OBJECT AUGMENTATION FOR OUT-OF-CONTEXT OBJECT RECOGNITION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OĞUL CAN ERYÜKSEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

MAY 2022

Approval of the thesis:

OBJECT AUGMENTATION FOR OUT-OF-CONTEXT OBJECT RECOGNITION

submitted by **OĞUL CAN ERYÜKSEL** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Assoc. Prof. Dr. Sinan Kalkan
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Assist. Prof. Dr. Emre Akbaş
Computer Engineering, METU _____

Assoc. Prof. Dr. Sinan Kalkan
Computer Engineering, METU _____

Assoc. Prof. Dr. Nazlı İkizler Cinbiş
Computer Engineering, Hacettepe University _____

Date: 18.05.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Oğul Can Eryüksel

Signature :

ABSTRACT

OBJECT AUGMENTATION FOR OUT-OF-CONTEXT OBJECT RECOGNITION

Eryüksel, Oğul Can
M.S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Sinan Kalkan

May 2022, 42 pages

The visual context in an image contains rich information about and between foreground objects and the background. Deep learning models learn contextual information implicitly in general. However, since training datasets generally do not include all possible contexts, deep models tend to memorize contextual details. This can lead to poor recognition performance when models are deployed in real-world applications since objects may appear in unexpected contexts or places. These types of objects are called out-of-context objects. In this work, we propose an object-level augmentation framework for more robust recognition of out-of-context objects. Our proposed augmentation methodology applies object removal and object placement operations to images during the training phase. Moreover, we proposed a contrastive learning pipeline using object-level augmentations to increase performance further. Our results show that, by using object-level augmentations and contrastive learning, the out-of-context recognition performance of models can increase without losing performance on regular images. To analyze the effectiveness of the proposed method, we conducted a series of experiments for a multi-label image classification problem on the MS COCO dataset. Moreover, we provide a tool to generate images with

out-of-context objects using the proposed augmentation framework.

Keywords: Deep learning, Computer Vision, out-of-context objects, object removal, object placement, augmentation, multi-label classification

ÖZ

BAĞLAM DIŞI NESNE TANIMA İÇİN NESNE ÇEŞİTLİLİĞİ ARTIRIMI

Eryüksel, Oğul Can
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Sinan Kalkan

Mayıs 2022, 42 sayfa

Bir görüntüdeki görsel bağlam, ön plan nesneleri ve arka plan hakkında ve bunlar arasında zengin bilgiler içerir. Derin öğrenme modelleri, genel olarak bağlamsal bilgileri dolaylı olarak öğrenir. Bununla birlikte, eğitim veri kümeleri genellikle tüm olası bağlamları içermediğinden, derin modeller bağlamsal ayrıntıları ezberleme eğilimindedir. Bu, nesneler beklenmedik bağlamlarda veya yerlerde görünebileceğinden, modellerin gerçek dünya uygulamalarında düşük tanıma performansına yol açabilir. Bu tür nesnelere bağlam dışı nesneler denir. Bu çalışmada, bağlam dışı nesnelerin daha gürbüz bir şekilde tanınması için nesne düzeyinde bir çeşitleme çerçevesi öneriyoruz. Önerilen çeşitleme metodolojimiz, eğitim aşamasında görüntülere rastgele nesne kaldırma ve nesne yerleştirme işlemleri uygular. Ayrıca, performansı daha da artırmak için nesne düzeyinde çeşitleme kullanan bir karşılaştırmalı öğrenme hattı önerdik. Sonuçlarımız, nesne düzeyinde çeşitleme ve karşılaştırmalı öğrenme kullanılarak modellerin bağlam dışı tanıma performansının normal görüntülerde performans kaybetmeden artırılabileceğini göstermektedir. Önerilen yöntemin etkinliğini analiz etmek için, MS COCO veri setinde çok etiketli bir görüntü sınıflandırma problemi için bir dizi deney yapılmıştır. Ayrıca, önerdiğimiz nesne çeşitliliğini artırma

yöntemlerini kullanarak bağlam dışı nesnelerle görüntüler oluşturmak için bir araç sağlıyoruz.

Anahtar Kelimeler: Derin öğrenme, Bilgisayarlı Görü, bağlam dışı nesneler, nesne çıkarma, nesne ekleme, çeşitleme, çok etiketli sınıflandırma

This thesis is dedicated to my family and friends...

ACKNOWLEDGMENTS

I would like to thank my advisor, Assoc. Prof. Dr. Sinan Kalkan, for providing me with all kinds of support and guidance in writing this thesis.

I would also like to thank my thesis jury members, Assoc. Prof. Dr. Nazlı İkizler Cinbiş and Assist. Prof. Dr. Emre Akbaş for accepting to review my thesis.

I want to thank my teammates at the OBSS AI team for their valuable support and help, especially Sinan Onur Altınuç, Kamil Anıl Özfuttu, and Fatih Çağatay Akyön.

Moreover, I cannot forget to thank the Weights & Biases platform [1]. Without WandB, I would not have been able to track all my training and evaluation results.

Last but not least, I would like to thank my love, Dijan Teymur, for the effort and endless support you have shown me on this challenging path.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

MS COCO	Microsoft Common Objects in Context
mAP	Mean Average Precision
cGAN	Conditional Generative Adversarial Network
RGB	Red-Green-Blue channels for an image
HSV	Hue-Saturation-Value color space for an image
BCE	Binary Cross Entropy

CHAPTER 1

INTRODUCTION

1.1 Motivation: Out-of-Context Object Recognition

In computer vision problems, the importance of visual context in scene interpretation tasks is well-understood [9]. Unfortunately, to what extent image recognition models (should) rely on context to produce predictions is unclear. An image recognition model can easily be fooled when the context of a scene is different from the distribution of the training data (see Figure 1.1 for some examples). Therefore, it is crucial to determine these contextual dependencies to better apply our models in real-world scenarios.

The problem of poorer performance in situations such as those listed in Figure 1.1 can be attributed to *out-of-context* data distribution. These types of data generally may not be present in the training data. Furthermore, out-of-context data are not easy to find in public datasets or may be hard to collect. Although we can increase the amount of out-of-context data, finding data for all out-of-context scenarios is not possible.

Some examples for out-of-context images can be seen in Figure 1.1. In these out-of-context examples, the trained models manifest the memorized object-scene relations. For example, the couch and bed are misclassified as benches since they are in an outdoor context. However, beds and couches are mostly found in indoor environments in the training dataset.

Figure 1.1: Out-of-context image samples [2]. Under each image, multi-label classification model, that trained without augmentations, predictions are placed.

1.2 Proposed Methods and Models

Robustness against out-of-context data is an important challenge for many real-world applications. To address this for object recognition, we propose a method that can automatically remove and add random objects to the scenes, unlike image-level augmentations that perform transformations at the image level. With these augmentations, we can increase the generalization performance of a Deep Learning Model without sacrificing performance on original scenes.

Object-level augmentations consist of two object-level transformations:

1. Object Removal: Object removal augmentation is applied with a user-defined randomization probability as follows:
   (a) Extract the segmentation map and select the segment of an object.
   (b) Then remove corresponding pixels of the selected object from the scene.
   (c) Apply an image in-painting method to fill in the deleted pixels according to the background and the other objects in the scene.
2. Object Placement: Object placement augmentation is applied again with a user-defined probability as follows:
   (a) Take a segment for an object from a random image in the dataset.
   (b) Then paste the pixels of the selected object to a random location in the target image.

(c) Optional: While pasting objects, apply scaling, rotation or color change on the object pixels.

To evaluate the effectiveness of our method, we conduct experiments with a multi-label classification (object recognition) problem. For this, we have used the MS COCO Dataset [10]. To be able to conduct extensive experiments, we have utilized a mini version of the COCO Dataset [11], which is a carefully chosen subset of the original dataset that reflects the methods' performances on the original dataset. We have converted the COCO mini-train dataset to a multi-label classification dataset, having a label (tag) for each object category in the scene.

1.3 Contributions of the Thesis

Our contributions are as follows:

- Object-level augmentation for increasing the generalization performance of deep networks for out-of-context object recognition examples. Although object removal [12] and object placement augmentations [7], [13] are used in the literature separately, their combinations are not investigated for object recognition in the literature to the best of our knowledge.
- Applying geometric (rotation, flipping, translation etc.) and color (Gaussian blur, channel shuffle, color modifications etc.) transformations for object-level.
- A contrastive learning approach using our object-level augmentations to learn more robust representations for better performance on out-of-context objects.
- A new dataset of out-of-context images, constructed from the MS COCO validation 2017 dataset using our object placement augmentations.

1.4 The Outline of the Thesis

Chapter 2 reviews the related work and provides the background for the thesis research problem. In Chapter 3, we present our object-level augmentation pipeline in

detail. Furthermore, the details of the multi-label classification model are explained in Chapter 3. Then Chapter 4 describes our experimental setup, the experiments, and the results. Finally, Chapter 5 concludes the thesis with a summary and a discussion of limitations.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Out-of-Context Object Recognition

Visual context is a rich source of information about how the scene and objects are related. This relationship can be between the relative positions of objects or where they are located in the scenes. Some objects may be more likely to coexist in the same environment, e.g., trees and birds. Furthermore, particular objects are more likely to coexist with a special relationship in specific environments, like cars on the road. Objects that are in different environments or in combination with different objects than they usually should be are called "out-of-context" objects [3]. Recognizing out-of-context objects is generally more challenging than recognizing in-context objects since recognition models "learn" relations between context and objects. Increasing the recognition performance on out-of-context objects provides an important step for solving real-world problems more efficiently.

Deep neural network models can quickly memorize information between objects and the scene context. Furthermore, they tend to memorize contextual information between objects that are seen together often. Object removal augmentations may help models overcome these types of situations. For example, Shetty et al. [12] showed that the object removal approach could increase the robustness of model predictions in out-of-context object conditions.

There have been many studies focusing on out-of-context object recognition. For example, Choi et al. [3] proposed a method to identify out-of-context objects and scenes. Their method relies on using contextual relationships between objects. They have created a scene-object relation graph for each scene and built a model that can

use such a graph, as seen in Figure 2.1.

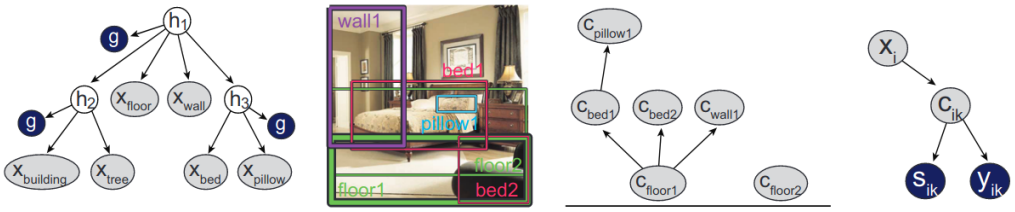

Figure 2.1: Scene-object context relation graph mechanism of Choi et al. [3] (Image taken from [3]).

Shetty et al. [12] used data augmentation based on object removal to investigate the nature of context relation of objects for semantic image segmentation and classification problems. Their approach showed that object removal-based data augmentation could increase accuracy on out-of-context conditions without compromising performance on common data scenarios.

Divornik et al. [13] utilize instance segmentation annotations in the augmentation pipeline. They used visual contextual information in an image to find an appropriate location while placing an object. They show that with the help of visual contextual modeling, object placement augmentations can increase mean average precision (mAP) for object detection problems.

Ghiasi et al. [7] proposed a simple copy-and-paste data augmentation scheme to increase accuracy, for instance segmentation problems. Their method copies random instances to the target image. By applying these augmentations, they have improved the mean average precision (mAP) score on the MS COCO dataset.

Ntvalies et al. [14] proposed a framework for semantic scene editing to add, manipulate or erase objects. Their method depends on Generative Adversarial Networks (cGANs) to edit images. This type of image editing is getting more and more widely used.

2.2 Deep Image Classification

Image classification is one of the most known and studied problems in Computer Vision. The image classification task aims to recognize the foreground objects in the scene. Deep Learning based approaches have become very popular and dominant in Computer Vision after the seminal AlexNet model [15] achieved incredible performance improvement on ImageNet Classification Challenge [16]. After the extraordinary success of the AlexNet, better, more complex, and more robust Deep Neural Network architectures are proposed. Even today, new architectures appear in literature, pushing the state-of-the-art performance on the ImageNet challenge. However, out-of-context prediction problems have not been sufficiently studied or addressed, even with the most complex model architectures.

2.2.1 Multi-label Image Classification

In conventional image classification, images contain only one label. On the other hand, scenes may have more than one label in multi-label image classification. Therefore, multi-label classification can be considered an extended version of the classification problem. Although the same network architectures may be used for multi-label classification, some changes need to be applied in practice: For example, the loss function and the evaluation metrics should be changed for multi-label classification.

2.3 Image Augmentation

Deep Neural Networks have accomplished astonishing results in Computer Vision tasks. However, they generally depend on big data to achieve remarkable performance. Unfortunately, a limited amount of labeled data is available for some Computer Vision tasks. Also, it can be tricky (or impossible sometimes) to increase labeled data. To overcome these issues, data augmentation methods are generally used. Moreover, data augmentation is a beneficial method to increase the generalization of deep models. With data augmentation, Deep Neural Networks can perform more robustly in real-world conditions [17].

In this section, different image augmentation strategies are explained.

2.3.1 Image-Level Augmentations

Image-level augmentations are transformations that modify color, orientation, size, etc. Examples include horizontal/vertical flip, rotation, random cropping, color space transformations, noise addition, etc. Image-level augmentations are very popular and widely applied in Computer Vision problems. They are accommodated to increase data diversity and generalization. In Figure 2.2, various widely-used image-level augmentations are visualised.

Figure 2.2: Some image-level augmentation examples (Image taken from [4]).

2.3.2 Patch-Level Augmentations

Patch-level augmentations aim to alter or transform patches in images. Erasing random patches from an image [5], and copying & pasting random patches to an image [6] are just two examples of patch-level augmentations. Patch-level augmentations are successful in reducing model overfitting. Also, they are beneficial in increasing the robustness of occluded objects. In Figure 2.3, various types of patch erasing augmentation scenarios are visualized. For particular types of problems, different erasing strategies may be applied.

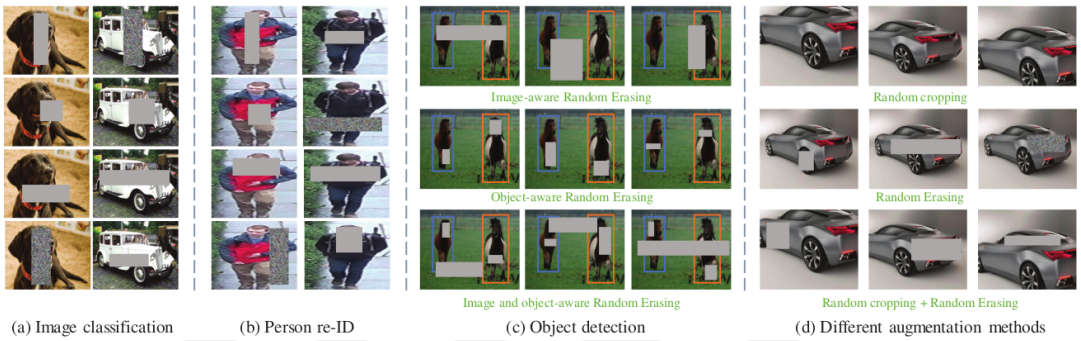

Figure 2.3: Different type of patch erasing augmentations (Image taken from [5])

Copying and pasting random patches are useful to increase model accuracy for classification problems. Moreover, they are beneficial in making models more robust to adversarial attacks. Examples of copy-paste patch augmentations can be seen in Figure 2.4.

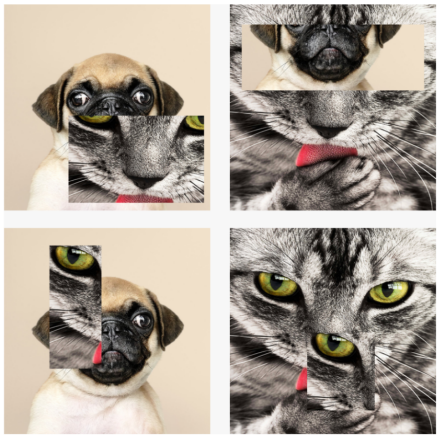

Figure 2.4: Resulting examples for copy-paste patch examples (Image taken from [6])

2.3.3 Copy-Paste Object Augmentations

Copy-paste object augmentations differ from patch-level augmentations in that copy-paste augmentations focus on the objects to augment. They do not copy-paste random patches but random objects to scenes. Ghiasi et al. [7] used copy-paste object augmentations for instance segmentation problem. They showed that copy-paste object augmentations are extremely useful for increasing rare class prediction accuracy. In Figure 2.5, their proposed copy-paste augmentations scheme can be seen.

Figure 2.5: Object copy-paste augmentations (Image taken from [7])

2.3.4 Object Removal Augmentations

Object removal augmentations can be considered a combination of two tasks: Extracting the desired object area from the image and in-painting the pixels of the removed object in the image [12]. Segmentation (or instance segmentation) models can be used to extract a desired object's pixels. Also, if dataset annotations contain segmentation maps, they can be used directly. On the other hand, filling the gaps in the image after object removal is a more challenging task. Gaps must be filled realistically. Otherwise, some bizarre artifacts may occur in the scene, leading to undesired performances for a model. To address in-painting, several network architectures are proposed in the literature [18, 19, 20, 21].

2.4 Discussion

As seen in the detailed literature review above, object-level augmentation strategies are not exploited sufficiently well for computer vision problems. Object-removal [12] and simple object placement [7] augmentations were utilized but only independently. However, to the best of our knowledge, no work has addressed combining both object-level augmentations – see Table 2.1 for a comparison. In this thesis, we studied the effects of object-level removal and placement augmentations separately and together. Moreover, we integrated image-level augmentations into object-level while applying placement augmentations. Furthermore, we proposed a contrastive learning approach to utilize object-level augmentations more effectively.

Table 2.1: Comparison between applications of different types of augmentations. *Image Level*, *Obj. Rem.*, *Obj. Place.*, *Obj. Trans.* columns are explain image level, object removal, object placement and object-level transformation augmentations are used or not respectively.

	Augmentation Type			
Study	*Image Level*	*Obj. Rem.*	*Obj. Place.*	*Obj. Trans.*
Krizhevsky et al. [15]	✓	✗	✗	✗
Shetty et al. [12]	✗	✓	✗	✗
Ghiasi et al. [7]	✗	✗	✓	✗
Ours	✗	✓	✓	✓

CHAPTER 3

PROPOSED METHOD: OBJECT-LEVEL AUGMENTATION

Object-level augmentations have been attracting increasing attention in the literature. These augmentations can increase the generalization of deep learning models in Computer Vision. Moreover, they can be used to increase the amount of training data, where there are a limited amount of labeled data in some cases. In this work, we used object-level augmentations: Object Removal and Object Placement.

3.1 Overview

Under this section, details of the Object-Level Augmentation methodology are explained. Our proposed object-level augmentation workflow is explained in Figure 3.1. Even though we follow the MS COCO dataset format, it is possible to use these augmentations for other dataset formats. Object removal augmentations must be applied in our proposed pipeline before object placement augmentations, otherwise, some of the placement augmentations may disappear. In other words, augmentations may cancel out each other. It is important to apply object removal augmentations at first not to decrease the effect of augmentations.

3.2 Object Removal Augmentation

We have created a pipeline to generate object removal augmentations. The object removal augmentation pipeline can be seen in Figure 3.2. In this pipeline, we have used the Mask R-CNN [22] instance segmentation model trained on the MS COCO dataset. However, it is possible to use any other segmentation model. Moreover,

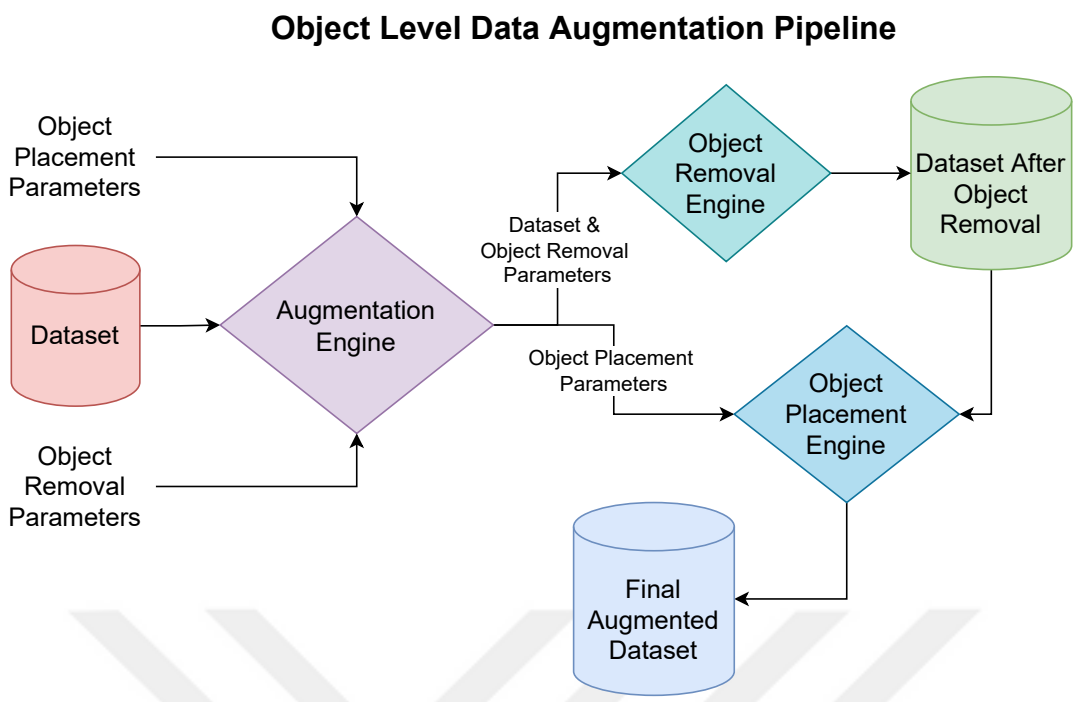

Figure 3.1: The proposed pipeline for Object-Level Augmentations.

ground truth segmentation maps can be used if available.

For in-painting, we have tried two different models. The first one is the pre-trained EdgeConnect in-painting model [18]. The second one is the pre-trained LaMa model [19]. The LaMa in-painting model provided visually more pleasing in-painting and more robust results in our experiments. Therefore, we decided to use the LaMa in-painting model in our object removal augmentation pipeline.

Moreover, we have designed a modular object removal pipeline. Thus, it is possible to use any other in-painting model inside the object removal pipeline. Some example results for object removal augmentations can be seen in Figure 3.3.

Before applying object removal augmentations, we used some predefined rules to get more feasible results:

- If the target scene does not contain more than two different object categories, we do not use object removal augmentation to avoid creating images with no objects.

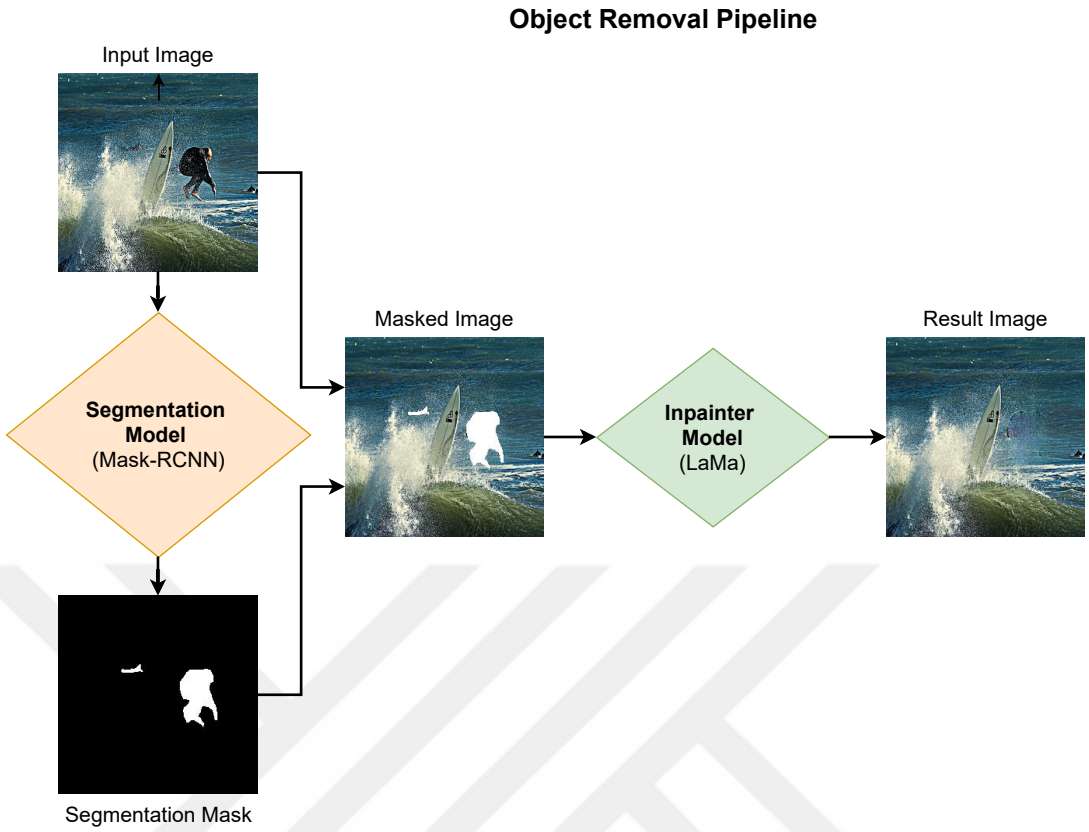

Figure 3.2: The proposed pipeline for object removal augmentations.

- For an object to be deleted, we are limited to the area of the object. If the object we are trying to remove occupies more than 75% of the image, we do not remove this object. Since in-painting models fail to fill large areas in the scene, we want to prevent strange in-painting results.

- Sometimes, removing an object results in the deletion of another object from the scene. This can happen when objects overlap. To overcome this problem, we have placed back any other objects that are deleted unintentionally.

3.3 Object Placement Augmentation

Our object placement augmentation pipeline can be seen in Figure 3.4. We use the same pre-trained instance segmentation model used in the Object Removal pipeline.

Figure 3.3: Object removal samples. The first line: the original images. The second line: Images with an object removed.

We first extract a segmentation map from a random image in the dataset to apply object placement augmentation. After, we use this segmentation map to place objects in the target image. Figure 3.5 displays some output images that are created using our object placement pipeline.

3.4 Contrastive Learning

Contrastive learning is one of the popular methods for learning robust representations by contrasting similar and dissimilar samples. Chopra et al. [23] suggested a contrastive training method to use in-face verification problems. This was the first usage of contrastive learning in deep learning. After this work, many methods have been proposed for contrastive learning [24, 25, 26]. Contrastive learning approaches can be useful if the amount of labeled data is limited. Moreover, it can be used to increase generalization performance for supervised learning tasks.

Traditional contrastive learning methods use negative and positive samples during training for image classification. However, our problem, i.e. multi-label image classification, is not suitable for traditional methods that rely on making feature embeddings closer or distant: While augmenting, we are adding or removing some objects

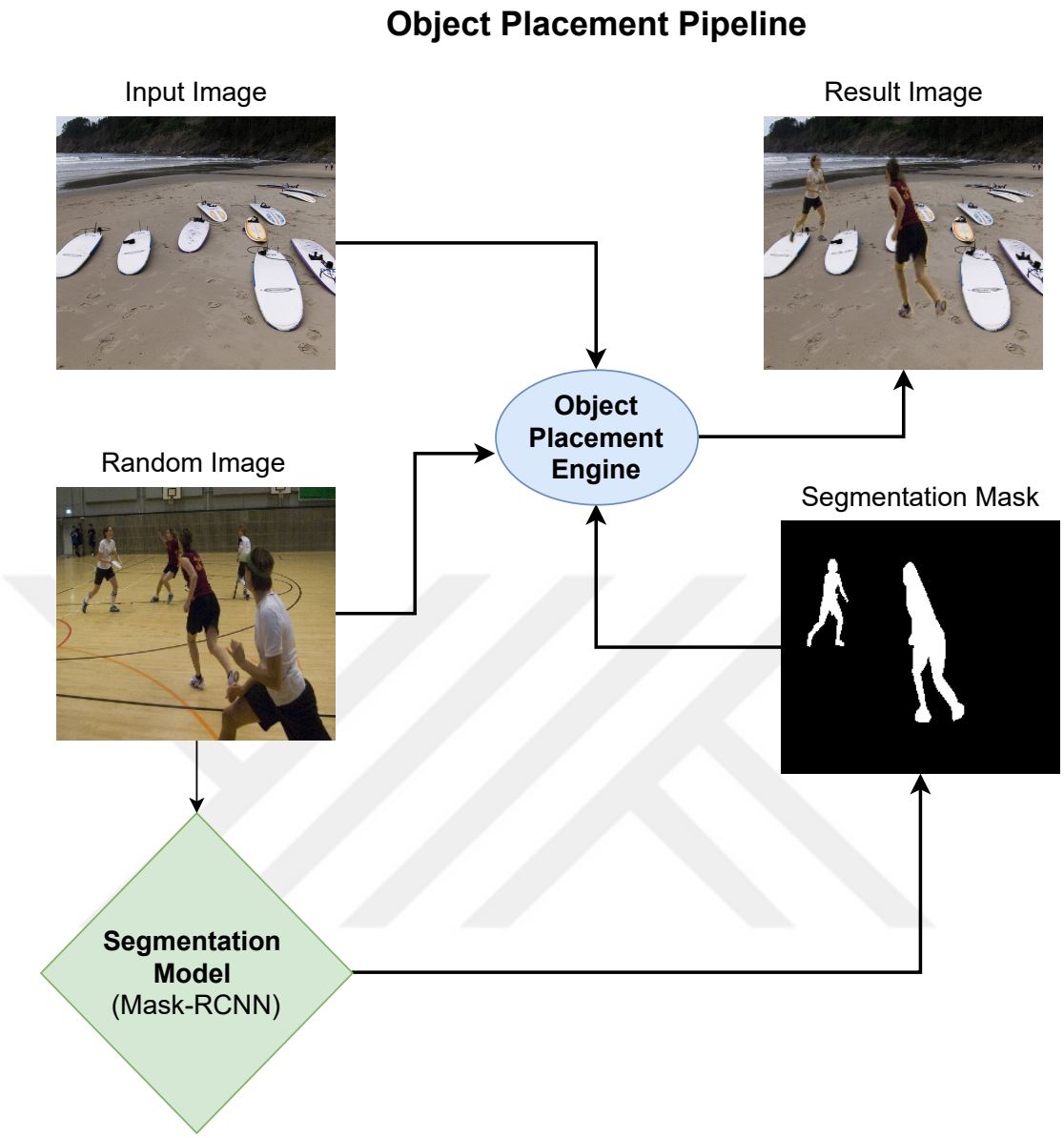

Figure 3.4: The proposed pipeline for object placement augmentations.

from the scene. Since this alters the labels of the original image, it is not feasible to directly formulate a contrastive objective based on visual feature vectors.

We have proposed a training pipeline with a contrastive loss term to facilitate contrastive learning. Details of our contrastive learning pipeline can be seen in Figure 3.6. Our contrastive learning pipeline uses both original image and its corresponding object-level augmented image. Both images are fed to a ResNet-50 model. $Sigmoid$ output vector of both images used to compute loss with respect to their corresponding

Figure 3.5: Object placement sample results. First line: The original images. Second line: Images with additional objects.

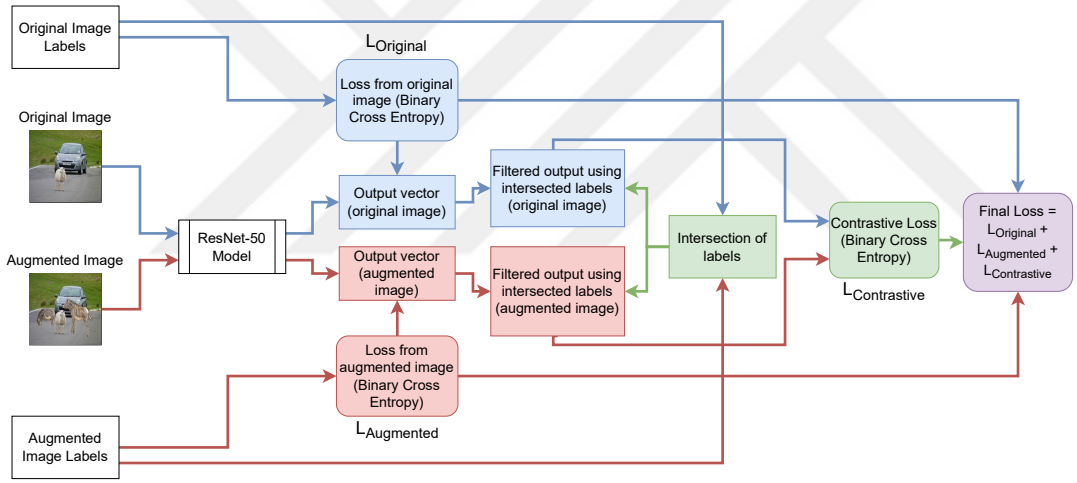

Figure 3.6: Proposed contrastive learning pipeline.

ground truth label, $L_{Original}$ and $L_{Augmented}$ respectively. We added a new contrastive loss term called $L_{Contrastive}$ to facilitate contrastive learning. To compute this loss first, we get the intersection of ground truth labels of the original image and augmented image. Then, output vector values (for both original and augmented) are filtered using these intersected ground truth labels. Finally, we computed BCE loss between these filtered output vectors. In the last step, we have summed up these losses, $L_{Original}$, $L_{Augmented}$, and $L_{Contrastive}$, to create the final loss. After backpropagation is applied with using this final loss. With the help of the contrastive loss term,

we are able to make a closer prediction of two images on the same labels.

CHAPTER 4

EXPERIMENTS AND RESULTS

In this chapter, our experimental setup and experiments are presented.

4.1 Experiment and Training Details

To evaluate the effectiveness of our methodology, we follow a multi-label image classification setting.

4.1.1 Network Architecture and Training Details

There are many deep learning models in the literature that we can use for image classification. In our experiments, we chose the ResNet-50 [27] architecture since it is considerably straightforward, easy to use, and widely used. We have modified the original network architecture to make it suitable for a multi-label image classification problem. The Adam optimizer [28] is chosen for training the networks with its default parameters. We have trained our models for a maximum of 10 epochs with early stopping using the validation loss.

4.1.2 Datasets

We chose to use the MS COCO dataset [10] for our experiments. For training, to be able perform more extensive experiments, a small version of the training dataset, called COCO mini-train [11], is used. The mini-train dataset has 25k images with 80 object categories. For validation, we created a new dataset from samples not included

in mini-train. For testing, we used the original MS COCO val2017, which contains 5k images with 80 object categories.

The MS COCO dataset is commonly used for object detection and instance segmentation problems. We have converted this dataset to a multi-label classification format: For each image, the labels are the labels of the bounding boxes in the image.

4.1.3 UnRel Dataset

Peyre et al. [8] proposed a weakly-supervised method to learn visual relations between pairs of objects. To demonstrate their methodology's effectiveness, they released a dataset named UnRel. This dataset contains 1000 images with unusual object relations. Although this dataset may not be perfect for out-of-context scene conditions, it is advantageous to investigate out-of-context object relations. Since Deep Neural Networks also tend to memorize contextual information between objects in the scene, the UnRel dataset is highly beneficial for investigating this problem for trained models. Therefore, we have also decided to use the UnRel dataset to evaluate our augmentations' impact. In Figure 4.1, example images from the UnRel dataset are displayed.

4.1.4 Out-of-Context Object Placements Dataset

We have created a new dataset using our Object Placement augmentations on the MS COCO Validation 2017 dataset to evaluate out-of-context recognition performance. This dataset contains 1000 images with 77 object categories. While creating this dataset, contextual scene information has been used from the MS COCO Validation 2017 dataset. Placement probabilities are highly increased for vehicle objects (car, train, truck, bus, etc.) for in-door scenes and sky. Furthermore, placement probabilities for wild animals (horse, zebra, elephant, etc.) increased for targeting city scenes and in-door places. The resulting examples of our new created out-of-context dataset can be seen in Figure 4.2.

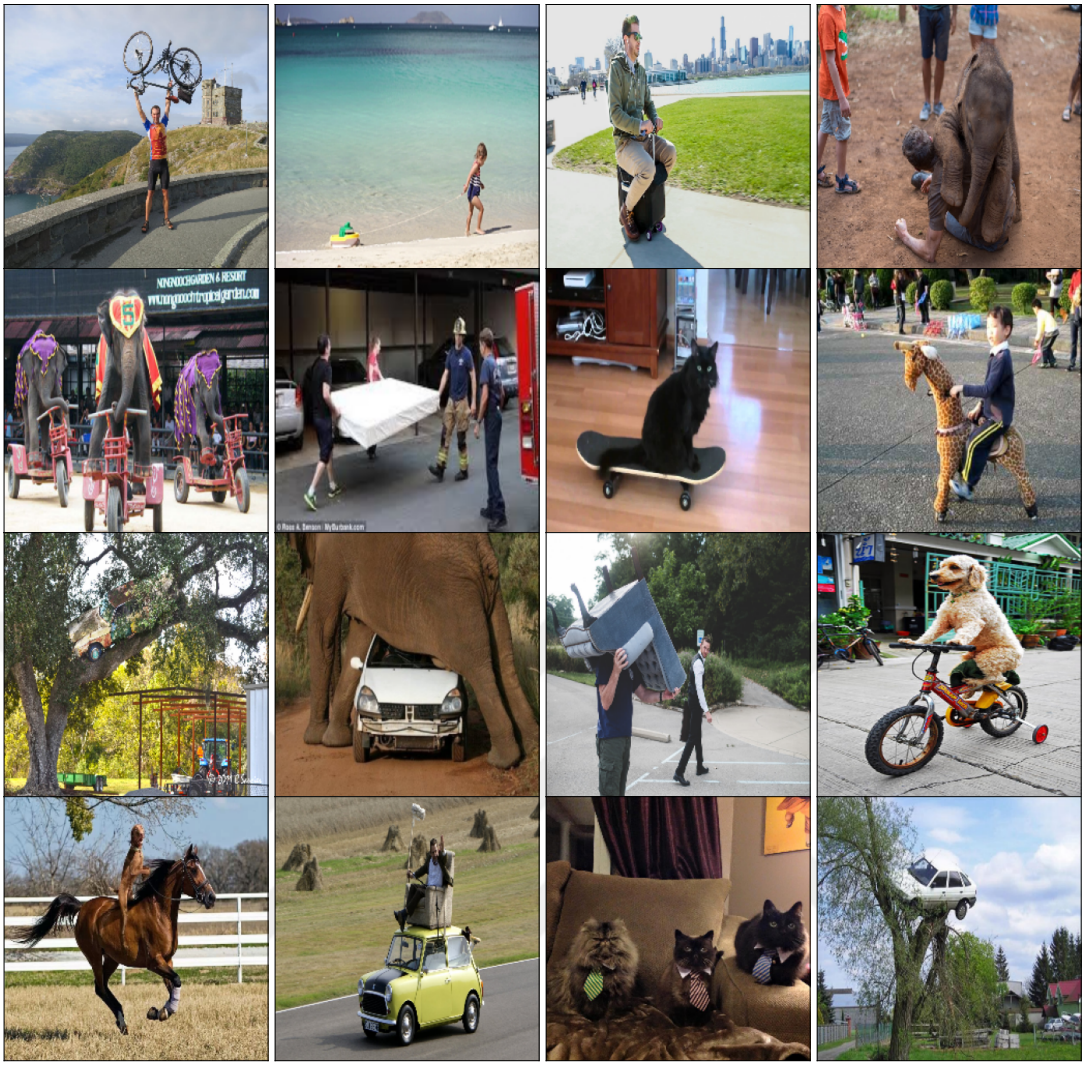

Figure 4.1: Sample images from the UnRel dataset [8].

4.1.5 Evaluation Measures

Many measures are used in multi-label classification, extending binary classification evaluation measures. F_1, $Precision$, $Recall$, $Accuracy$ measures are widely used in binary classification problems. In multi-label classification, those metrics are computed via *macro averaging* and *micro averaging*. Macro averaging is performed by computing a measure on individual class labels and then averaging the values over all classes. On the other hand, micro averaging calculates the measure globally on all instances and all class labels [29]. The measures obtained by micro-averaging and

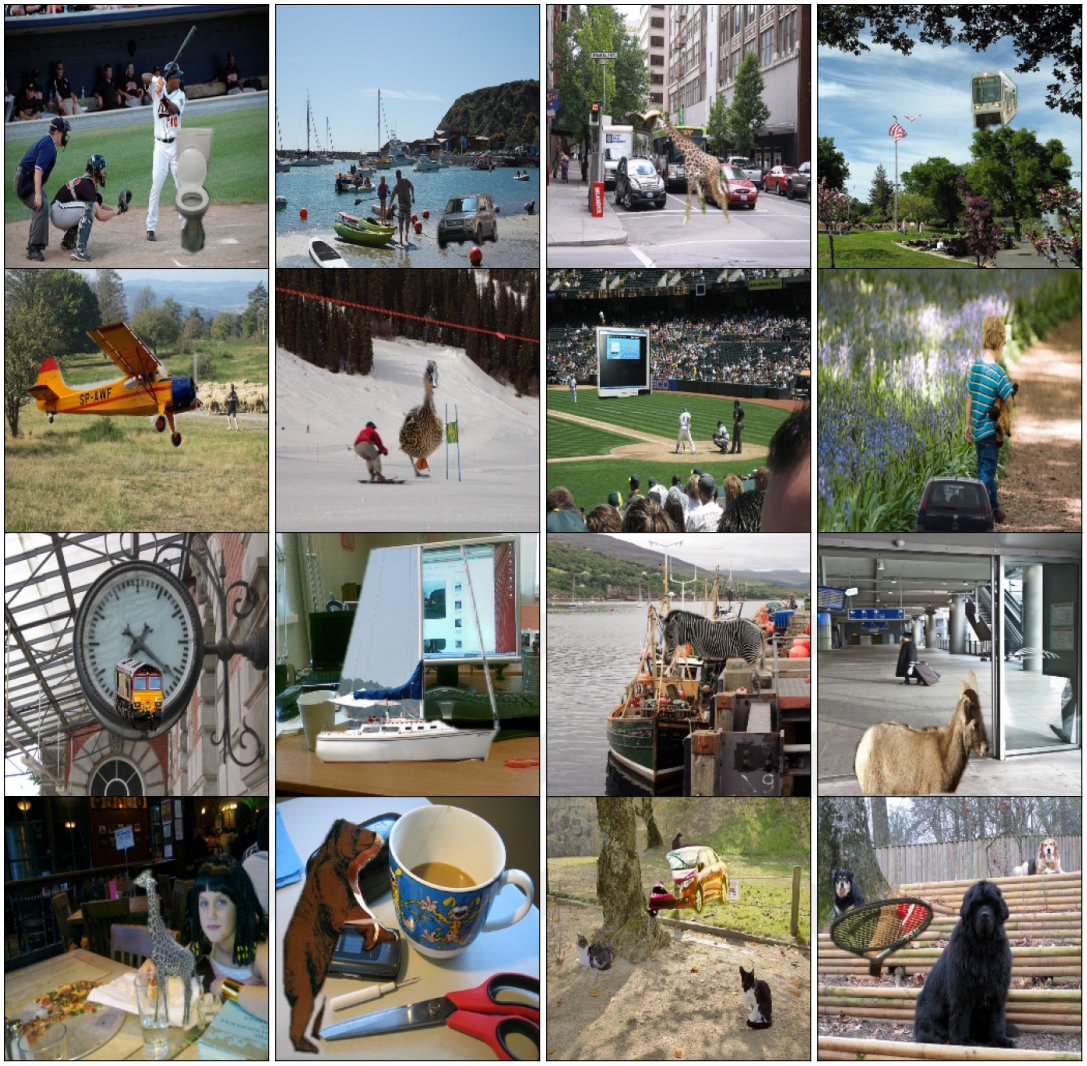

Figure 4.2: Sample images from our out-of-context dataset, created using object placement augmentations on the MS COCO Validation 2017 dataset.

macro-averaging are defined as follows (definitions are adapted from [29]):

$$\text{Accuracy:} \quad A = \frac{1}{k}\sum_{i=1}^{k}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}, \tag{4.1}$$

$$\lambda\text{-Precision:} \quad P^{\lambda}_{macro} = \frac{\sum_{i=1}^{n} Y_i^{\lambda} Z_i^{\lambda}}{\sum_{i=1}^{n} Z_i^{\lambda}}, \tag{4.2}$$

$$\lambda\text{-Recall:} \quad R^{\lambda}_{macro} = \frac{\sum_{i=1}^{n} Y_i^{\lambda} Z_i^{\lambda}}{\sum_{i=1}^{n} Y_i^{\lambda}}, \tag{4.3}$$

$$F^{\lambda}_{1-macro} = \frac{2\sum_{i=1}^{n} Y_i^{\lambda} Z_i^{\lambda}}{\sum_{i=1}^{n} Y_i^{\lambda} + \sum_{i=1}^{n} Z_i^{\lambda}}, \tag{4.4}$$

$$\text{Precision:} \quad P_{macro} = \frac{1}{k}\sum_{i=1}^{k} P^{\lambda}_{macro}, \tag{4.5}$$

$$\text{Recall:} \quad R_{macro} = \frac{1}{k}\sum_{i=1}^{k} R^{\lambda}_{macro}, \tag{4.6}$$

$$F_{1-macro} = \frac{1}{k}\sum_{i=1}^{k} F^{\lambda}_{1-macro}, \tag{4.7}$$

where λ denotes a class, k is the number of samples, n is the number of labels, Y_i is the set of predicted labels for a sample and Z_i is the set of true labels for a sample.

In our experiments, we decided to use $F_{1-macro}$ and Accuracy scores as they capture the different aspects of classification performance and are widely used. The accuracy measure is slightly different for multi-label classification problems. In multi-label classification, accuracy is calculated as in Equation 4.1 and named Hamming Score.

4.1.6 Implementation Details

In general, augmentations are applied on-the-fly while training a model. However, it was not feasible in our case because of the high computational requirements of our object-level augmentations. For example, object removal augmentations may increase the training time for epoch up to 10 times in our training set (changes with augmentation probability). Therefore, we first created the datasets with different augmentations and then trained models on these augmented datasets.

We have created 35 different datasets to examine the effectiveness of our object-level augmentations. As a $baseline$ dataset, we used the original coco-mini train without object-level augmentations. Later, a new dataset is created for different object-level augmentation probabilities. Augmentation probabilities range between 0.0 and 1.0, where 0.0 means no augmentation, and 1.0 means 100% augmentation probability, with an increase of 0.2 probability. This results in 35 different datasets.

- Coco Mini-Train: Original training dataset, without object-level augmentations. This dataset is used as a baseline.

- Coco Mini-Train with Object Removal Augmentations: Only object removal augmentations are applied.
- Coco Mini-Train with Object Placement Augmentations: Only object placement augmentations are applied.
- Coco Mini-Train with Both Object Level Augmentations: Object placement and removal augmentations are applied.

All experiments are trained for multi-label classification problems with the ResNet50 model. ResNet50 model is initialized with pre-trained ImageNet weights for all experiments. Adam optimizer [28] is used with its default values in training. As a validation dataset, we constructed a new dataset using randomly selected 5k samples from the MS COCO dataset, that are not included in the COCO mini-train dataset. Each model is trained for ten epochs, and the best model is saved using validation loss calculated using our validation dataset.

4.2 Experiment 1: Object-Level Augmentations

In this section, we evaluate the effect of different object-level augmentations by varying their probabilities. We visualize our results in a heatmap over a 6x6 matrix in Figure 4.3. We see that object-level augmentations significantly increase (61.50 vs. 60.10) in accuracy compared to the baseline (probability of 0 for both augmentations). However, we note that increasing the augmentation probability after 0.4 results in a poor accuracy score. Moreover, $F_{1-macro}$ scores are visualized in Figure 4.4, where we see that augmentation probability of 0.4 provides the best performance.

4.2.1 Experiments with More Object-level Transformations

Geometric and color augmentations are extremely popular in computer vision problems. To increase data diversity, they are used to alter images. However, these augmentations are applied on the image level. Here, we apply image-level augmentations at the object level. In this way, we can increase the diversity of our object-level augmentations. Although there are too many options for image augmentations, we chose

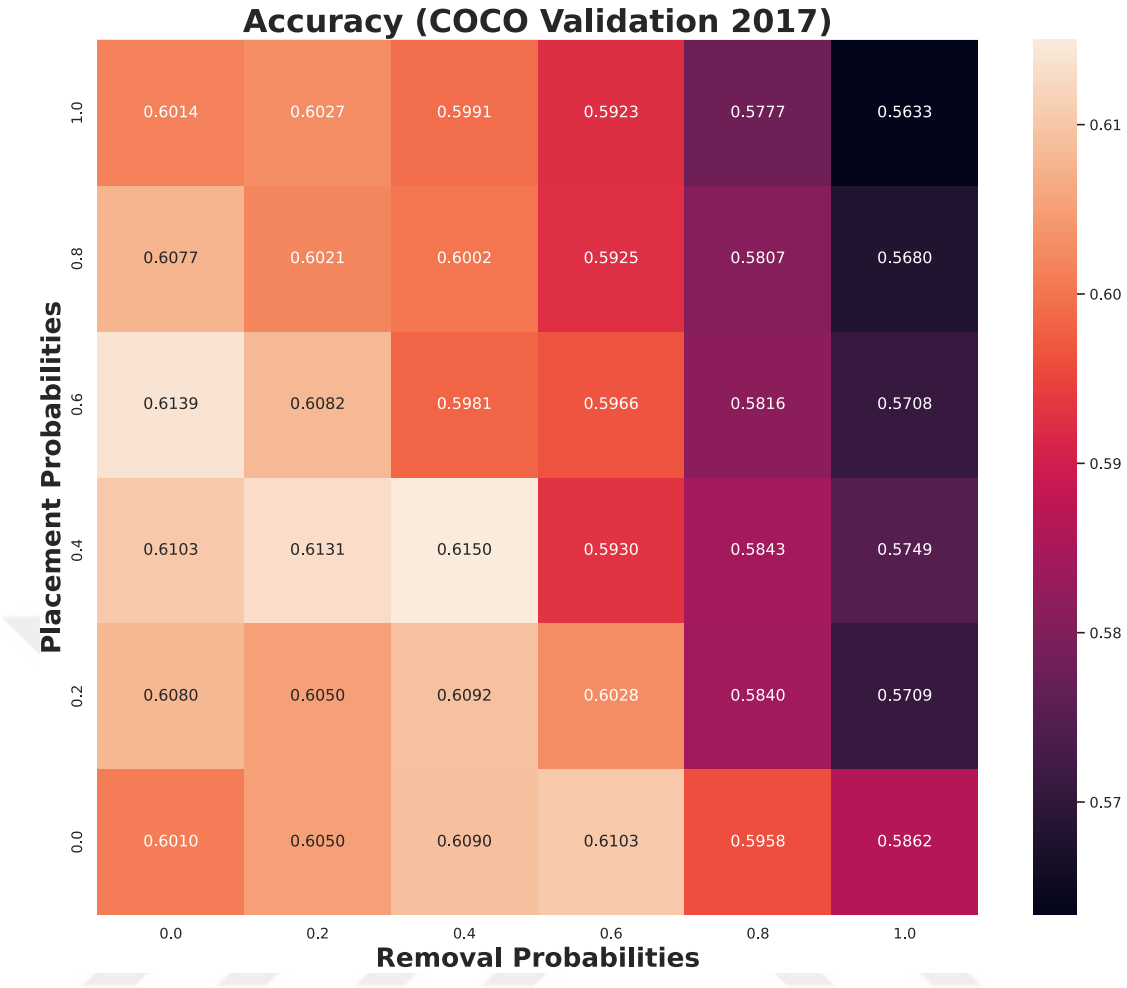

Figure 4.3: Accuracy score comparison for COCO Validation 2017 dataset between different probabilities of object-level augmentations.

a commonly used [17] subset of these augmentations to apply object-level. Chosen augmentations are divided into two main categories: geometric transformations and color transformations. Geometric transformations only alter geometric properties of the object and we choose resize (make smaller or bigger), rotation, shear (similar to rotation but includes stretching also), translation (place to a random location in image space), flipping left-to-right and flipping upside-down. Results of geometric transformations on object-level can be seen in Figure 4.5. On the other hand, color augmentations modify RGB values of objects. Applied color transformations are; Gaussian blur (blurs the object using a Gaussian kernel), sharpen (increase details of the object and color contrast), grayscale (converts objects from RGB to grayscale), channel shuffle (randomly shuffles RGB channels of an object), gamma contrast (modifies the

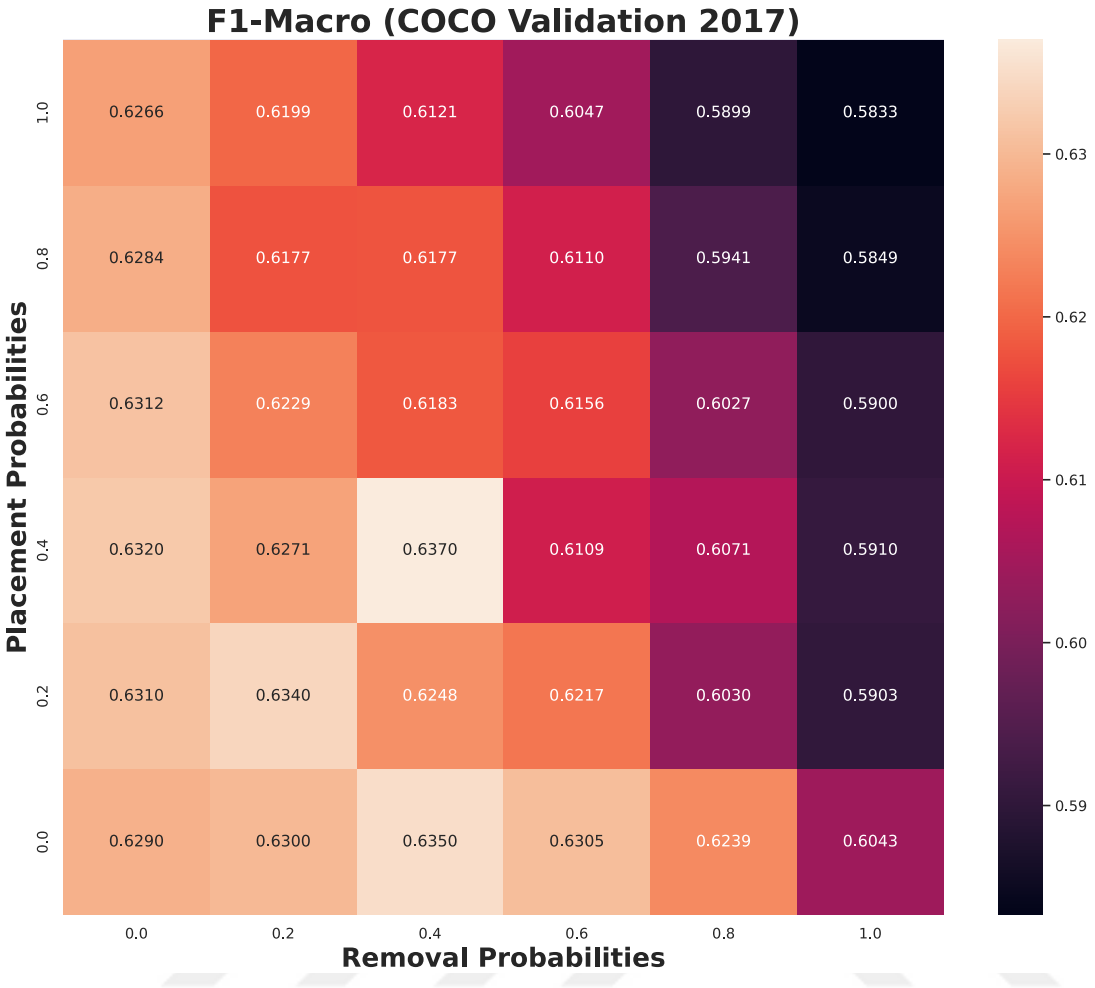

Figure 4.4: $F_{1-macro}$ score comparison for COCO Validation 2017 dataset between different probabilities of object-level augmentations.

contrast of the object with given gamma value), multiply saturation (converts image to HSV color space, and multiplies saturation channel with given value), add to hue and saturation (converts image to HSV color space, and adds given value to both hue and saturation channels). Sample results are provided in Figure 4.6. Imgaug library [30] is utilized while implementing object-level transformation augmentations, except for rotation and translation operations. Although rotation operation is included in imgaug library, custom implementation is needed not to lose details of the object. Moreover, since translation operation is not available in imgaug library, it is implemented manually.

We conducted experiments to see the effects of object-level transformations. We investigated the impact of only geometric transformations, only color transformations,

Table 4.1: Accuracy score comparison between different object-level transformation augmentation applications on the dataset MS COCO Validation 2017.

Model	*Obj. Place.*	*Geo. Trans.*	*Color Trans.*	*Accuracy*
Baseline	✗	✗	✗	0.6020
Place. only	✓	✗	✗	0.6095
Place. + Geo.	✓	✓	✗	0.6151
Place. + Color	✓	✗	✓	0.6148
Place. + Geo. + Color	✓	✓	✓	**0.6186**

and combinations of both augmentations. Figure 4.7 shows few resulting samples of combination of augmentations.

Figure 4.5: Object-level geometric transformations.

Experiments have been conducted with different augmentation probabilities in the range of 0 and 1 with an increasing step of 0.1. In all experiments, object placement probability is kept fixed at 0.5, and only geometric, and color transformation augmentation probabilities are altered. In these experiments, we have observed the best results are obtained with a 0.5 augmentation probability for geometric and color transformations. In Table 4.1, $accuracy$ comparison of different object-level trans-

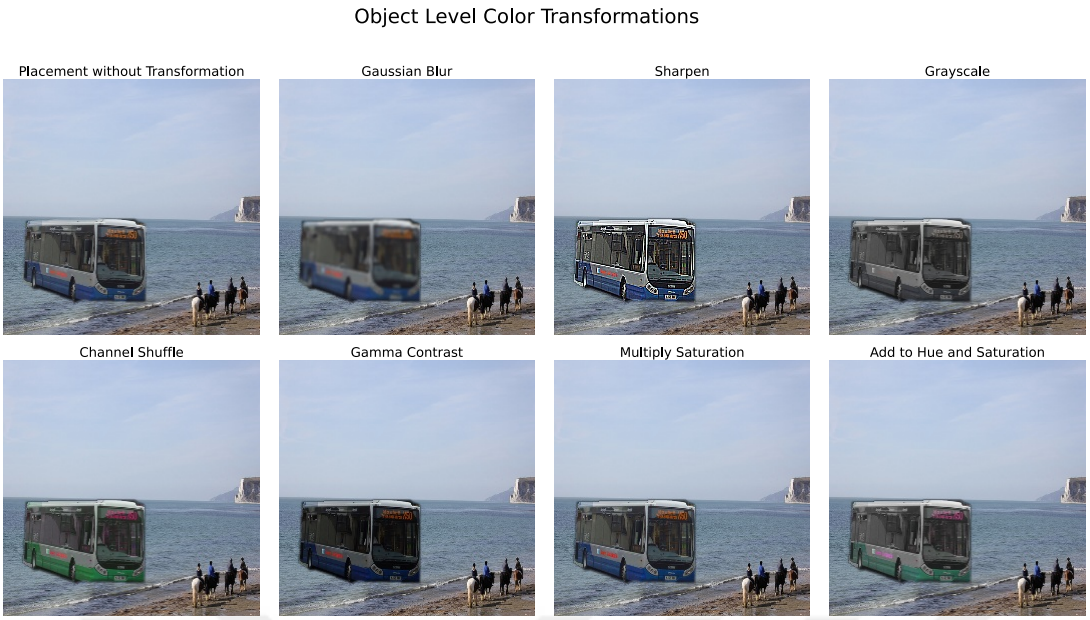

Figure 4.6: Object-level color transformations.

Figure 4.7: Object-level geometric and color transformation combinations.

formation augmentation results on the MS COCO 2017 Validation dataset are presented. Also, all hyperparameters are kept fixed rather than augmentation strategy and probability.

From Table 4.1, it can be clearly inferred that object-level transformation augmentations (both geometric and color) is increasing $accuracy$ on regular data. By combining geometric and color transformation augmentations on top of the object placement augmentations, we have increased $accuracy$ by almost 0.01 points.

4.2.2 Experiments on Combining Object-Level Augmentations

Table 4.2: Accuracy score comparison between different object-level augmentation applications on the dataset MS COCO Validation 2017.

Model	*Obj. Place.*	*Obj. Rem.*	*Obj. Trans.*	*Accuracy*
Baseline	✗	✗	✗	0.6020
Place. only	✓	✗	✗	0.6173
Rem. only	✗	✓	✗	0.6142
Place. + Rem.	✓	✓	✗	0.6196
Place. + Rem. + Trans.	✓	✓	✓	**0.6217**

In this section, results for all object-level augmentations are presented. We combined object removal, object placement, and object-level transformation augmentations. In the previous section, we conducted an ablation study for object removal and placement augmentation probabilities. We observed that $accuracy$ is decreasing between 0.4 and 0.6 augmentation probabilities. Therefore, we also tried 0.5 object removal and placement augmentation probability. By using 0.5 augmentation probabilities, we get an even better $accuracy$ score than 0.4 augmentation probabilities. Moreover, in the experiments for object-level transformation augmentations, we concluded that using 0.5 augmentation probability for geometric and color transformations results in best $accuracy$ scores. Thus, all reported results in this section are obtained using 0.5 object-level augmentation probability with each strategy.

In Table 4.2, results with different object-level augmentation strategies can be found. It is clear that the model's performance can be increased by combining all object-level augmentation with optimum augmentation probabilities. By combining object removal, object placement, and object-level transformations, we achieved to increase $accuracy$ score by almost 0.02 points.

4.3 Experiment 2: Contrastive Learning with Object-Level Augmentations

In this section, we presented our experimental results for the proposed contrastive learning approach. While training our contrastive learning pipeline, we tried to optimize hyper-parameters, like batch size, number of epochs, and learning rate. We observed that the model generally overfits after the seventh epoch.

In Table 4.3 we presented $accuracy$ comparison results with and without contrastive learning. $Accuracy$ values are computed on the MS COCO 2017 Validation dataset. For object-level augmentations, we have chosen the best object-level augmentation probabilities found in previous experiments. We see from the table that contrastive learning without any tuning slightly increases accuracy.

Table 4.3: Accuracy score comparison for contrastive learning on the dataset MS COCO Validation 2017.

Model	*Obj. Level Aug.*	*Contrast. Learning*	*Accuracy*
Baseline	✗	✗	0.6020
Obj. Level Aug.	✓	✗	0.6217
Cont. Learning	✓	✓	**0.6274**

4.4 Experiment 3: Out-of-context Object Recognition

In this section, we evaluate our model on scenes with out-of-context objects. For this, we use the UnRel dataset and our Out-of-Context Object Placements dataset (described in Section 4.1.3). Since $F_{1-macro}$ and $accuracy$ measure results have similar score distribution across our experiments, only $accuracy$ measure comparison results are shared in this section. Moreover, only the best models from previous experiments are used to simplify the analysis. The best models are chosen by looking at $accuracy$ scores on the MS COCO 2017 Validation dataset. In Table 4.4, at each row, the same model is evaluated on different datasets.

In Table 4.4 we observe that all object-level augmentations result in better $accuracy$ scores for out-of-context datasets. We chose best model using regular dataset (MS

Table 4.4: Accuracy score comparison between different augmentations on datasets that are used for evaluation. $COCOVal2017$ is the original MS COCO 2017 dataset, and $OOCPlacement$ is our out-of-context object placement dataset.

Model	*COCO Val 2017*	*UnRel*	*OOC Placement*
Baseline	0.6020	0.5540	0.4305
Placement only	0.6173	0.5621	0.5090
Removal only	0.6142	0.5893	0.3987
Place. + Rem.	0.6196	0.6017	0.5219
Place. + Rem. + Trans.	0.6217	0.6158	0.5305
Cont. Learning	**0.6274**	**0.6244**	**0.5428**

COCO 2017 Validation), to not lose $accuracy$ on ordinary data, while increasing $accuracy$ for out-of-context dataset.

In our experiments, we also observed that applying object-level augmentations more aggressively (giving augmentation probability more than 50%) may increase $accuracy$ on out-of-context scenarios while decreasing $accuracy$ on regular conditions. So, there is an $accuracy$ trade-off between regular and out-of-context scenarios after some point. This means that if you want to gain more success on out-of-context scenarios, you may lose some $accuracy$ on ordinary data.

In Table 4.4, we see that the baseline model has the worst score for the UnRel dataset. Moreover, applying a combination of object-level augmentations achieved better accuracy than using a single object-level augmentation. With the combination of object-level augmentations and contrastive learning, we have marginally increased $accuracy$ by 0.05 points without hyper-parameter tuning.

For the OOC dataset, there is a similar trend with the UnRel dataset. However, this time, applying only object removal augmentation has a negative impact on the accuracy score. Otherwise, we can say that using object-level augmentations and contrastive learning marginally increase $accuracy$ score by 0.11 point.

In Figure 4.8, example prediction results are visualized for the baseline model and the best model with object-level augmentation (the combination of object removal, object

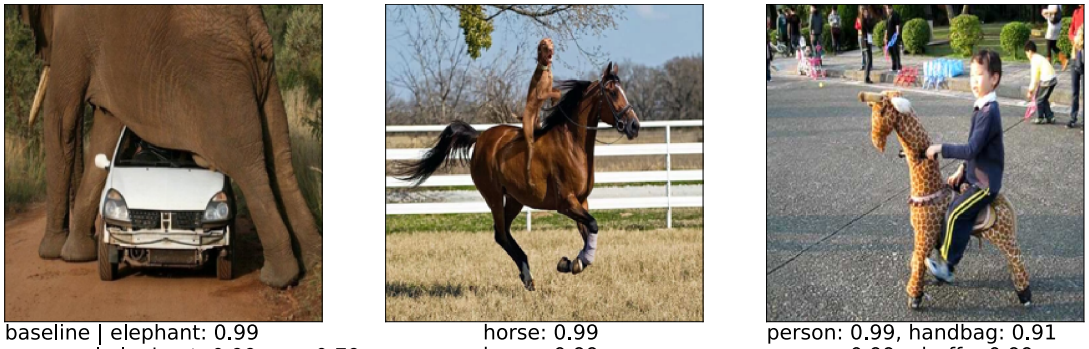

Figure 4.8: Example prediction results with baseline model and object level augmentation applied model. Under each image first line is the baseline model prediction results, and second line is the object level augmentation applied model results.

placement with object-level transformations in Table 4.2). Although it still may fail on some out-of-context scenarios (middle image in Figure 4.8), our proposed object-level augmentations perform considerably better than the baseline on out-of-context scenarios.

4.5 Experiment 4: Prioritizing Certain Classes During Augmentation

Class imbalance is a widespread problem in machine learning and long-tailed class-sample distributions are a nuisance for object recognition and detection problems in Computer Vision. Such a long-tailed distribution is illustrated in Figure 4.9 for the COCO mini-train dataset. Since the number of categories (80) is too much to visualize efficiently, we randomly selected 40 class categories. In Figure 4.9, it is obvious that the "person" class is dominating the dataset, which causes a machine learning method to focus more on learning the "person" class and provide sub-optimal performance on other classes.

Oksuz et al. provided a detailed review of imbalance problems in object detection [31]. They offered a comprehensive taxonomy that defines different types of imbalance problems. According to their definition, we focus on solving foreground-foreground class imbalance problems using object-level augmentations.

We conduct experiments by prioritizing augmentations for certain classes. To be

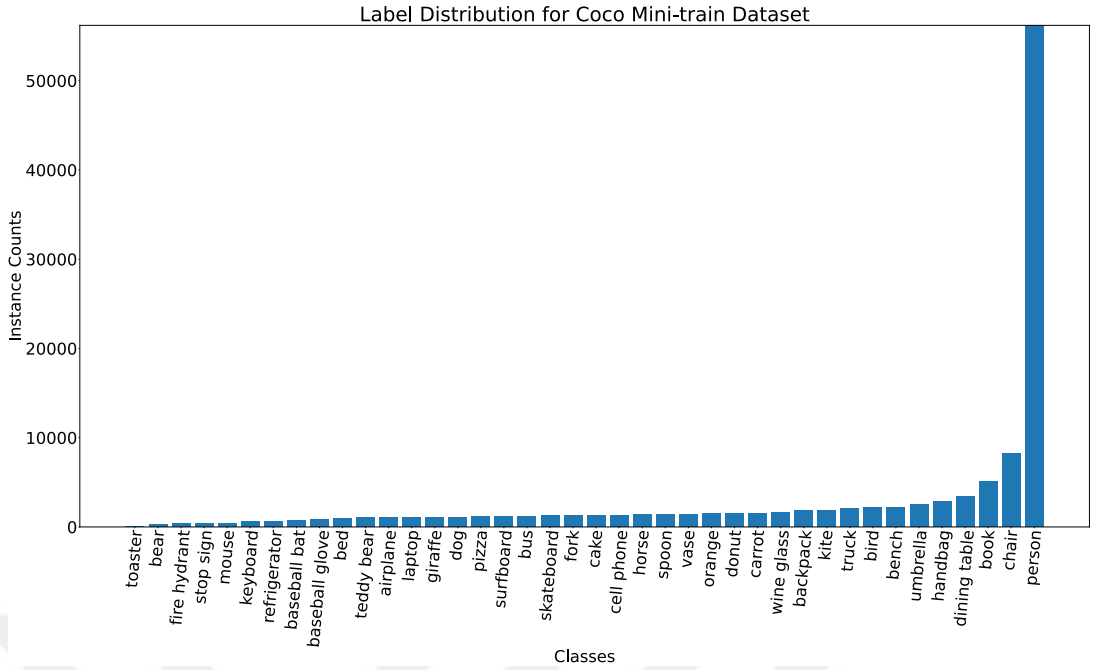

Figure 4.9: The long-tailed distribution of the COCO Mini-train Dataset classes. For the sake of clarity, randomly chosen 40 classes are visualized.

specific, we prioritized the removal of “person” objects and the placement of rare classes during the augmentation phase. 10 rarest classes are chosen from the training dataset: “toaster", “hair dryer", “parking meter", “scissors", “bear", “toothbrush", “hot dog", “stop sign", “microwave" and “fire hydrant". To investigate the effects of prioritizing certain classes during experiments, we trained three different models:

- Prioritizing removal of “person" classes: Object removal is applied with 90% probability for the “person" class while this probability is 50% probability for other classes.
- Prioritizing placement of rare classes: Object placement is applied with 80% for rare classes, whereas it is 50% probability for other classes.
- Combination of both prioritization with object removal and object placement.

In Table 4.5, $accuracy$ scores on the COCO Val 2017 dataset are presented. It can be clearly seen that prioritizing the removal of the “person” class is decreasing $accuracy$ score for the “person” class as expected. However, prioritizing the removal of the

Table 4.5: $Accuracy$ score comparison between prioritizing different classes during augmentation on the COCO Val 2017 dataset.

Model	*All Classes*	*Person*	*Rare Classes*
Baseline	0.6010	**0.8656**	0.4439
Rem. Person	0.6129	0.8414	0.4481
Place. Rare Classes	0.6205	0.8622	0.4663
Rem. Person + Place. Rare Classes	**0.6218**	0.8567	**0.4763**

"person" class can increase the overall $accuracy$ score compared to the baseline model. As expected, prioritizing placement of rare classes increases the $accuracy$ score for rare classes and the overall $accuracy$ score. Finally, with the combination of both augmentations, we marginally increase the overall $accuracy$ score by 0.02 points. Moreover, we gained a 0.032 increase for rare classes.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this work, we have focused on the problem of recognizing out-of-context objects in images. To this end, we have proposed an Object-Level Augmentation workflow and a contrastive learning pipeline that uses the proposed object-level augmentations for out-of-context object recognition. On various datasets, we have shown that, by placing objects with different color and geometric transformations or removing objects, a deep network can be trained to perform better on out-of-context scenes. Moreover, we have created a new dataset with out-of-context objects by applying our object placement augmentation pipeline on an existing dataset with regular scenes.

5.1 Limitations and Future Work

Although we have reported promising results with our object-level augmentations, there are certain limitations. For example, object removal can lead to visually unrealistic results. Especially when a removed object occupies a significant portion of the image, there may not be enough information (context) in the image to complete the region of the removed object. Therefore, the object removal approach may not be feasible for all vision problems.

Since creating object-level augmentations, especially the object removal process, are very time-consuming, we generated augmented samples before the training phase. Then used the same data for all epochs in training. Generally, augmentations are used to increase the number of examples in the training dataset. However, in this work, we used object-level augmentations to increase the variety of data. To extend our approach, we plan to implement a new version of object-level augmentations that can

be used while training deep learning models.

Finally, we plan to use the representations learned by the deep network trained with our augmentation scheme for object detection and instance segmentation problems. We believe that our augmentation scheme has a high potential for different computer vision applications.

REFERENCES

[1] L. Biewald, "Experiment tracking with weights and biases," 2020. Software available from wandb.com.

[2] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *IEEE Conference on Computer VIsion and Pattern Recognition (CVPR)*, 2010.

[3] M. J. Choi, A. Torralba, and A. S. Willsky, "Context models and out-of-context objects," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 853–862, 2012.

[4] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020.

[5] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 13001–13008, 2020.

[6] M. D. Bloice, P. M. Roth, and A. Holzinger, "Patch augmentation: Towards efficient decision boundaries for neural networks," *arXiv preprint arXiv:1911.07922*, 2019.

[7] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2918–2928, 2021.

[8] J. Peyre, I. Laptev, C. Schmid, and J. Sivic, "Weakly-supervised learning of visual relations," in *ICCV*, 2017.

[9] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Computer Vision, IEEE International Conference on*, vol. 2, pp. 273–273, IEEE Computer Society, 2003.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[11] N. Samet, S. Hicsonmez, and E. Akbas, "Houghnet: Integrating near and long-range evidence for bottom-up object detection," in *European Conference on Computer Vision*, pp. 406–423, Springer, 2020.

[12] R. Shetty, B. Schiele, and M. Fritz, "Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8218–8226, 2019.

[13] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 364–380, 2018.

[14] E. Ntavelis, A. Romero, I. Kastanis, L. Van Gool, and R. Timofte, "Sesame: semantic editing of scenes by adding, manipulating or erasing objects," in *European Conference on Computer Vision*, pp. 394–411, Springer, 2020.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[17] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[18] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "Edgeconnect: Generative image inpainting with adversarial edge learning," *arXiv preprint arXiv:1901.00212*, 2019.

[19] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2149–2159, 2022.

[20] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Aggregated contextual transformations for high-resolution image inpainting," *arXiv preprint arXiv:2104.01431*, 2021.

[21] Y. Zeng, Z. Lin, J. Yang, J. Zhang, E. Shechtman, and H. Lu, "High-resolution image inpainting with iterative confidence feedback and guided upsampling," in *European Conference on Computer Vision*, pp. 1–17, Springer, 2020.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[23] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539–546, IEEE, 2005.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[25] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18661–18673, 2020.

[26] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] M. S. Sorower, "A literature survey on algorithms for multi-label learning," *Oregon State University, Corvallis*, vol. 18, pp. 1–25, 2010.

[30] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko,

K. Pfeiffer, B. Cook, I. Fernández, F.-M. De Rainville, C.-H. Weng, A. Ayala-Acevedo, R. Meudec, M. Laporte, *et al.*, "imgaug." `https://github.com/aleju/imgaug`, 2020. Online; accessed 01-Feb-2020.

[31] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3388–3415, 2020.