve ST-Cace'nin çıkarım aşamasında CUHK03'teki sorgu görüntüleri ile aynı kimliğe sahip galeri görüntülerinin ortalama uzaklığı ve sorgu görüntüsü bazında uzaklık farkları (a) ST-Cace (b) DY-Cace (c) uzaklık farkları (DY-Cace - ST-Cace).

Kullanılan tüm veri setleri ile ST-Cace ve DY-Cace'nin çıkarımlarında gözlemlenen Re-ID performansları Şekil 4.13'te verilmiştir. ST-Cace ve DY-Cace'de farklı veri setleriyle çıkarımlarda ReID performanslarında, kullanılan metrikler bazında önemli bir fark görülmemiştir.



(a)



(b)

Şekil 4.13 : Kullanılan her bir veri seti için DY-Cace ve ST-Cace çıkarım aşamasında gözlemlenen Re-ID performansları (a) mAP(%) (b) top-1(%).

4.6 Güncel Kişi Tanılama Ağları ile Karşılaştırma

Statik ve dinamik omurga ağ mimarilerine sahip Baseline ve CaceNet ağları ile DukeMTMC-reID, CUHK03 ve Market-1501 veri setlerinde gözlemlenen çıkarım performanslarının, literatürde bulunan bazı güncel kişi tanılama ağları ile karşılaştırmaları Çizelge 4.10'da görülmektedir.

İlgili veri seti için sonucu bulunmayan ağlar Çizelge 4.10'da "-" ile belirtilmiştir. Görüldüğü gibi DukeMTMC-reID, CUHK03 ve Market-1501 veri setlerinde performans sıralamasında DY-BL, ST-BL ile karşılaştırıldığında çok daha iyi bir

yerdedir. ST-Cace ve DY-Cace ise literatürde bulunan kişi tanılama yöntemleri ile karşılaştırıldığında çoğu yöntemden daha yüksek kişi tanılama performanslarına erişmiştir.

Çizelge 4.10 : ST-Cace, DY-Cace, ST-BL ve DY-BL ağlarının DukeMTMC-reID, CUHK03 ve Market-1501 veri setlerinde literatürdeki kişi tanılama ağları ile karşılaştırılması.

Yöntem	DukeMTMC-reID			CUHK03		Market-1501		
	mAP	top-1	top-5	mAP	top-1	mAP	top-1	top-5
PL-Net [29]	-	-	-	-	82,7	69,3	88,2	-
MGCAM [30]	-	-	-	50,2	50,1	74,3	83,9	-
MLFN [31]	62,8	81,0	-	49,2	54,7	74,3	90,0	-
HA-CNN [32]	63,8	80,5	-	41,0	44,4	75,5	91,2	-
PCB-RPP [33]	69,2	83,3	-	-	-	81,6	93,8	97,5
CBN+BoT [34]	70,1	84,8	92,5	-	-	83,6	94,3	97,9
Mancs [35]	71,8	84,9	-	63,9	69,0	82,3	93,1	97,6
VMP [36]	72,6	83,6	91,7	-	-	80,8	93,0	97,8
SNR [37]	72,4	84,4	-	-	-	84,7	94,4	-
CAMA [38]	72,9	85,8	-	66,5	70,1	84,5	94,7	98,1
CASN (PCB) [39]	73,7	87,7	-	68,0	73,7	82,8	94,4	-
IANet [40]	73,4	87,1	-	-	-	83,1	94,4	-
SPReID [41]	73,3	85,9	92,9	-	-	83,4	93,7	97,6
DSA [42]	74,3	86,2		75,2	78,9	87,6	95,7	-
DG-Net [43]	74,8	86,6	-	-	-	86,0	94,8	-
ST-BL	76,7	87,5	94,21	72,4	76,1	87,2	94,8	98,2
Baseline [44]	77,2	88,3	-	-	-	87,6	94,8	-
MHN [45]	77,2	89,1	94,6	72,4	77,2	85,0	95,1	98,1
BAT-net [46]	77,3	87,7	94,7	76,1	78,6	87,4	94,1	98,2
MGN [21]	78,4	88,7	-	67,4	68,0	86,9	95,6	-
ABD-Net [47]	78,6	89,0	-	-	-	88,3	95,6	-
PISNet [48]	78,7	88,8	-	-	-	87,1	95,6	-
Pyramid-101 [22]	79,0	89,0	-	76,9	78,9	88,2	95,7	98,4
SCSN [49]	79,0	90,1	-	83,3	86,3	88,5	95,7	-
DY-BL	79,0	89,4	94,8	73,1	75,5	88,3	95,5	98,2
RGA-SC [50]	-	-	-	77,4	81,1	88,4	96,1	-
ST-Cace	81,3	90,2	**95,7**	80,0	82,4	90,2	95,7	98,6
CaceNet [5,10]	81,3	90,9	-	-	-	90,3	96	-
OPReID [51]	81,3	91,1	-	-	-	89,3	96,1	-
APNet-C [52]	81,5	90,4	95,6	**85,3**	**87,4**	**90,5**	**96,2**	**98,8**
DY-Cace	81,6	**91,5**	95,6	80,1	82,1	89,9	95,7	98,5
TransReID [53]	82,0	90,7	-	-	-	88,9	95,2	-
PFD [54]	**83,2**	91,2	-	-	-	89,7	95,5	-

Tez kapsamında kullanılan ağlarda, Occluded-DukeMTMC veri setinde gözlemlenen performansların literatürdeki ağlarla karşılaştırmaları ise Çizelge 4.11'de görülmektedir.

Çizelge 4.11 : ST-Cace, DY-Cace, ST-BL ve DY-BL ağlarının, Occluded-DukeMTMC veri setinde literatürdeki kişi tanılama ağları ile karşılaştırılması.

Yöntem	Occluded-DukeMTMC	
	mAP	top-1
PCB [20]	33,8	42,6
PGFA [15]	37,3	51,4
HOReID [55]	43,8	55,1
OAMN [58]	46,1	62,6
Visibility-aware [59]	46,3	62,2
SGSFA [56]	47,4	62,3
ST-BL	48,0	57,1
DY-BL	50,2	60,1
CaceNet [5,10]	50,8	58,8
ISP [57]	52,3	62,8
DY-Cace	53,6	**62,8**
ST-Cace	**54,0**	62,1

Eğitim ve çıkarımlarda kullanılan Occluded-DukeMTMC veri seti, [60] sitesindeki Python kodları kullanılarak DukeMTMC-reID veri setinden elde edilmiştir. Eğitim ve çıkarım hiperparametreleri [5]'te belirtildiği gibi her veri seti için aynı şekilde uygulanmıştır ve [5]'in aksine çıkarımda her bir giriş verisi için sadece omurga ile çıkarılan öznitelik haritaları kullanılarak eşlemeler yapılmıştır. Occluded-DukeMTMC veri setinde statik omurgalı Cacenet mimarisinin 80 döngü eğitimi sonucu gözlemlenen kişi tanılama performansı, [5]'te Occluded-DukeMTMC için raporlanan performanstan çok daha yüksek gözlemlenmiştir.

Çizelge 4.11'de görüldüğü gibi CaceNet mimarileri, literatürde Occluded-DukeMTMC veri seti için raporlama yapan yöntemlerle karşılaştırıldığında iyi sonuçlar elde etmiştir. Ayrıca Baseline ve CaceNet ağlarının çıkarım adımları tamamen aynı olsa da hem Çizelge 4.10 hem de Çizelge 4.11'den görülebileceği gibi CaceNet ağları, Baseline ağlarına göre çok daha yüksek performanslara ulaşmıştır. Bu durum birleşik öğrenmenin omurga performansına etkisini göstermektedir.

Sonuç olarak, Baseline gibi ayırt ediciliği düşük özniteliklerle eşlemeler yapan, performansı direkt olarak omurga ağının performansına bağlı yöntemlerde dinamik omurga ağ mimarisi kullanımının, hem öğrenme hem de çıkarım performansını önemli ölçüde arttırabileceği görülmüştür. CaceNet gibi ayırt ediciliği yüksek özniteliklerle detaylı eşlemeler yapan yöntemlerde ise dinamik omurga ağ mimarisi kullanımının katkısı sınırlı kalmıştır.

5. SONUÇLAR VE TARTIŞMA

Tez kapsamında, kişi tanılama uygulamasında daha ayırt edici öznitelikler çıkararak performansı arttırmak amacıyla, iki farklı kişi tanılama ağında dinamik konvolüsyonlu omurga ağ mimarisi kullanımı önerilmiş ve dinamik omurga ağ mimarisinin performansa etkileri incelenmiştir.

Dinamik omurga mimarisinin gerçeklenmesinde, literatürde bulunan kanal tümleştirme mekanizmalı dinamik konvolüsyon [2] kullanılmıştır. Bu yöntemde öncelikle katman girişi, statik bir konvolüsyon kerneli kullanılarak daha düşük boyutlu bir uzaya izdüşürülür ve katman girişinden elde edilen bilgiler kullanılarak üretilen bir kanal tümleştirme matrisi ile düşük boyutlu uzayda kanallar dinamik şekilde tümleştirilir. Sonrasında tümleştirilmiş kanallara sahip öznitelik haritası başka bir statik konvolüsyon kerneli kullanılarak tekrar yüksek boyutlu uzaya izdüşürülür. Böylece giriş verisine uygulanacak konvolüsyon kernelleri, giriş verisinden elde edilen bilgiler kullanılarak ilgili veriye göre özelleştirilir ve statik konvolüsyon katmanları kullanan ağlara kıyasla benzer işlem yükü ile daha ayırt edici öznitelikler elde edilebilir. [2]'de tez kapsamında kullanılan omurga mimarisi için (ResNet-50 [3]) statik konvolüsyon ile gerçeklenen ağda işlem yükü 3,8G MAdds olarak raporlanırken kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanlarıyla gerçeklenen omurga mimarisinde 3,9G MAdds olarak raporlanarak dinamik konvolüsyonun çok fazla ek yük getirmediği gösterilmiştir.

Dinamik omurga mimarisi, literatürde bulunan iki farklı kişi tanılama ağında kullanılarak statik ve dinamik omurga mimarileriyle elde edilen performanslar karşılaştırılmıştır. Kullanılan kişi tanılama ağlarından biri olan Baseline ağı (ST-BL) [26], omurga mimarisine ek olarak genel öznitelikleri elde etmek için kullanılan havuzlama katmanı, boyut düşürmek için kullanılan bir konvolüsyon katmanı ve sadece eğitimde kullanılan bir FC katmanı içerir. Baseline, giriş görüntülerinin sadece genel özniteliklerini kullanarak eşleme yapan basit bir mimaridir. Kullanılan

diğer kişi tanılama ağı olan CaceNet ağında (ST-Cace) [5,10] ise omurga ağı ile çıkarılan öznitelik haritaları ve bu öznitelik haritalarından seçilen önemli piksel çiftleri kullanılarak elde edilen, görüntülerin birbirlerine göre koşullu öznitelik haritaları kullanılarak oldukça detaylı karşılaştırmalar sonucunda eşleme yapılır. CaceNet, bütün bu detaylı işlemler sayesinde yüksek performanslara ulaşabilir. Baseline ise basit ve az katmanlı bir ağ olduğundan gözlemlenen performanslar, CaceNet ağıyla karşılaştırıldığında daha düşüktür.

İki kişi tanılama ağında da omurga olarak ResNet-50 mimarisi (ST-ResNet-50) [3] kullanılmış ve dinamik omurga gerçeklemek amacıyla ST-ResNet-50'nin ilk katmanı ve izdüşüm bağlantıları dışındaki konvolüsyon katmanları kanal tümleştirme mekanizmalı dinamik konvolüsyon katmanlarıyla değiştirilerek DY-ResNet-50 elde edilmiştir. Statik ve dinamik omurga ağ mimarileri ile gerçeklenen Baseline ağlarında (ST-BL ve DY-BL) öğrenilebilir parametre sayıları sırasıyla 26.655.296 ve 30.704.328, statik ve dinamik omurgalı CaceNet ağlarında (ST-Cace ve DY-Cace) ise sırasıyla 30.398.176 ve 34.447.208'dir. Hem dinamik omurgalı, hem de statik omurgalı kişi tanılama ağlarının eğitiminde, ST-ResNet-50 ve DY-ResNet-50 ağları, ILSVRC-2012 veri setinde [11] sınıflandırıcı olarak ön-eğitilmiş şekilde kullanılmıştır. Kişi tanılama ağlarının eğitimleri, Market-1501 [12], DukeMTMC-reID [13], CUHK03 [14] ve Occluded-DukeMTMC [15] veri setlerinin eğitim setlerinde yapılmış olup çıkarım aşamasında da aynı veri setlerinin test setleri kullanılmıştır. ST-Cace ve DY-Cace ağlarında çıkarım aşamasında işlem maliyetini azaltmak amacıyla sadece omurga ağ mimarisi ile çıkarılan öznitelik haritaları kullanılarak eşlemeler yapılmıştır.

Kanal tümleştirme mekanizmalı dinamik konvolüsyon ile elde edilen ayırt edici özniteliklerin kişi tanılama ağlarında performansa etkisi nesne sınıflandırma ve kişi tanılama problemleri üzerinden raporlanmış olup nesne sınıflandırma performansının raporlanmasında SR, kişi tanılama performansının raporlanmasında ise mAP ve top-1 metriklerinden yararlanılmıştır. Karşılaştırma amacıylka, ST-BL ve DY-BL'de de benzer öğrenme performansları raporlanmıştır. 80 döngü eğitilmiş modellerle çıkarım sonucunda ise Baseline'da dinamik omurga ağ mimarisi kullanımının, çoğu durumda Re-ID performansını önemli ölçüde arttırdığı gözlemlenmiştir. ST-BL ile karşılaştırıldığında DY-BL'de, Market-1501, DukeMTMC-reID, CUHK03 ve

Occluded-DukeMTMC veri setleri ile çıkarımlarda sırasıyla %1,12, %2,31, %0,73 ve %2,24 daha yüksek mAP değerleri gözlemlenmiştir. top-1 bazında ise Market-1501, DukeMTMC-reID ve Occluded-DukeMTMC ile sırasıyla %0,63, %1,93 ve %3,08 artış gözlemlenirken CUHK03 veri setinde %0,57'lik bir düşüş gözlemlenmiştir.

ST-Cace ve DY-Cace'de ise 80 döngü eğitim sonunda, kullanılan performans metrikleri bazında öğrenme ve çıkarım performanslarında benzer değerler gözlemlenmiştir. CUHK03 ve Occluded-DukeMTMC veri setlerinde çıkarım adımında ise, sorgu görüntülerinin sırasıyla %99,43 ve %96,92'sinin ST-Cace ile karşılaştırıldığında DY-Cace'de aynı kimlikten galeri görüntülerine daha düşük ortalama uzaklıklarla eşlendiği görülmüştür. Market-1501 veri setinde ise bu oran %80,58'dir. Bu gözlemler sonucunda CaceNet'te dinamik omurga ağ mimarisi kullanımının, CUHK03, Occluded-DukeMTMC ve Market-1501 veri setleriyle çıkarımlarda eşlemelerin daha yüksek güvenle yapılabilmesini sağladığı görülür. DukeMTMC-reID ile çıkarımda ise DY-Cace ve ST-Cace ağları arasında eşleme uzaklığı bazında önemli bir fark görülmemiş olup ortalama eşleme uzaklığı azalan sorgu görüntüsü sayısının tüm sorgu görüntülerinin sayısına oranı %57,90'dır.

Buna ek olarak, eğitim ve çıkarım aşamalarında farklı döngülerde karşılaştırmalar yapıldığında CUHK03 veri setinde hem eğitimde sınıflandırma ve Re-ID performanslarının hem de çıkarımda Re-ID performansının, ilk döngülerde ST-Cace'yle karşılaştırıldığında DY-Cace'de daha yüksek olduğu görülmüştür. Yüksek döngü değerlerinde ise iki ağda da SR, mAP ve top-1 değerleri benzer seviyelere ulaşmıştır. Market-1501, DukeMTMC-reID ve Occluded-DukeMTMC veri setleri ile eğitim ve çıkarımda ise eğitim boyunca SR, mAP ve top-1 değerleri ve farklı döngülerde elde edilen modellerle çıkarımlarda mAP ve top-1 değerleri benzerdir.

Sonuç olarak, dinamik omurga ağ mimarisi kullanımının, Baseline gibi ayırt ediciliği sınırlı öznitelikler kullanan basit ağ mimarilerinde çoğu durumda performansı önemli ölçüde arttırabileceği gözlemlenmiştir. CaceNet gibi ayırt ediciliği yüksek özniteliklerle detaylı karşılaştırmalar sonucunda eşleme yapan ağlarda ise dinamik omurga ağ mimarisi kullanımının katkısının sınırlı olduğu görülmüştür. Ancak mimarilerdeki üst katmanlar omurga ağ öznitelik haritalarından kişi tanılamaya özgü

öznitelikler çıkarılmasını sağladığından, üst katmanlarda da dinamik konvolüsyon kullanılarak performansın arttırılması sağlanabilir.

Tez kapsamında yapılan eğitimlerde ve çıkarım testlerinde aynı veri setinden görüntüler kullanılmıştır. Çıkarımların genelleştirilebilmesi açısından, dinamik konvolüsyonlu kişi tanılamanın çapraz veri seti testlerindeki performansının raporlanması yararlı olacaktır.

Farklı veri setleri birlikte kullanılarak yapılacak dinamik eğitim ile kişi tanılama performansının arttırılması mümkündür. Bu amaçla, çoklu veri seti eğitimi için Market-1501, DukeMTMC-reID ve CUHK03 veri setleri birlikte kullanılarak dinamik ve statik omurga kullanan Baseline ve CaceNet ağlarında eğitim gerçeklenmiş, ardından her bir veri setinde test verisi üzerinde performans incelenmiştir. Ancak, çoklu veri seti ile eğitimde, donanımsal olanakların kısıtlılığı nedeniyle, gerekli "batch" boyutunda eğitim uygulanamadığından veri setlerinde istenilen kişi tanılama performans artışına erişilememiştir.

Dinamik ağlar son dönemde yazılımsal ve donanımsal gerçeklemede önemli bir araştırma alanını oluşturmaktadır. Bu kapsamda konvolüsyon katmanlarının yanı sıra aktivasyon fonksiyonları gibi işlemlerin de girişe göre özelleştirilmesini sağlayan öğrenme modelleri ve mimarileri geliştirilmektedir. Tez kapsamında gerçeklenen ağlara bu mimariler eklenerek ağın giriş verisine adaptifliği daha da arttırılabilir. Bunun yanı sıra, ileriye yönelik çalışmalarda kişi tanılama ağları, literatürde önerilen farklı dinamik konvolüsyon mekanizmalarına sahip mimarilerle gerçeklenebilir. Böylece farklı dinamik konvolüsyon mekanizmalarının kişi tanılama performansına etkileri gözlemlenebilir. Ayrıca literatürde önerilen dinamik konvolüsyonlu mimarilerde performans kazanımları genelde sınıflandırıcı olarak eğitilen ağlar için raporlanmıştır. Tez kapsamında ise dinamik konvolüsyonlu bir mimari, kişi tanılama mimarilerinde omurga ağı olarak kullanılarak sınıflandırma ve kişi tanılama performanslarının arttırılması hedeflenmiştir. Kullanılan dinamik konvolüsyonlu mimari, daha farklı uygulamalara yönelik ağlarda sınanabilir.

KAYNAKLAR

[1] **Chen, Y.**, **Dai, X.**, **Liu, M.**, **Chen, D.**, **Yuan, L. ve Liu, Z.** (2020). Dynamic convolution: Attention over convolution kernels, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.11030–11039.

[2] **Li, Y.**, **Chen, Y.**, **Dai, X.**, **Liu, M.**, **Chen, D.**, **Yu, Y.**, **Yuan, L.**, **Liu, Z.**, **Chen, M. ve Vasconcelos, N.** (2021). Revisiting dynamic convolution via matrix decomposition, *In Proc. International Conference on Learning Representations (ICLR)*.

[3] **He, K.**, **Zhang, X.**, **Ren, S. ve Sun, J.** (2016). Deep residual learning for image recognition, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.770–778.

[4] **Wei, W.**, **Yang, W.**, **Zuo, E.**, **Qian, Y. ve Wang, L.** (2021). Person re-identification based on deep learning-An overview, *Journal of Visual Communication and Image Representation*, 103418.

[5] **Yu, F.**, **Jiang, X.**, **Gong, Y.**, **Zheng, W.S.**, **Zheng, F. ve Sun, X.** (2022). Conditional Feature Embedding by Visual Clue Correspondence Graph for Person Re-Identification, *IEEE Transactions on Image Processing*, *31*, 6188–6199.

[6] **Yang, B.**, **Bender, G.**, **Le, Q.V. ve Ngiam, J.** (2019). Condconv: Conditionally parameterized convolutions for efficient inference, *In Proc. The 33rd Conference on Neural Information Processing Systems (NIPS)*, s.1307–1318.

[7] **Jia, X.**, **De Brabandere, B.**, **Tuytelaars, T. ve Gool, L.V.** (2016). Dynamic filter networks, *In Proc. The 30th Conference on Neural Information Processing Systems (NIPS)*.

[8] **Chen, Y.**, **Dai, X.**, **Liu, M.**, **Chen, D.**, **Yuan, L. ve Liu, Z.** (2020). Dynamic relu, *In Proc. European Conference on Computer Vision (ECCV)*, Springer, s.351–367.

[9] **Kouris, A.**, **Venieris, S.I.**, **Laskaridis, S. ve Lane, N.** (2022). Multi-exit semantic segmentation networks, *In Proc. European Conference on Computer Vision (ECCV)*, Springer, s.330–349.

[10] **Yu, F.**, **Jiang, X.**, **Gong, Y.**, **Zhao, S.**, **Guo, X.**, **Zheng, W.S.**, **Zheng, F. ve Sun, X.** (2021). Devil's in the Details: Aligning Visual Clues for Conditional

Embedding in Person Re-Identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[11] **Krizhevsky, A.**, **Sutskever, I. ve Hinton, G.E.** (2012). Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems (NIPS)*, *25*.

[12] **Zheng, L.**, **Shen, L.**, **Tian, L.**, **Wang, S.**, **Wang, J. ve Tian, Q.** (2015). Scalable person re-identification: A benchmark, *In Proc. IEEE International Conference on Computer Vision (ICCV)*, s.1116–1124.

[13] **Ristani, E.**, **Solera, F.**, **Zou, R.**, **Cucchiara, R. ve Tomasi, C.** (2016). Performance measures and a data set for multi-target, multi-camera tracking, *In Proc. European Conference on Computer Vision (ECCV)*, Springer, s.17–35.

[14] **Li, W.**, **Zhao, R.**, **Xiao, T. ve Wang, X.** (2014). Deepreid: Deep filter pairing neural network for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.152–159.

[15] **Miao, J.**, **Wu, Y.**, **Liu, P.**, **Ding, Y. ve Yang, Y.** (2019). Pose-guided feature alignment for occluded person re-identification, *In Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, s.542–551.

[16] **Hu, J.**, **Shen, L. ve Sun, G.** (2018). Squeeze-and-excitation networks, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.7132–7141.

[17] **Li, X.**, **Wang, W.**, **Hu, X. ve Yang, J.** (2019). Selective kernel networks, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.510–519.

[18] **De Lathauwer, L.**, **De Moor, B. ve Vandewalle, J.** (2000). A multilinear singular value decomposition, *SIAM journal on Matrix Analysis and Applications*, *21*(4), 1253–1278.

[19] **Zhao, H.**, **Tian, M.**, **Sun, S.**, **Shao, J.**, **Yan, J.**, **Yi, S.**, **Wang, X. ve Tang, X.** (2017). Spindle net: Person re-identification with human body region guided feature decomposition and fusion, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.1077–1085.

[20] **Sun, Y.**, **Zheng, L.**, **Yang, Y.**, **Tian, Q. ve Wang, S.** (2018). Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline), *In Proc. European Conference on Computer Vision (ECCV)*, s.480–496.

[21] **Wang, G.**, **Yuan, Y.**, **Chen, X.**, **Li, J. ve Zhou, X.** (2018). Learning discriminative features with multiple granularities for person re-identification, *In Proc. The 26th ACM International Conference on Multimedia*, s.274–282.

[22] **Zheng, F.**, **Deng, C.**, **Sun, X.**, **Jiang, X.**, **Guo, X.**, **Yu, Z.**, **Huang, F. ve Ji, R.** (2019). Pyramidal person re-identification via multi-loss dynamic training, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.8514–8522.

[23] **Szegedy, C.**, **Vanhoucke, V.**, **Ioffe, S.**, **Shlens, J. ve Wojna, Z.** (2016). Rethinking the inception architecture for computer vision, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.2818–2826.

[24] **Schroff, F.**, **Kalenichenko, D. ve Philbin, J.** (2015). Facenet: A unified embedding for face recognition and clustering, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.815–823.

[25] **Hermans, A.**, **Beyer, L. ve Leibe, B.** (2017). In defense of the triplet loss for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[26] **URL-1**. <*https://github.com/TencentYoutuResearch/PersonReID-YouReID*>, erişim tarihi: 12.05.2022.

[27] **URL-2**. <*https://github.com/liyunsheng13/dcd*>, erişim tarihi: 04.10.2021.

[28] **Loshchilov, I. ve Hutter, F.** (2017). Sgdr: Stochastic gradient descent with warm restarts, *In Proc. International Conference on Learning Representations (ICLR)*.

[29] **Yao, H.**, **Zhang, S.**, **Hong, R.**, **Zhang, Y.**, **Xu, C. ve Tian, Q.** (2019). Deep representation learning with part loss for person re-identification, *IEEE Transactions on Image Processing*, *28*(6), 2860–2871.

[30] **Song, C.**, **Huang, Y.**, **Ouyang, W. ve Wang, L.** (2018). Mask-guided contrastive attention model for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.1179–1188.

[31] **Chang, X.**, **Hospedales, T.M. ve Xiang, T.** (2018). Multi-level factorisation net for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.2109–2118.

[32] **Li, W.**, **Zhu, X. ve Gong, S.** (2018). Harmonious attention network for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.2285–2294.

[33] **Sun, Y.**, **Zheng, L.**, **Li, Y.**, **Yang, Y.**, **Tian, Q. ve Wang, S.** (2019). Learning part-based convolutional features for person re-identification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(3), 902–917.

[34] **Zhuang, Z.**, **Wei, L.**, **Xie, L.**, **Zhang, T.**, **Zhang, H.**, **Wu, H.**, **Ai, H. ve Tian, Q.** (2020). Rethinking the distribution gap of person re-identification with camera-based batch normalization, *In Proc. European Conference on Computer Vision (ECCV)*, Springer, s.140–157.

[35] **Wang, C.**, **Zhang, Q.**, **Huang, C.**, **Liu, W. ve Wang, X.** (2018). Mancs: A multi-task attentional network with curriculum sampling for person re-identification, *In Proc. European Conference on Computer Vision (ECCV)*, s.365–381.

[36] **Sun, Y.**, **Xu, Q.**, **Li, Y.**, **Zhang, C.**, **Li, Y.**, **Wang, S. ve Sun, J.** (2019). Perceive where to focus: Learning visibility-aware part-level features for partial person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.393–402.

[37] **Jin, X.**, **Lan, C.**, **Zeng, W.**, **Chen, Z. ve Zhang, L.** (2020). Style normalization and restitution for generalizable person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.3143–3152.

[38] **Yang, W.**, **Huang, H.**, **Zhang, Z.**, **Chen, X.**, **Huang, K. ve Zhang, S.** (2019). Towards rich feature discovery with class activation maps augmentation for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.1389–1398.

[39] **Zheng, M.**, **Karanam, S.**, **Wu, Z. ve Radke, R.J.** (2019). Re-identification with consistent attentive siamese networks, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.5735–5744.

[40] **Hou, R.**, **Ma, B.**, **Chang, H.**, **Gu, X.**, **Shan, S. ve Chen, X.** (2019). Interaction-and-aggregation network for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.9317–9326.

[41] **Kalayeh, M.M.**, **Basaran, E.**, **Gökmen, M.**, **Kamasak, M.E. ve Shah, M.** (2018). Human semantic parsing for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.1062–1071.

[42] **Zhang, Z.**, **Lan, C.**, **Zeng, W. ve Chen, Z.** (2019). Densely semantically aligned person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.667–676.

[43] **Zheng, Z.**, **Yang, X.**, **Yu, Z.**, **Zheng, L.**, **Yang, Y. ve Kautz, J.** (2019). Joint discriminative and generative learning for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.2138–2147.

[44] **URL-3**. <*https://github.com/TencentYoutuResearch/PersonReID-YouReID/blob/main/docs/model_zoo.md*>, erişim tarihi: 23.10.2022.

[45] **Chen, B.**, **Deng, W. ve Hu, J.** (2019). Mixed high-order attention network for person re-identification, *In Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, s.371–381.

[46] **Fang, P.**, **Zhou, J.**, **Roy, S.K.**, **Petersson, L. ve Harandi, M.** (2019). Bilinear attention networks for person retrieval, *In Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, s.8030–8039.

[47] **Chen, T.**, **Ding, S.**, **Xie, J.**, **Yuan, Y.**, **Chen, W.**, **Yang, Y.**, **Ren, Z. ve Wang, Z.** (2019). Abd-net: Attentive but diverse person re-identification, *In Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, s.8351–8361.

[48] **Zhao, S.**, **Gao, C.**, **Zhang, J.**, **Cheng, H.**, **Han, C.**, **Jiang, X.**, **Guo, X.**, **Zheng, W.S.**, **Sang, N. ve Sun, X.** (2020). Do not disturb me: Person re-identification under the interference of other pedestrians, *In Proc. European Conference on Computer Vision (ECCV)*, Springer, s.647–663.

[49] **Chen, X.**, **Fu, C.**, **Zhao, Y.**, **Zheng, F.**, **Song, J.**, **Ji, R. ve Yang, Y.** (2020). Salience-guided cascaded suppression network for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.3300–3310.

[50] **Zhang, Z.**, **Lan, C.**, **Zeng, W.**, **Jin, X. ve Chen, Z.** (2020). Relation-aware global attention for person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.3186–3195.

[51] **Yan, C.**, **Pang, G.**, **Jiao, J.**, **Bai, X.**, **Feng, X. ve Shen, C.** (2021). Occluded person re-identification with single-scale global representations, *In Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, s.11875–11884.

[52] **Chen, G.**, **Gu, T.**, **Lu, J.**, **Bao, J.A. ve Zhou, J.** (2021). Person re-identification via attention pyramid, *IEEE Transactions on Image Processing*, *30*, 7663–7676.

[53] **He, S.**, **Luo, H.**, **Wang, P.**, **Wang, F.**, **Li, H. ve Jiang, W.** (2021). Transreid: Transformer-based object re-identification, *In Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, s.15013–15022.

[54] **Wang, T.**, **Liu, H.**, **Song, P.**, **Guo, T. ve Shi, W.** (2022). Pose-guided feature disentangling for occluded person re-identification based on transformer, *In Proc. AAAI Conference on Artificial Intelligence*, cilt 36, s.2540–2549.

[55] **Wang, G.**, **Yang, S.**, **Liu, H.**, **Wang, Z.**, **Yang, Y.**, **Wang, S.**, **Yu, G.**, **Zhou, E. ve Sun, J.** (2020). High-order information matters: Learning relation and topology for occluded person re-identification, *In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, s.6449–6458.

[56] **Ren, X.**, **Zhang, D. ve Bao, X.** (2020). Semantic-Guided Shared Feature Alignment for Occluded Person Re-IDentification, *In Proc. Asian Conference on Machine Learning (ACML)*, PMLR, s.17–32.

[57] **Zhu, K.**, **Guo, H.**, **Liu, Z.**, **Tang, M. ve Wang, J.** (2020). Identity-guided human semantic parsing for person re-identification, *In Proc. European Conference on Computer Vision (ECCV)*, Springer, s.346–363.

[58] **Chen, P.**, **Liu, W.**, **Dai, P.**, **Liu, J.**, **Ye, Q.**, **Xu, M.**, **Chen, Q. ve Ji, R.** (2021). Occlude them all: Occlusion-aware attention network for occluded person re-id, *In Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, s.11833–11842.

[59] **Yang, J.**, **Zhang, J.**, **Yu, F.**, **Jiang, X.**, **Zhang, M.**, **Sun, X.**, **Chen, Y.C. ve Zheng, W.S.** (2021). Learning to know where to see: a visibility-aware approach for occluded person re-identification, *In Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, s.11885–11894.

[60] **URL-4**. <*https://github.com/lightas/Occluded-DukeMTMC-Dataset*>, erişim tarihi: 30.09.2022.

ÖZGEÇMİŞ

Adı SOYADI: Elif Ecem AKBABA

ÖĞRENİM DURUMU:

- **Lisans:** 2019, İstanbul Teknik Üniversitesi, Elektrik Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği
- **Y. Lisans:** 2023, İstanbul Teknik Üniversitesi, Elektronik ve Haberleşme Mühendisliği, Telekomünikasyon Mühendisliği

MESLEKİ DENEYİMLER VE ÖDÜLLER:

- (2020-Halen) İstanbul Teknik Üniversitesi - Araştırma Görevlisi