

# Regularization, Uncertainty Estimation and Out of Distribution Detection in Convolutional Neural Networks

Ujwal Karthik Krothapalli

Dissertation submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Engineering

A. Lynn Abbott, Chair

Pinar Acar

Creed Jones III

Haibo Zeng

Yunhui Zhu

August 13, 2020

Blacksburg, Virginia

Keywords: Regularization, Uncertainty Estimation, Classifier, Convolutional Neural  
Network

Copyright 2020, Ujwal Karthik Krothapalli

# Regularization, Uncertainty Estimation and Out of Distribution Detection in Convolutional Neural Networks

Ujwal Karthik Krothapalli

(ABSTRACT)

Classification is an important task in the field of machine learning and when classifiers are trained on images, a variety of problems can surface during inference. 1) Recent trends of using convolutional neural networks (CNNs) for various machine learning tasks has borne many successes and CNNs are surprisingly expressive in their learning ability due to a large number of parameters and numerous stacked layers in the CNNs. This increased model complexity also increases the risk of overfitting to the training data. Increasing the size of the training data using synthetic or artificial means (data augmentation) helps CNNs learn better by reducing the amount of over-fitting and producing a regularization effect to improve generalization of the learned model. 2) CNNs have proven to be very good classifiers and generally localize objects well; however, the loss functions typically used to train classification CNNs do not penalize inability to localize an object, nor do they take into account an object's relative size in the given image when producing confidence measures. 3) Convolutional neural networks always output in the space of the learnt classes with high confidence while predicting the class of a given image regardless of what the image consists of. For example an ImageNet-1K trained CNN can not say if the given image has no objects that it was trained on if it is provided with an image of a dinosaur (not an ImageNet category) or if the image has the main object cut out of it (context only). We approach these three different problems using bounding box information and learning to produce high entropy predictions on out of distribution classes. To address the first problem, we propose a

novel regularization method called CopyPaste. The idea behind our approach is that images from the same class share similar context and can be ‘mixed’ together without affecting the labels. We use bounding box annotations that are available for a subset of ImageNet images. We consistently outperform the standard baseline and explore the idea of combining our approach with other recent regularization methods as well. We show consistent performance gains on PASCAL VOC07, MS-COCO and ImageNet datasets. For the second problem we employ objectness measures to learn meaningful CNN predictions. Objectness is a measure of likelihood of an object from *any* class being present in a given image. We present a novel approach to object localization that combines the ideas of objectness and label smoothing during training. Unlike previous methods, we compute a smoothing factor that is *adaptive* based on relative object size within an image. We present extensive results using ImageNet and OpenImages to demonstrate that CNNs trained using *adaptive* label smoothing are much less likely to be overconfident in their predictions, as compared to CNNs trained using hard targets. We train CNNs using objectness computed from bounding box annotations that are available for the ImageNet dataset and the OpenImages dataset. We perform extensive experiments with the aim of improving the ability of a classification CNN to learn better localizable features and show object detection performance improvements, calibration and classification performance on standard datasets. We also show qualitative results using class activation maps to illustrate the improvements. Lastly, we extend the second approach to train CNNs with images belonging to out of distribution and context using a uniform distribution of probability over the set of target classes for such images. This is a novel way to use uniform smooth labels as it allows the model to learn better confidence bounds. We sample 1000 classes (mutually exclusive to the 1000 classes in ImageNet-1K) from the larger ImageNet dataset comprising about 22K classes. We compare our approach with standard baselines and provide entropy and confidence plots for in distribution and out of distribution validation sets.

# Regularization, Uncertainty Estimation and Out of Distribution Detection in Convolutional Neural Networks

Ujwal Karthik Krothapalli

(GENERAL AUDIENCE ABSTRACT)

Categorization is an important task in everyday life. Humans can perform the task of classifying objects effortlessly in pictures. Machines can also be trained to classify objects in images. With the tremendous growth in the area of artificial intelligence, machines have surpassed human performance for some tasks. However, there are plenty of challenges for artificial neural networks. Convolutional Neural Networks (CNNs) are a type of artificial neural networks. 1) Sometimes, CNNs simply memorize the samples provided during training and fail to work well with images that are slightly different from the training samples. 2) CNNs have proven to be very good classifiers and generally localize objects well; however, the objective functions typically used to train classification CNNs do not penalize inability to localize an object, nor do they take into account an object's relative size in the given image. 3) Convolutional neural networks always produce an output in the space of the learnt classes with high confidence while predicting the class of a given image regardless of what the image consists of. For example, an ImageNet-1K (a popular dataset) trained CNN can not say if the given image has no objects that it was trained on if it is provided with an image of a dinosaur (not an ImageNet category) or if the image has the main object cut out of it (images with background only). We approach these three different problems using object position information and learning to produce low confidence predictions on out of distribution classes. To address the first problem, we propose a novel regularization method called CopyPaste. The idea behind our approach is that images from the same class share similar context and

can be ‘mixed’ together without affecting the labels. We use bounding box annotations that are available for a subset of ImageNet images. We consistently outperform the standard baseline and explore the idea of combining our approach with other recent regularization methods as well. We show consistent performance gains on PASCAL VOC07, MS-COCO and ImageNet datasets. For the second problem we employ objectness measures to learn meaningful CNN predictions. Objectness is a measure of likelihood of an object from *any* class being present in a given image. We present a novel approach to object localization that combines the ideas of objectness and label smoothing during training. Unlike previous methods, we compute a smoothing factor that is *adaptive* based on relative object size within an image. We present extensive results using ImageNet and OpenImages to demonstrate that CNNs trained using *adaptive* label smoothing are much less likely to be overconfident in their predictions, as compared to CNNs trained using hard targets. We train CNNs using objectness computed from bounding box annotations that are available for the ImageNet dataset and the OpenImages dataset. We perform extensive experiments with the aim of improving the ability of a classification CNN to learn better localizable features and show object detection performance improvements, calibration and classification performance on standard datasets. We also show qualitative results to illustrate the improvements. Lastly, we extend the second approach to train CNNs with images belonging to out of distribution and context using a uniform distribution of probability over the set of target classes for such images. This is a novel way to use uniform smooth labels as it allows the model to learn better confidence bounds. We sample 1000 classes (mutually exclusive to the 1000 classes in ImageNet-1K) from the larger ImageNet dataset comprising about 22K classes. We compare our approach with standard baselines on ‘in distribution’ and ‘out of distribution’ validation sets.

# Dedication

*To my parents.*

# Acknowledgments

I would like to express my gratitude to numerous people for helping me throughout this insightful and long journey and would like to mention a few of them.

I am forever indebted to my advisor, Dr. A. Lynn Abbott. Dr. Abbott welcomed me to his lab in 2013 and has remained a source of guidance throughout the completion of my program. His expertise, insight, and patience have helped me become the researcher that I am today. I am very thankful to my committee members, Dr. Haibo Zeng, Dr. Yunhui Zhu, Dr. Pinar Acar and Dr. Creed Jones III for their insightful comments and questions.

I would like to thank my current and past fellow lab members, especially Xiaolong Li for all the feedback, questions, and comments.

My parents, family and friends have always supported my relentless academic ambitions and encouraged me to pursue my passion in machine learning. I am forever grateful to my wonderful brother Gautam. A very special thank you to my late grandparents. Many thanks to my uncle Dr. Venkat Mummalaneni, Aunt Shobha and Aunt Lakshmi for their constant support and encouragement. I would like to thank my cousins Dr. Simha Mummalaneni and Dr. Vaishnavi Gummadi for setting a very high bar and my cousins Vaibhav and Vikram for their support and advice throughout the years.

I would like to express my gratitude for all the support and proofreading help I received from Boyce Lacy Burnett over the past year.

I was supported throughout my program by VTTI through many projects, and am especially thankful to Andy Petersen for the wonderful opportunities he provided at VTTI. I have spent all my summers since 2013 working on various projects and pushing the compute infrastructure limits at VTTI. I am thankful to Bill F., Brian L., Clark G., Calvin W. and Zeb B. for their advice, friendship and support over the years at VTTI.

# Contents

<b>List of Figures</b>	<b>xii</b>
<b>List of Tables</b>	<b>xxiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	4
1.1.1 Regularization . . . . .	4
1.1.2 Adaptive Label Smoothing . . . . .	7
1.2 Challenges . . . . .	10
1.3 Contributions . . . . .	10
1.4 Outline . . . . .	11
<b>2 Overview</b>	<b>13</b>
2.1 Deep Learning . . . . .	13
2.1.1 Neurons, Perceptrons and Multi-Layer Perceptrons . . . . .	13
2.1.2 Convolutional Neural Networks . . . . .	16
<b>3 CopyPaste</b>	<b>19</b>
3.1 Datasets . . . . .	19

3.2	Approach	20
3.3	Contributions	22
3.4	Related Work	23
3.4.1	Classic Data Augmentation	23
3.4.2	Random Noise	24
3.4.3	Mixed Sample	24
3.4.4	AutoAugment	25
3.5	Object and Context based Data Augmentation	25
3.6	Experiments	27
3.6.1	ImageNet Classification	28
3.6.2	Object Detection using Pretrained Model	30
3.6.3	Qualitative results	32
3.7	Conclusion	33
<b>4</b>	<b>Adaptive Label Smoothing</b>	<b>39</b>
4.1	Related Work	39
4.2	Method	43
4.3	Experiments	44
4.3.1	Datasets	45
4.4	Experimental setup	49

4.4.1	Datasets and splits . . . . .	49
4.4.2	Hardware and software . . . . .	50
4.4.3	Runtimes . . . . .	50
4.4.4	Classification and calibration . . . . .	56
4.4.5	Transfer learning for object detection . . . . .	56
4.4.6	Ablation studies . . . . .	58
4.5	Conclusion . . . . .	58
<b>5</b>	<b>Out of Distribution Detection</b>	<b>61</b>
5.1	Related Work . . . . .	61
5.2	Method . . . . .	62
5.3	Experiments . . . . .	62
5.4	Results . . . . .	77
5.5	Conclusion . . . . .	92
<b>6</b>	<b>Conclusions</b>	<b>93</b>
	<b>Bibliography</b>	<b>96</b>

# List of Figures

1.1	The general idea of a classifier, $x$ is the input to a learned function $f$ which outputs a class label. Future discussions of classifiers in this dissertation will refer to a CNN ( $f$ ) that takes an input color image ( $x$ ) and outputs a class label. . . . .	2
1.2	LeNet-5 [40], a CNN used for high accuracy digit recognition. Yellow colored layers are convolutional layers, red colored layers are max-pooling layers and teal colored layers are fully connected layers. The layers that are not directly connected to input and output (last layer in this figure) are called hidden layers. The output layer is shown in magenta. Figure plotted using the implementation of [31]. . . . .	4
1.4	Random crops of images are often used when training classification CNNs to help mitigate size, position and scale bias (left half of figure). Unfortunately, some of these crops miss the object as they do not have any object location information. Traditional hard label and smooth label approaches do not account for the proportion of the object being classified and use a fixed label of ‘1’ or ‘0.9’ in the case of label smoothing. Our approach (right half) smooths the hard labels by taking into account the objectness measure to compute an <i>adaptive</i> smoothing factor. The objectness is computed using bounding box information as shown above. Our approach helps generate accurate labels during training and penalizes low-entropy (high-confidence) predictions for context-only images (the main object is completely or mostly absent). . . .	7

1.3	A typical flow process of training a classifier is shown. The fourth step represents the theme of the work being pursued. . . . .	12
2.1	A simple neuron. . . . .	14
2.2	A simple perceptron. . . . .	15
2.3	LeNet-5, a CNN used for high accuracy digit recognition [39]. . . . .	17
3.1	In a classification approach, the location of the object is not accounted for while training and the expected outputs (post training) for all the 3 variations of the image should indicate Dog. The CNN has to learn to localize the pertinent object (dog in this case) and produce an output indicating the presence of the object. . . . .	20
3.2	Our method uses bounding box information to paste objects belonging to the same class in a given image. The red bounding box is an example of bounding box annotation and the rest of the image is considered context. The green bounding box shows the object that has been pasted from another image of the same class if bounding boxes are available. The bottom two rows are sample images generated on the fly by our approach. The labels of images in the second row (left to right) are, ‘Goldfish’, ‘Cauldron’, and ‘Alligator lizard’. The labels of images in the third row (left to right) are, ‘Tench’, ‘Snow leopard’, and ‘Robin’. . . . .	21

3.3	We show the differences in recent augmentation methods and the corresponding labels for images generated using the different approaches. Because CutMix and RICAP have no localization information, they are more likely to assign the wrong label to a given crop (red bordered images indicate wrong labels and green bordered images indicate the correct labels), whereas our approach generates correct labels all the time. ImageNet column shows the standard images from the dataset without any augmentation so the labels are not changed during training like the other methods. (Note: The border colors are purely for illustrative purposes. The CNN does NOT receive the samples with the colored borders.) . . . . .	34
3.4	We use a compute node that is different from our main GPU node to handle the tremendous compute imposed by modifying the train set on the fly. The gaps in between the high CPU use show the times during which we utilize clean ImageNet samples. . . . .	35
3.5	We plot the validation error during the training of the baseline approach as well as our approach . . . . .	35
3.6	We plot the last lowest error (last good performance) during the course of training the baseline model as well as our approach . . . . .	36
3.7	Qualitative results using class activation maps to show the most pertinent regions used by each method to make the prediction. . . . .	36
3.8	More qualitative results using class activation maps to show the most pertinent regions used by each method to make the prediction. . . . .	37

3.9	Qualitative results using class activation maps to show the most pertinent regions used by each method to make the prediction. Our approach also helps CutMix and RICAP localize the pertinent objects better. . . . .	38
4.1	Hard-label and label-smoothing based approaches (top half of the figure) do not take into account the proportion of the object being classified. Our approach (bottom half) weights soft labels using the objectness measure to compute an <i>adaptive</i> smoothing factor. . . . .	42
4.2	Examples of class activation maps (CAMs). These were obtained using the implementation of [6]. Two columns on the left show results for baseline CNNs using hard labels and standard label smoothing. Our approach, <i>adaptive</i> label smoothing (“adaptive l.s.”), is illustrated in the three columns on the right. Our technique produces high-entropy predictions and shows an improved localization performance. The values under each CAM represent the top three probabilities, with green indicating the pertinent class and red indicating an incorrect prediction. . . . .	46
4.3	Examples of class activation maps (CAMs). These were obtained using the implementation of [6]. The second and third columns from the left show results for baseline CNNs using hard labels and standard label smoothing. Our approach, <i>adaptive</i> label smoothing (‘Adaptive l.s’), is illustrated in the three rightmost columns. Our technique produces high-entropy predictions and shows an improved localization performance. The values under each CAM represent the top three probabilities, with green indicating the pertinent class and red indicating an incorrect prediction. . . . .	47

4.4 Examples of class activation maps (CAMs). These were obtained using the implementation of [6]. The second and third columns from the left show results for baseline CNNs using hard labels and standard label smoothing. Our approach, *adaptive* label smoothing (‘Adaptive l.s’), is illustrated in the three rightmost columns. Our technique produces high-entropy predictions and shows an improved localization performance. The values under each CAM represent the top three probabilities, with green indicating the pertinent class and red indicating an incorrect prediction. . . . . 48

4.5 The first row of images in the left half of the figure are an example of the ImageNet dataset (N=0.474M) that have bounding box annotations. We match the images from the training set of ImageNet-1K dataset with the corresponding ‘.xml’ files included in the ImageNet object detection dataset. We then create object masks for each of the images. When applying any scaling and cropping operation to training samples, we apply the same transformation to the corresponding object masks as well. By counting the number of white pixels, we can determine the object proportion post transformation. We describe the two other approaches in the figure, the ‘mask’ version of our approach has a single object (for images with multiple bounding box annotations) and this version has 0.528M samples. Our approach helps generate accurate labels during training and penalizes low-entropy (high-confidence) predictions for context-only images like the example on the right half of the figure. . . . . 51

4.6	Top half of the figure shows the count per class for the ImageNet dataset, the highest number of images in a given class is ‘1349’ and the lowest count is ‘190’. The distribution in this case is not as skewed as the OpenImages (bottom half) dataset. About 60 classes in our subset of the OpenImages dataset account for half the dataset. The maximum and minimum counts are 55K and 28K respectively. . . . .	53
4.7	Reliability diagrams help understand the calibration performance [12, 54] of classifiers. We compute $ECE_1$ using the implementation of [71] on the validation set of ImageNet. The deviation from the dashed line (shown in gray), weighted by the histogram of confidence values, is equal to Expected Calibration Error [71]. The top half of the figure shows classifiers trained using the same dataset ( $N=0.528M$ ), but with different values of $\beta$ . The leftmost reliability diagram is the classic hard label setting and the rightmost reliability diagram is the <i>adaptive</i> label setting. The bottom half of the figure compares classifiers trained on the complete ImageNet (leftmost) with 3 classifiers trained on the subset of ImageNet with bounding box labels using different values of the $\alpha$ hyperparameter. . . . .	57
5.1	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for the hard label case. Clearly, there is no distinct way to threshold the different distributions as they severely overlap with one another.	63

5.2	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for the standard uniform label smoothing case. The separation is better than the hard label case, but there is plenty of overlap. . . . .	64
5.3	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method. The separation is better than the hard label and label smoothing cases, as we produce low confidence scores for a much larger number of out of distribution samples. . . . .	65
5.4	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ . . . . .	66
5.5	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.25$ . . . . .	67
5.6	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with 5 percent of training samples are context only and 5 percent of samples are from 500 classes of OImageNet training set. . . . .	68

5.7	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with 5 percent of training samples are context only and 5 percent of samples are from 1000 classes of OImageNet training set. . . . .	69
5.8	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for the hard label case. Clearly, there is no distinct way to threshold the different distributions as they severely overlap with one another. . . . .	70
5.9	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for the standard uniform label smoothing case. The separation is better than the hard label case, but there is plenty of overlap. . . . .	71
5.10	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method. The separation is better than the hard label and label smoothing cases, as we produce low Entropy scores for a much larger number of out of distribution samples. . . . .	72
5.11	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ . . . . .	73
5.12	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.25$ . . . . .	74

5.13	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with 5 percent of training samples are context only and 5 percent of samples are from 500 classes of OImageNet training set. . . . .	75
5.14	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with 5 percent of training samples are context only and 5 percent of samples are from 1000 classes of OImageNet training set. . . . .	76
5.15	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 0 percent of samples are from 1000 classes of OImageNet training set. . . . .	78
5.16	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 0 percent of samples are from 1000 classes of OImageNet training set. . . . .	79
5.17	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 5 percent of samples are from 1000 classes of OImageNet training set. . . . .	80

5.18 Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 5 percent of samples are from 1000 classes of OImageNet training set. . . . .	81
5.19 Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 25 percent of samples are from 1000 classes of OImageNet training set. . . . .	82
5.20 Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 25 percent of samples are from 1000 classes of OImageNet training set. . . . .	83
5.21 Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 5 percent of samples are from 500 classes of OImageNet training set. . . . .	84
5.22 Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 5 percent of samples are from 500 classes of OImageNet training set. . . . .	85

5.23	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 25 percent of samples are from 500 classes of OImageNet training set. . . . .	86
5.24	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 25 percent of samples are from 500 classes of OImageNet training set. . . . .	87
5.25	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 50 percent of samples are from 500 classes of OImageNet training set. . . . .	88
5.26	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 50 percent of samples are from 500 classes of OImageNet training set. . . . .	89

5.27	Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 75 percent of samples are from 500 classes of OImageNet training set. . . . .	90
5.28	Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our <i>adaptive</i> label smoothing method with $\beta = 0.75$ , 5 percent of training samples are context only and 75 percent of samples are from 500 classes of OImageNet training set. . . . .	91

# List of Tables

3.1	ImageNet classification accuracies using ResNet-50 architecture. ‘*’ denotes results reported in their paper. . . . .	28
3.2	ImageNet cross entropies and gaps using ResNet-50 architecture. . . . .	29
3.3	Object detection mean average precision values using ResNet-50 backbone and Faster-RCNN. We report the best out of last 3 epochs for all methods.	31
3.4	Object detection mean average precision values using ResNet-50 backbone and Faster-RCNN. We report the best out of last 4 epochs for all methods.	31
3.5	Object detection mean average precision values using ResNet-50 backbone and Faster-RCNN. We report the best out of last 2 epochs for all methods.	31
3.6	Object detection mean average precision values using ResNet-50 backbone and Faster-RCNN for small objects in COCO. We report the best out of last 2 epochs for all methods. . . . .	32
4.1	Confidence and accuracy metrics on the validation set of ImageNet with all the objects removed using bounding box annotation provided by [8]. . . . .	44
4.2	Classification and calibration results with ImageNet. For a detailed explanation of the metrics please refer to the ‘Experimental setup’ section. ‘A.conf’, ‘O.conf’ and ‘U.conf’ refer to average confidence, overconfidence, and underconfidence scores. We provide ECE values for 100 bins and 15 bins mean scores along with their standard deviation (std). . . . .	52

4.3	Classification and calibration results with OpenImages. For a detailed explanation of the metrics please refer to the ‘Experimental setup’ section. ‘A.conf’, ‘O.conf’ and ‘U.conf’ refer to average confidence, overconfidence, and underconfidence scores. We provide ECE values for 100 bins and 15 bins mean scores along with their standard deviation (std). . . . .	54
4.4	Fine-tuning on MS-COCO using FRCNN for object detection. For a detailed explanation of the results please refer to the ‘Experimental setup’ section. AP refers to average precision and AR refers to average recall at the specified Intersection over union (IoU) level. We also provide AP values for small, medium, and large objects using ‘S’, ‘M’, and ‘L’ respectively. . . . .	55

# Chapter 1

## Introduction

Categories are ubiquitous in almost all datasets. Classification can be defined as the problem of recognizing the category of a given observation. Classification is a well studied statistical problem with a significant amount of progress made in the past decade. It is the basic building block of many applications pertaining to the areas of Computer Vision, Speech Recognition and Natural Language Processing. To train a good classifier, usually a large number of diverse (independent and identically distributed) samples are needed and it is important to prevent the classifier from overfitting to the training samples.

The task of classifying/identifying objects in images is very easy for humans with good vision capabilities however, machines have a difficult problem of relying on millions of pixel values to produce a probability distribution for the same classification task. The human visual cortex is very advanced compared to the state of the art artificial neural networks. The latest advances in the field of deep neural networks (DNNs) show that surpassing human capability when it comes to certain computer vision tasks is possible. Deep convolutional neural networks (CNNs) have been used for solving various computer vision problems, especially image classification [40] with tremendous success using large datasets since 2013 [32, 60]. By pairing the hierarchical approach of DNNs with cross-correlation operations, CNNs can learn complex representations required for classification. Overfitting in this context refers to the phenomenon of simple memorization of the input samples by the CNN (model) rather than reasoning about the salient features of the samples so the classifier may not generalize to

unseen samples. With ever increasing size of neural networks [25, 61], there is a need for vast amount of labeled data [47] and better generalization, as simply increasing the number of parameters in a neural network will often lead to overfitting of the training data. Model capacity is defined as the ability of a model to learn complex tasks, and a larger model usually will be able to approximate and learn to represent more complex data distributions. Training data and an objective function are some of the first design choices that one must consider when training a classifier. Modern day CNNs have a large number of parameters and often suffer from many problems when deployed in the real world. Lack of diverse training data is one of the problems. A popular class of methods used to improve the diversity of training data is known as data augmentation.

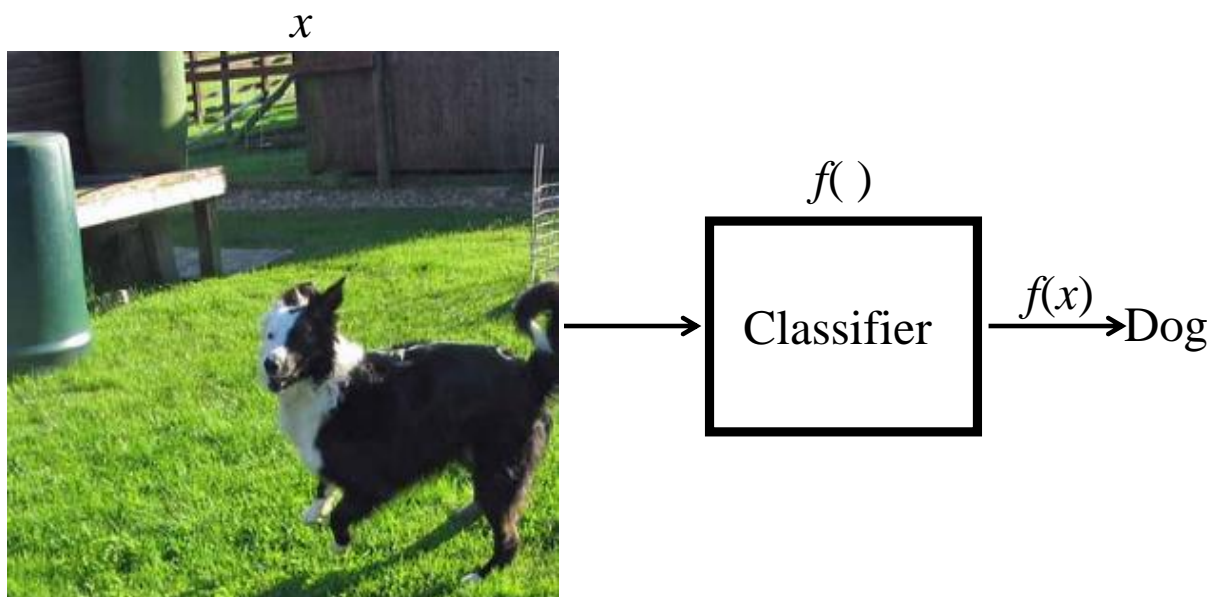


Figure 1.1: The general idea of a classifier,  $x$  is the input to a learned function  $f$  which outputs a class label. Future discussions of classifiers in this dissertation will refer to a CNN ( $f$ ) that takes an input color image ( $x$ ) and outputs a class label.

As shown in figure 1.1, a classifier can be understood as a function  $f$  that maps a given input to a class label. To learn  $f$ , the classifier is supplied with a finite number of examples from

the training dataset. The learnt parameters of a classifier are often referred to as weights. In the case of CNNs, an input layer is the part of a CNN that takes the raw input and outputs a representation of the input. In the case of images, the input layer usually consists of simple image operations like edge detectors that have been learnt.

The output layer of a classification CNN is usually a vector whose length is equal to the number of classes present in the training data. Any layer that is sandwiched between an input layer and an output layer of a neural network is referred to as a hidden layer, as shown in figure 1.2. To solve complex problems, a higher number of hidden layers are used as this allows the neural network to approximate complex distributions of data. The ability of a neural network to learn nuanced differences in the data is referred to as the network's expressive power. The higher the number of hidden layers the higher the expressive power of the classifier. In the case of a simple square RGB image with each side equal to 224 pixels and each pixel having an intensity of 0 to 255, the number of possible images is extremely large,  $256^{3 \times 224 \times 224}$  to be precise. The classification CNNs in this dissertation are trained to detect 1000 different classes of images, as the popular ImageNet 1-K [60] dataset consists of 1000 classes and is often used by researchers to benchmark new approaches for classifying data. So in this case, the CNN is taking raw input from a very large space and learning to represent the input in a much smaller space, 1000 in our case. We also have trained CNNs on the popular OpenImages dataset, and we identify these experiments separately.

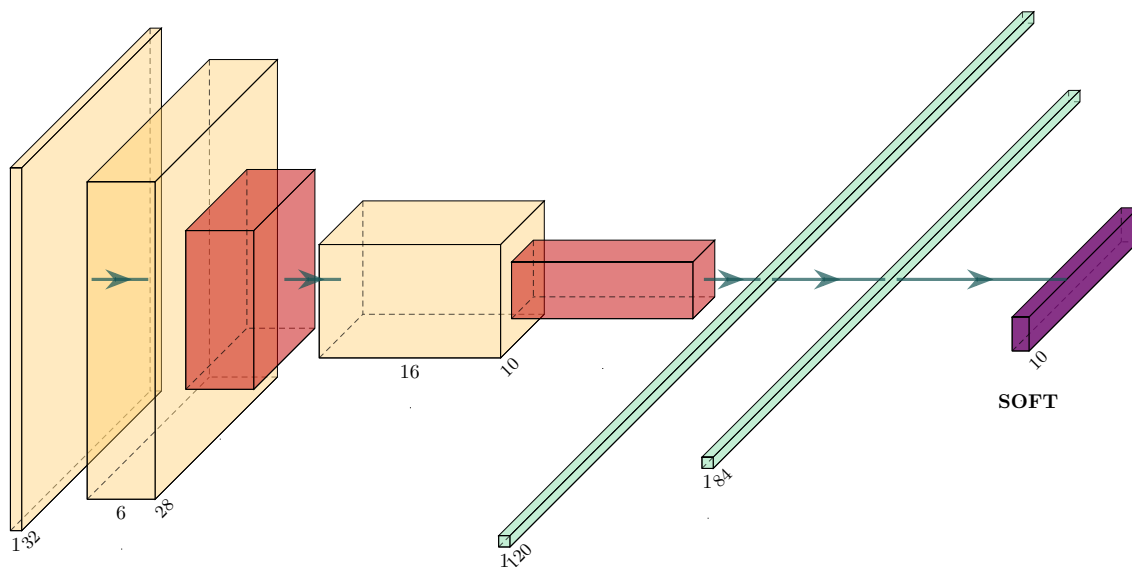


Figure 1.2: LeNet-5 [40], a CNN used for high accuracy digit recognition. Yellow colored layers are convolutional layers, red colored layers are max-pooling layers and teal colored layers are fully connected layers. The layers that are not directly connected to input and output (last layer in this figure) are called hidden layers. The output layer is shown in magenta. Figure plotted using the implementation of [31].

## 1.1 Background

### 1.1.1 Regularization

Training large CNNs has been possible with the tremendous compute power available nowadays. However, a large model requires a large training dataset and collecting large datasets is expensive computationally, manually and logistically. In order to generalize well, a classifier should perform as accurately on unseen examples as it did on seen examples during training. Regularization of a classifier refers to the idea of improving the classifier's gen-

eralization performance when a finite number of training samples are present. Since 2013, numerous data augmentation and regularization methods have been proposed and are a part of standard training procedure for CNNs [28, 32, 63, 67]. In order to prevent simple memorization of the training data (over-fitting) and to constrain the complexity of the classifier, strong regularization is needed. Figure 1.3 shows a flowchart of the steps typically followed in the process of training a classifier. Data augmentation has proven to improve classifier performance at no additional computational cost at test time and is a very active area of research. Categorical labels are used for training CNNs. These labels indicate which class (category) of object is present in a given image. These labels do not have any information on the location of the object in a given image.

*The goal of this work is to show that regularization of existing classifiers can also be done using labels that are not only categorical but also with labels that localize the relevant object in a given image. We explore the idea of using the labels that localize the object to train better classifiers and object detectors as this is a very active area of research [32, 60, 69, 74].*

Early stopping of the training scheme to prevent over-fitting, dropout (dropping hidden layer activations randomly while training) and weight decay (penalizing the norm of the model's parameters) fall under the umbrella of explicit regularization methods, whereas image operations such as randomly cropping a subregion, randomly changing the aspect ratio and randomly adding a positive or negative value to the pixel intensities of an image fall under the umbrella of data augmentation methods. Some of the data augmentation methods can be applied in the feature (hidden) space and these methods are then closer to explicit regularization methods (for example, [63]).

Dropout [63], when applied to the input layer of a CNN, is a type of data augmentation commonly known as random noise data augmentation. The neural activations of the hidden layers of the CNNs are easily confused by texture as opposed to the shape of the object [19].

However, when some of the new data augmentation methods are used, the classifier (model) learns shape specific features of the objects rather than focusing on just their texture. This is one of the several motivating factors for our work.

Randomly dropping hidden layer activations as proposed in dropout [63] and randomly dropping part of input pixels as [13, 20, 62, 79] have improved the ability of CNNs to localize and classify objects in a given image. All the above mentioned data augmentation methods have a regularization effect and most of the methods can be used in conjunction with one another after empirical validation, as the theory behind such methods is not fully understood.

Our regularization method is called CopyPaste and is inspired by the state of the art approaches described above.

### 1.1.2 Adaptive Label Smoothing

Examples of random crops and labels generated by adaptive label smoothing during training

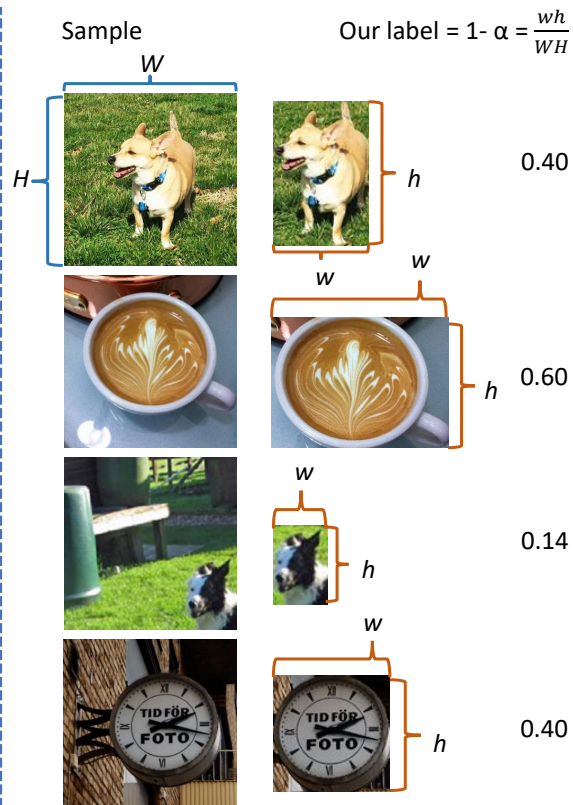
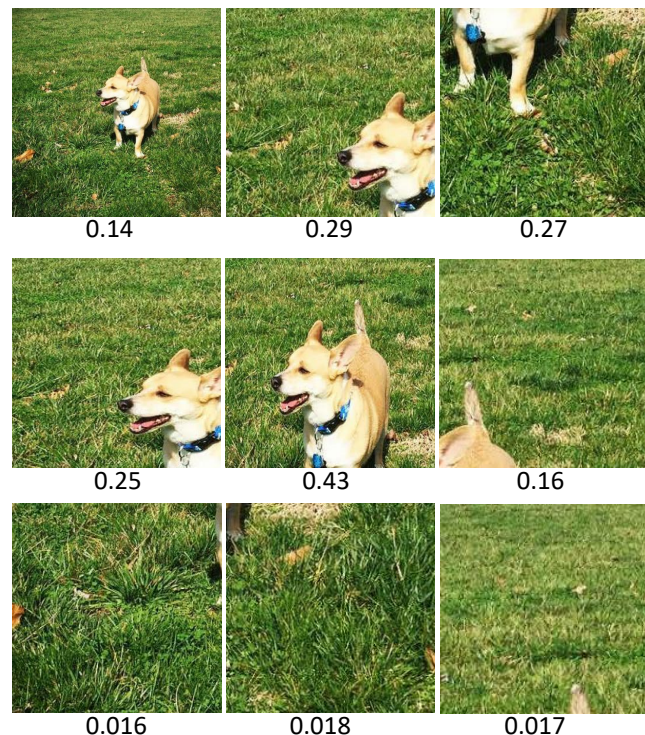


Figure 1.4: Random crops of images are often used when training classification CNNs to help mitigate size, position and scale bias (left half of figure). Unfortunately, some of these crops miss the object as they do not have any object location information. Traditional hard label and smooth label approaches do not account for the proportion of the object being classified and use a fixed label of ‘1’ or ‘0.9’ in the case of label smoothing. Our approach (right half) smooths the hard labels by taking into account the objectness measure to compute an *adaptive* smoothing factor. The objectness is computed using bounding box information as shown above. Our approach helps generate accurate labels during training and penalizes low-entropy (high-confidence) predictions for context-only images (the main object is completely or mostly absent).

We introduce the concept of adaptive label smoothing in this dissertation. Modern CNNs are overconfident in their predictions [27, 37]. Overconfidence in this context refers to the problem where a classifier predicts the wrong label with a high confidence. CNNs also suffer from reliability issues as they are miscalibrated to begin with [23], and miscalibration refers

to the gap in accuracy and confidence of a classifier. There is a growing demand for labeled data [47] to improve generalization performance, as increasing the number of parameters in a neural network [64, 76] will often lead to overfitting of training data, and obtaining an exponentially large labeled dataset is very expensive. Safely deploying deep learning based models has become an immediate challenge [2] as CNNs are being used in a variety of applications. As a community, apart from obtaining high accuracies, we also need to provide reliable uncertainty measures of CNNs. By having reliable confidence measures, we can improve precision (the number of correct predictions divided by all the predictions pertaining to a class) by not acting with certainty when uncertain predictions are produced, as in the case of safety-critical systems.

Regularization is key in improving generalization and minimizing overfitting characteristics of CNNs. Any change to the learning algorithm of a classifier that improves the generalization performance but does not reduce the classifier’s training error can be referred to as regularization in this context. In the case of classification CNNs, ground-truth labels are typically provided as a one-hot representation of class probabilities. These labels consist of 0s and 1s and if the number of classes are  $K$ , the label vector has a single 1 indicating the pertinent class in a given label vector of length  $K$  with  $K - 1$  0s. Recently [66] employed label smoothing, providing soft labels that are a weighted average of the hard targets and the uniform distribution over labels during training to improve learning speed and generalization performance. Usually, in the case of label smoothing, the index corresponding to the relevant label of a given training image has a value of 0.9 and all the other indices in the vector have a value of  $0.1/(K - 1)$ . Soft targets improve the training signal by not providing hard targets to compute the cross entropy loss but a weighted average with a uniform distribution over all classes using a fixed smoothing factor [50, 66]. Label smoothing minimizes the gap between the logits (unnormalized log probabilities output by CNNs) of the classes and shows

improvement in learning speed and generalization; in contrast, hard targets tend to increase the values of the logits and produce overconfident predictions [50, 66]. We illustrate the different labels used by CNNs in figure 1.4.

Object detection [21] is a well-studied problem and most approaches need bounding box information during training. Recently, [16] proposed using novel synthetic images to improve the object detection performance by augmenting training data using object location information. However, classification CNNs have not exploited bounding box information to regularize CNNs on large datasets to our knowledge. The concept of ‘Objectness’ was first introduced by [1], and the role of objectness has been studied extensively since then. Quantifying the likelihood an image window contains an object belonging to any class makes the measure class agnostic.

When training a classifier, cross entropy loss is employed but it does not penalize incorrect spatial attention, making CNNs often overfit to context or texture rather than the pertinent object [19], as shown in the left half of figure 1.4. The bottom row displays samples with negligible amounts of ‘Dog’ pixels and traditional methods would label them as ‘Dog’, causing CNNs to output incorrect predictions with a high confidence when presented with images of backgrounds or just context. Adaptive label smoothing (our approach) involves using bounding box information to smooth the hard labels of a classifier, as displayed to the right in figure 1.4. Traditional approaches [32, 60, 69, 74] use random resize and random crop augmentation, and sometimes lose the pertinent object in the training sample. Our approach adapts label smoothing by deriving the smoothing factor using the objectness measure. When compared to approaches based on hard labels, sample mixing and label smoothing, our approach improves object detection and calibration performance.

## 1.2 Challenges

There are many challenges that plague modern day CNNs. This dissertation brings to light the following problems.

- 1) Augmentation methods used for images used to train classification CNNs do not distinguish object pixels from context pixels. This inattention (to object pixels) introduces context dependence (prediction the class of an image using context rather than object).
- 2) Reliability: CNNs are overconfident and fail to provide confidence measures that are reliable.
- 3) Out of distribution detection: CNNs produce highly confident *wrong* predictions even on images belonging to classes that were not used during training. This is problematic in the real world where many novel classes exist.

## 1.3 Contributions

We develop novel and intuitive solutions to the above challenges. The major contributions are as follows:

- 1) A novel way to use bounding box annotations for data augmentation (training images) with complete control over the augmented object location and scale. By exploiting the bounding box information, objects belonging to the same classes can be placed at random locations on a given image. These objects can be augmented independently using random augmentations before being placed on a training sample to improve the diversity of training data. Results show that our approach improves the generalization performance of CNNs.

- 2) A classifier whose confidence is grounded in object size. We develop a novel way to force a CNN to produce confidence values that correspond to the relative object proportion in a given image. We show that this approach improves localization performance and produces more reliable predictions.
- 3) Out of object detection by generating high entropy/low confidence predictions on unseen data. Using this approach we can neglect predictions whose confidence is below a certain threshold, thereby improving precision.

## 1.4 Outline

This dissertation is organized as follows.

- 1) Chapter 1: Provides an introduction to this dissertation.
- 2) Chapter 2: Provides an overview and history of CNNs.
- 3) Chapter 3: Presents our regularization approach CopyPaste, that leverages object location information during training.
- 4) Chapter 4: Presents our work, Adaptive Label Smoothing.
- 5) Chapter 5: Presents our work on Out of Distribution Detection.
- 6) Chapter 6: We summarize our work and describe exciting new directions for future work.

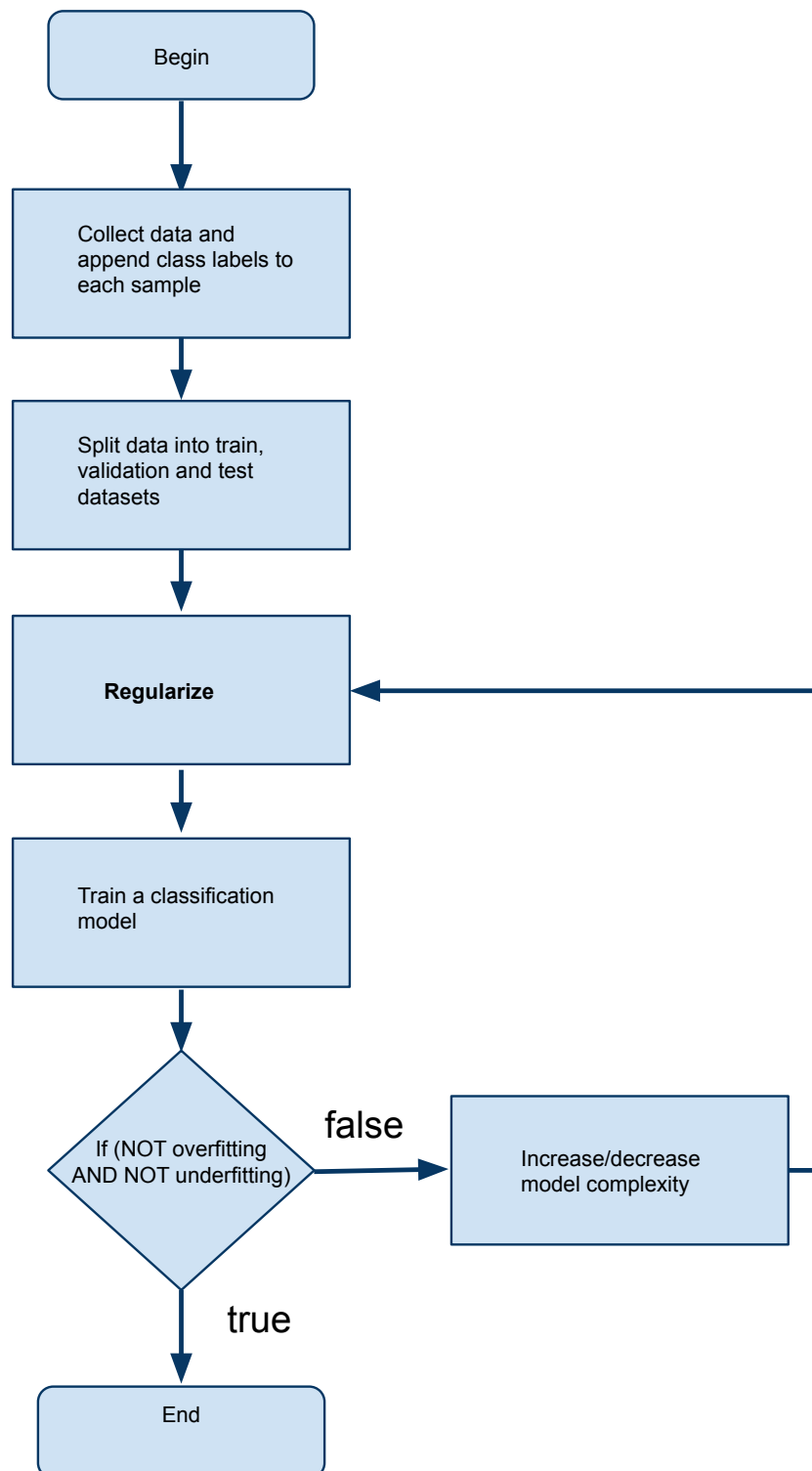


Figure 1.3: A typical flow process of training a classifier is shown. The fourth step represents the theme of the work being pursued.

# Chapter 2

## Overview

This chapter of dissertation will introduce the history and theory behind deep learning in a supervised setting and provide motivation for this work. A detailed description of related work is also provided.

### 2.1 Deep Learning

The subsections will describe the history behind modern Convolutional Neural Networks (CNNs) beginning with the idea of a Neuron and then Multi-Layer Perceptrons (MLPs).

#### 2.1.1 Neurons, Perceptrons and Multi-Layer Perceptrons

The idea of an artificial neuron can be traced to 1943 when Warren McCulloch and Walter Pitts proposed the McCulloch–Pitts neuron. The authors presented a mathematical basis to model an artificial neuron with inputs and a binary output. The McCulloch–Pitts neuron was modeled to accept multiple binary inputs  $x_i$  which were multiplied by the weights  $w_i$  (represented between -1 and +1). The values  $w_i x_i$  were then summed to compute a weighted sum  $S$ . The neuron is said to have fired when the weighted sum  $S$  was greater than a threshold  $T$  [48]. The threshold  $T$  is also called an activation function. The neuron was able to perform logical operations like AND, OR and NOT by using appropriate thresholds

and inputs. The activation function of the McCulloch–Pitts neuron is a step function or a Heaviside function. Mathematically, the weighted sum  $S$  is computed using the equation:

$$S = \sum_{i=1}^n w_i x_i \quad (2.1)$$

The output  $y$  of the McCulloch–Pitts neuron is a binary variable. The output can be 0 or 1 and is computed using the equation:

$$y = f(s) = \begin{cases} 1 & \text{for } s \geq T \\ 0 & \text{for } s < T \end{cases} \quad (2.2)$$

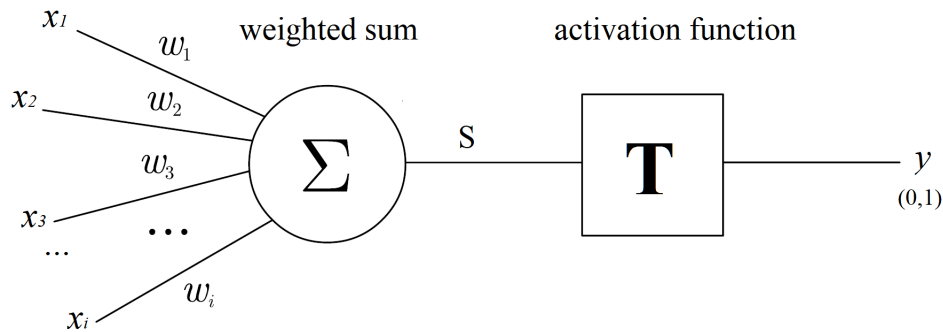


Figure 2.1: A simple neuron.

The neuron was improved to a linear classifier called the Perceptron by Frank Rosenblatt in 1958 [59]. The perceptron was essentially a binary classifier. The perceptron added another parameter to the McCulloch-Pitts neuron called bias. The bias is a constant weight in the case of the perceptron and can take a value of -1 or +1. The output of the perceptron could be +1 or -1. The weights of a perceptron can be ‘learnt’ using a technique developed by Donald Hebb called Hebbian Learning [26]. Adding the bias term to 2.1 changes the

mathematical representation to:

$$S = \sum_{i=1}^n w_i x_i + b \quad (2.3)$$

The weight updates can be represented as:

$$\Delta w_{ij} = \eta y_j x_i \quad (2.4)$$

Learning the perceptron is made possible by using the  $w_{ij}$  which is referred to the weight update. The weight update is equal to the product of learning rate of the perceptron  $\eta$ , the inputs  $x_i$  and the output  $y_j$ .

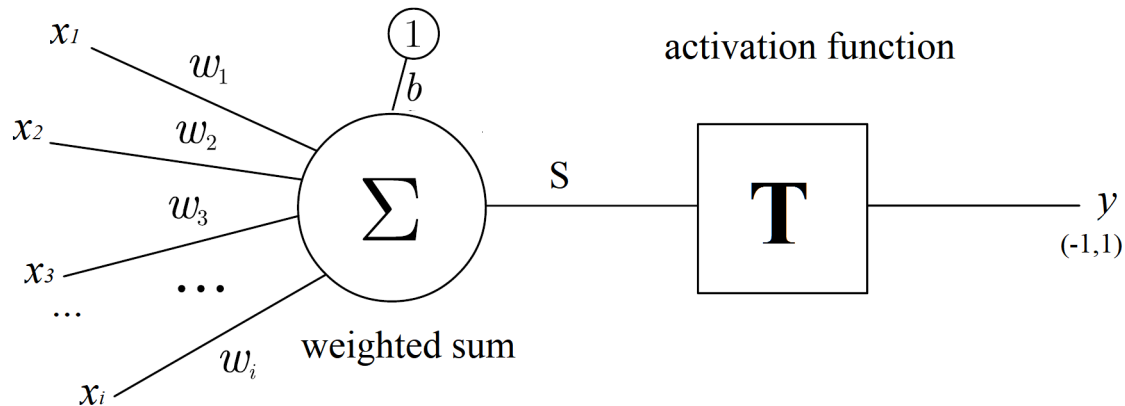


Figure 2.2: A simple perceptron.

Limitation of the perceptron was the input data had to be linearly separable. Apart from simple logic gates, non-linearly separable cases like XOR could not be solved using the perceptron model.

In 1989, George Cybenko published the idea of using a combination of linear perceptrons to approximate continuous functions [11]. These networks are also called Feed Forward Neural Networks (FFNNs) as the inputs are fed in the forward direction to layers of perceptrons.

The architecture of FFNNs is simple, a layer of perceptrons feeds the next layer and this process connects the input layer with that of the output with multiple hidden layers. To train FFNNs, backpropagation is employed, this method takes advantage of the chain-rule to propagate the updates. The weights of the network are updated based on the error measured at the output layer and propagated all the way back to the input layer in a layer by layer fashion. When backpropagation is performed in an iterative fashion over all the samples in a training set, the process is called gradient descent.

### 2.1.2 Convolutional Neural Networks

The Multi-Layer Perceptrons (MLPs) were extended to work with images. In 1980 Kunihiko Fukushima proposed the Neocognitron. By using a combination of simple and complex cells the Neocognitron was able to provide translational invariance for recognition tasks [18].

The term Convolutional Neural Network (CNN) was popularized by [LeCun et al.](#) in 1990. It was used to classify handwritten digits with an accuracy of 99% [41]. The model was called ‘LeNet’, the network was able to classify handwritten digits with variance in position, scale and appearance.

CNNs evolved with the advent of parallel computing resources like GPUs and the availability of labeled data. In 2012 a CNN called “AlexNet” by [Krizhevsky et al.](#) formally brought the methods to a wide audience by competing in the 2012 “ImageNet” challenge and surpassing all classic computer vision baselines. The AlexNet enabled modern CNNs which are being used every day by consumers all over the world.

Convolutional neural networks have a computational advantage over FFNNs that are fully connected, the number of parameters are minimized in a CNN because of weight sharing of neurons. Spatial and shift invariances are a result of the shared parameters and pooling

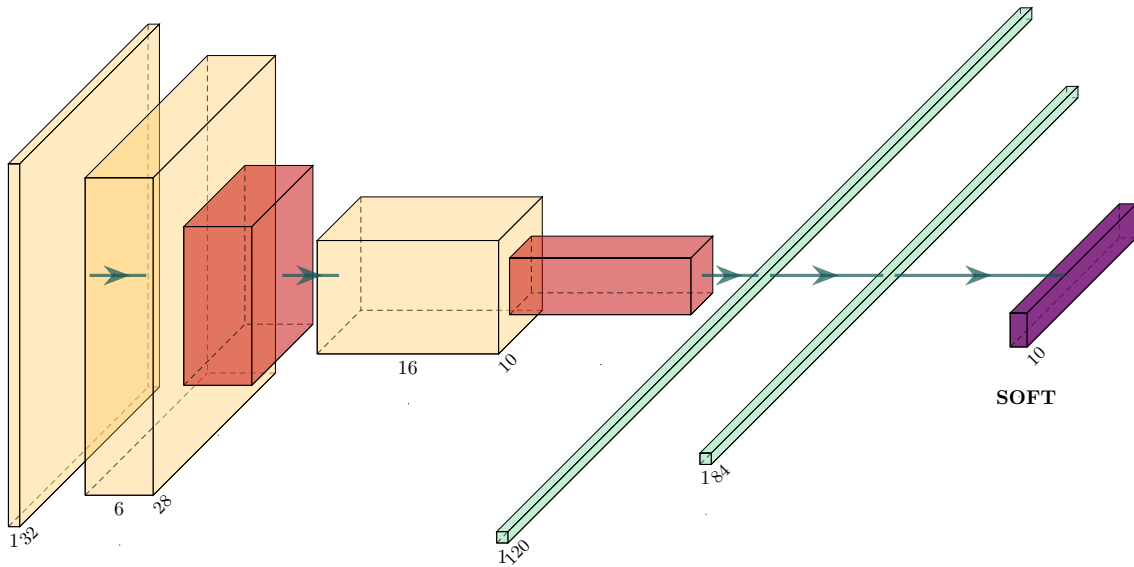


Figure 2.3: LeNet-5, a CNN used for high accuracy digit recognition [39].

layers.

Convolution in this case is an operation involving two functions like,  $f$  which is the input and  $g$  the kernel. The result of convolution can be represented by  $h$ . The function  $h$  can be computed by integrating the overlap of  $f$  and  $g$ . In a signal processing approach this is represented as:

$$h(t) = (f * g)(t) = \int_{-\infty}^{\infty} f(\delta t)g(t - \delta t) d\delta t \quad (2.5)$$

The convolution operation is commutative and when the inputs are images, we can discretize the above equation and can work in the integer space. Also the convolution needs to be per-

formed over two dimensions in the case of images. The convolution operation with kernel  $K$  over image  $I$  is progressed by the sliding window method. The kernel  $K$  is slid incrementally over the input image producing a feature map  $H$ . Mathematically:

$$H(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n). \quad (2.6)$$

To suppress the unimportant regions in the feature maps, a pooling operation layer is employed in the network, this layer also reduces the image dimensions. The common types of pooling layers are Max pool and Average pool. This layer adds some spatial invariance to the network as well. Simply composing a network with multiple convolutional and pooling layers is not enough, to inject non-linearity, a transfer function is employed. A rectified linear unit or (ReLU) is one of the most commonly used activation or transfer functions. The output of ReLU differs from that the sigmoid function as ReLU is not limited between 0 and 1. The lower bound of ReLU and sigmoid functions is 0 and their outputs are always positive. The ReLU's upper bound is the input itself  $x$ .

The final layer of the network is known as the output layer and the output is a vector with a number of elements equal to the classes in the training data in the classification setting. To obtain the class probabilities a specific layer known as a softmax layer is used after the final layer. To train a CNN, a gradient descent approach is used typically. In the context of this dissertation, all the methods use a modified Stochastic Gradient Descent (SGD) with Nesterov momentum to minimize the cross entropy loss. For a more thorough understanding of the various layers and optimization methods used in the training of CNNs please refer to [22] as a detailed discussion of these methods is out of scope of this document.

# Chapter 3

## CopyPaste

This chapter is dedicated to our regularization technique called, CopyPaste.

### 3.1 Datasets

We evaluate our classification approach on the ImageNet-1K dataset [60]. The dataset consists of about 1.28M million training images and 50 thousand validation images with 1000 categories (classes). In addition to class labels, about 38 percent of the training images have 2 dimensional coordinates specifying a rectangular bounding box to enclose the pertaining object. We use the bounding box annotations for these images in our approach for classification. ImageNet pre-trained models are used for many other visual recognition tasks like object recognition. And to evaluate the object detection performance, we use the MS-COCO dataset [44]. It has 80 object classes for object detection and there are about 118 thousand images for training and 5 thousand images for validation. We also evaluate on Pascal VOC 2007 and 2012 [17] `trainval` data of about 33 thousand images and 20 categories. The object detection models are then validated on VOC 2007 `test` data consisting of about 5 thousand images.

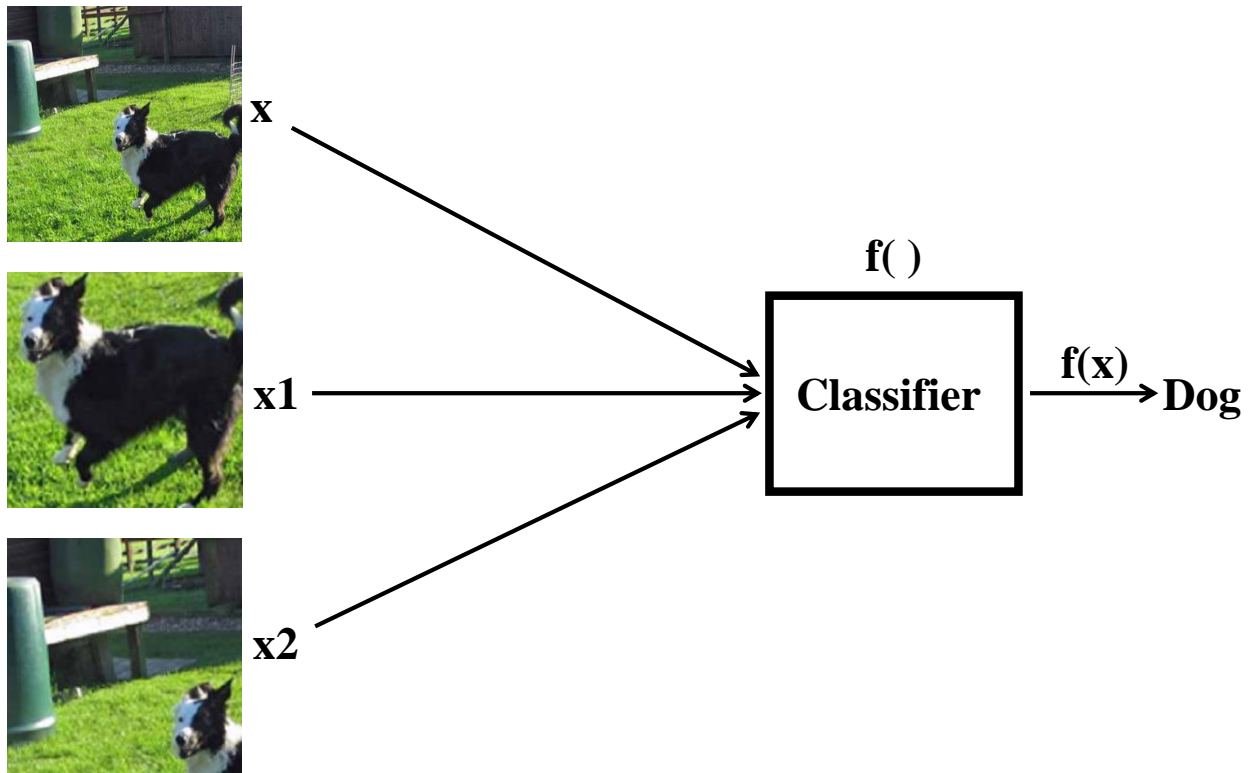


Figure 3.1: In a classification approach, the location of the object is not accounted for while training and the expected outputs (post training) for all the 3 variations of the image should indicate Dog. The CNN has to learn to localize the pertinent object (dog in this case) and produce an output indicating the presence of the object.

## 3.2 Approach

In the past year, CNNs that have been trained using numerous data augmentation approaches have shown state of the art performance on standard datasets [69, 74] and our approach is inspired by them. The standard training of CNNs on images, including the recent data augmentation approaches are based on the idea that for a classification problem as long as the object corresponding to the class is present in any location in a given image then the label of the image is the same as that of the object as shown in figure 3.1. In a given image, the location of the object is irrelevant to a classification CNN and the CNN has to learn to localize and suppress the context and other irrelevant objects in the image. This is true for



Figure 3.2: Our method uses bounding box information to paste objects belonging to the same class in a given image. The red bounding box is an example of bounding box annotation and the rest of the image is considered context. The green bounding box shows the object that has been pasted from another image of the same class if bounding boxes are available. The bottom two rows are sample images generated on the fly by our approach. The labels of images in the second row (left to right) are, ‘Goldfish’, ‘Cauldron’, and ‘Alligator lizard’. The labels of images in the third row (left to right) are, ‘Tench’, ‘Snow leopard’, and ‘Robin’.

all strict classification problems.

Our approach involves using bounding box information to augment the training dataset for a classification task. The location of the object in a given image is specified by a rectangular bounding box. In the first row of figure 4.1, red dotted lines are used to illustrate a bounding box. Using bounding box annotations to address object detection is well studied [21], but it has not been used in classification problems for augmentation to our knowledge on the ImageNet dataset. Nikita et al. [16] have shown that using the training data to create novel synthetic images and respecting the context can improve the object detection performance.

However, even the most recent approaches in the area of classification do not incorporate bounding box information of the objects present in them. We use bounding box information when available during training of our classifier by identifying the class of a given image and pasting an object of the same class on it. Figure 4.1 shows our approach on some of the training samples. The bottom two rows are samples generated by our approach. We use objects and paste them anywhere in a given image, the intuition being that objects of the same class share similar context. Our approach is label preserving because we do not have to change the labels of our transformed images. We can generate infinitely many combinations of objects and context at different scales and locations. This is a powerful way to regularize CNNs. Our approach also improves classification performance, small object detection and localization in general.

Figure 3.3 shows the differences between our method and the other recent approaches. CutMix and RICAP change the label of an image based on the proportion of pixels represented by each class and the ‘mixing’ of samples is random. Our method is label preserving (correct labels are passed to the classifier during training) as opposed to CutMix and RICAP which use random crops to mix samples from different classes. Incorrect labels hurt the training of CNNs, and CutMix and RICAP often use context crops instead of object crops, but supply the label of the object in the image to the CNN. We argue that this inconsistency can lead to context dependence as the authors of RICAP [69] point out in their work.

### 3.3 Contributions

In this chapter, we have contributed to the research of data augmentation approaches for classification CNNs. The contributions include:

- A novel way to use bounding box annotations for data augmentation of classification CNNs with complete control over the augmented object location and scale.
- We are able to produce a quantifiable improvement in classification accuracy of the standard CNN baseline on ImageNet dataset [60].
- We outperform standard CNN baseline on MS-COCO [44] object detection.
- We combine our approach with other data augmentation methods to realize further improvements in performance of the other methods.

## 3.4 Related Work

We propose a novel data augmentation technique and show how it is related to recent developments in this area.

### 3.4.1 Classic Data Augmentation

Traditionally, any label preserving transformation on an input image is considered to help regularize a classifier. The authors of AlexNet [32] employed random cropping and horizontal flipping methods when they surpassed the performance of traditional machine learning approaches in 2012. Randomly cropping a given image during training prevents CNNs from over-fitting to the scale or location of the object. Flipping an image improves the generalization to view points. The authors of [32] used principal component analysis for each of red, green and blue channels to add lighting noise to a given image. Color jitter is another commonly used augmentation approach and it applies a constant value to the hue, value, and saturation channels of an image. These approaches are still a part of standard CNN

training. However, they are not sufficient to reduce the effect of over-fitting on their own, as newer CNN architectures can have almost 100 million parameters and billions of floating point operations.

### 3.4.2 Random Noise

The random noise class of data augmentation methods [13, 79] work by masking random regions of an input image with zeros. These methods may accidentally erase the pertaining object in a given image, forcing the CNN to rely purely on context at times to make a prediction. Relying on context only is not an efficient way to train neural networks. This approach can also be applied to the inputs of hidden layers (also known as feature space) as well. The authors of DropBlock [20] have used this technique (applying random noise to the feature space) to obtain better generalization. The authors of [7, 62] use random noise based methods to obtain better localization characteristics.

### 3.4.3 Mixed Sample

In the case of classification CNNs, the ground truth/labels are represented by a one-hot representation of class probabilities. These labels consist of 0s and 1s, with a single 1 indicating the pertinent class in a given label vector. The latest work in the area of data augmentation involves using samples from different classes and changing the expected output of the CNN to output a probability distribution based on the number of pixels/intensity of pixels represented by each class.

The authors of Mixup [70, 77] use alpha blending (weighted sum of pixels from two different classes) and apply the blending weights to the corresponding labels. The authors of [74] show that Mixup is detrimental to the performance as the generated images are not natural

as opposed to cut-paste methods like RICAP and CutMix.

Having soft labels (label values in between 0 and 1) is an advantage while training CNNs as the probabilities output by a CNN are never perfect 0s and 1s and the CNNs usually produce overconfident results even when their predictions are wrong. The authors of CutMix and RICAP [69, 74] also use soft labels in their approach. In our case since the samples being mixed are from the same class, we do not have to change the label. This allows for easy integration with CutMix and RICAP [69, 74] as well.

### 3.4.4 AutoAugment

In order to learn the best augmentation strategy dynamically during training, the authors of AutoAugment [10] use reinforcement learning to learn the best combination of existing data augmentation methods.

## 3.5 Object and Context based Data Augmentation

In this section, we describe the data augmentation methods of interest in mathematical detail.

Consider  $D = \langle (\mathbf{x}_i, z_i) \rangle_{i=1}^N$  to be a dataset consisting of  $N$  independent and identically distributed real-world images belonging to  $K$  different classes. Let  $\mathcal{X}$  represent the set of images, and let  $\mathcal{Y}$  denote the set of ground-truth class labels. Sample  $i$  consists of the image  $\mathbf{x}_i \in \mathcal{X}$  along with its corresponding label  $z_i \in \mathcal{Y} = \{1, 2, \dots, K\}$ . Let  $f_\theta$  represent the CNN classifier with model parameters  $\theta$ . The predicted class is  $\hat{y}_i = \operatorname{argmax}_{y \in \mathcal{Y}} \hat{p}_{i,y}$ , where  $\hat{p}_{i,y} = f_\theta(y|\mathbf{x}_i)$  is the computed probability that the image  $\mathbf{x}_i$  belongs to class  $y$ .

Let  $z_i$  represent the one-hot encoding of label  $y_i$ .

In the case of Mixup with two samples,  $(x_i, z_i) \sim D$  and  $(x_j, z_j) \sim D$ , the transformed image and label pair  $(\tilde{x}, \tilde{z})$  are computed as,

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j \quad (3.1)$$

$$\tilde{z} = \lambda z_i + (1 - \lambda)z_j \quad (3.2)$$

where  $\lambda \sim \text{Beta}(\gamma, \gamma)$  and  $\gamma$  is a hyperparameter set to 0.3 by default.

In the case of RICAP with 4 samples,  $(x_1, z_1) \sim D$ ,  $(x_2, z_2) \sim D$ ,  $(x_3, z_3) \sim D$  and  $(x_4, z_4) \sim D$ . The authors patch the upper left, upper right, lower left, and lower right regions of the four images to generate a new sample.

Let  $W$  and  $H$  denote the proportions of the original image.

To crop the four images  $k$  following the sizes  $(w_k, h_k)$ , the authors generate the coordinates of the upper left corners of the cropped areas randomly.

The ratio  $W_i$  is computed proportional to the area occupied by each patch from one of the four images. The label  $\tilde{z}$  is computed as:

$$\tilde{z} = \sum_{k \in \{1,2,3,4\}} W_k z_k \quad (3.3)$$

$$\text{where } W_k = \frac{w_k h_k}{WH} \quad (3.4)$$

In the case of CutMix with two samples, the label  $\tilde{z}$  is computed as:

$$\tilde{z} = \sum_{k \in \{1,2\}} W_k z_k \quad (3.5)$$

$$\text{where } W_k = \frac{w_k h_k}{WH} \quad (3.6)$$

Our method is called CopyPaste and it is used to generate a new training sample  $(\tilde{x}, \tilde{z})$  by combining two training samples  $(x_i, z_i)$  and  $(x_j, z_j)$  from the same class.

The training sample  $(\tilde{x}, \tilde{z})$  is used to train the model with the same label as the classes of both samples are the same. We use the 480K images from the ImageNet dataset that have bounding box annotations and the label remains unchanged.

$$\tilde{z} = z_i \quad (3.7)$$

For implementation, we use a network storage as our approach continuously changes the proportion of original samples and CopyPaste-d samples over the course of training and is compute intense. We handle our augmentation operation using a dedicated node.

Our approach can be used without any changes to the loss function or the network architecture.

We use traditional data augmentation methods and scale/crop objects before pasting them on images of the same class. We believe this helps our performance when detecting small and multiple objects.

## 3.6 Experiments

We show results using extensive experiments on both classification and object detection tasks.

Table 3.1: ImageNet classification accuracies using ResNet-50 architecture. ‘\*’ denotes results reported in their paper.

Model	# Parameters	Top-1 Err (%)	Top-5 Err (%)
ResNet-50 (Baseline)	25.6 M	23.28	6.92
ResNet-50 + RICAP [69]	25.6 M	21.9	6.17
ResNet-50 + CutMix [74]	25.6 M	21.60	6.04
ResNet-50 + DropBlock* [20]	25.6 M	21.87	5.98
ResNet-50 + CutMix [74] + CopyPaste	25.6 M	<b>21.48</b>	<b>5.85</b>
ResNet-50 + RICAP [69] + CopyPaste	25.6 M	21.75	6.16
ResNet-50 + CopyPaste	25.6 M	22.23	6.376

### 3.6.1 ImageNet Classification

We train our classifiers on ImageNet-1K dataset [60]. The classification CNNs in this dissertation are trained to detect 1000 different classes of images as the popular ImageNet [60] dataset consists of 1000 classes and is widely used by researchers to benchmark new approaches for classifying data and to adapt to secondary tasks such as object detection.

The dataset consists of 1.2M training images and 50K validation images of one thousand categories. We use the usual data augmentation strategies for all methods. We train all the CNN models for 300 epochs starting with a learning rate of 0.1 and decayed by 0.1 at epochs 75, 150, and 225 using a batch size of 256 using ResNet-50 [25] architecture for all of our experiments for a fair comparison. We use the standard data augmentation used by [25, 29, 65] for all the methods. We do not use any dropout [63] but employ Batch Normalization [30] as it is a part of the standard ResNet architecture.

Results for the classification task on ImageNet are provided in table 3.1. Our approach by itself does not have the best performance as it can be considered as a special case of CutMix. However, when combined with CutMix we see the best performance **21.488%** top-1 error.

In addition to measuring classification accuracy we are also interested in computing the dis-

Table 3.2: ImageNet cross entropies and gaps using ResNet-50 architecture.

Model	# Parameters	Cross Entropy Training	Cross Entropy Validation	Distribution and Generalization Gap
ResNet-50 (Baseline)	25.6 M	0.685	0.965	0.280
ResNet-50 + CutMix [74]	25.6 M	2.022	0.880	-1.142
ResNet-50 + CopyPaste	25.6 M	0.755	0.904	0.149

tribution and generalization gaps of the classifiers. Heavy augmentation like MixUp during training produces samples that are unnatural compared to the distribution of unaugmented validation samples. Ideally, the distribution gap between augmented and unaugmented sample distributions (augmented samples belong to a slightly different distribution compared to the unaugmented samples) can be measured by computing the mean cross entropy loss across all training samples. We compute the difference between the mean cross entropy of augmented training samples and the mean cross entropy of unaugmented validation samples thereby computing both distribution gap as well as generalization gap together 3.2. Our approach produces the lowest gap compared to the other baselines.

Our approach is called CopyPaste, but we also refer to it as CutPaste as it involves the same operation of using bounding box operation to copy or cut the object region and paste it on top of a target image belonging to the same class. We do not use additional images than the train set of ImageNet, we cross reference the images that have bounding box annotation with the images in the train set and only use those images to augment.

Unsurprisingly, we begin to overfit during training as only 38 percent of ImageNet train set images have bounding boxes. To resolve this problem we employ a cyclic schedule when augmenting, for some epochs we supply clean ImageNet data and then slowly start increasing the proportion of CopyPaste-d samples. We repeat the cycle of using clean samples and CopyPaste-d samples about 15 times during training. We follow this approach when using our method in conjunction with CutMix or RICAP as well.

When copying we use the standard data augmentation operations on the object patch before pasting in on a given image. Figure 3.4 shows the compute plot of our dedicated node performing augmentation and copying the clean ImageNet samples cyclically. Figure 3.5 shows the validation error of the ImageNet dataset using standard ResNet-50 on CopyPasted data and the clean ImageNet data. Standard ImageNet training starts over-fitting around epoch 150. However, our approach continues to improve and produces a lower top 1 error rate.

Figure 3.6 shows that our model produces the lower error and is improving continuously as our last good error rate drops consistently.

### 3.6.2 Object Detection using Pretrained Model

We use the implementation of Faster RCNN [58] adapted to use the ResNet-50 backbone. The ResNet-50 backbone is trained with different augmentation approaches and fine-tuned on Pascal VOC 2007 and 2012 [17] `trainval` data. The object detection models are then validated on VOC 2007 `test` data using the mAP measure.

We follow the fine-tuning strategy of the original methods [58]. We use the implementation of [73] to benchmark the backbones trained using different approaches. Results are provided in table 3.3. Our approach obtained the best mAP (mean average precision) of the two most important baselines.

We train the models with a batch size of 8 and initial learning rate of 0.01 decayed after 5 epochs and trained for a total of 12 epochs. We report the best mAP out of the last 3 epochs for all the methods. We drop the learning rate at 8 epochs and train for 12 epochs and report the best mAP out of the last 4 epochs in table 3.4

For MS-COCO, we use the same implementation. but train for 4 epochs and drop the

Table 3.3: Object detection mean average precision values using ResNet-50 backbone and Faster-RCNN. We report the best out of last 3 epochs for all methods.

Backbone	# Parameters	mAP mAP (%)
ResNet-50 (Baseline)	25.6 M	77.63
ResNet-50 + CutMix [74]	25.6 M	77.69
ResNet-50 + CutMix [74] + CopyPaste	25.6 M	77.69
ResNet-50 + CopyPaste	25.6 M	<b>77.90</b>

Table 3.4: Object detection mean average precision values using ResNet-50 backbone and Faster-RCNN. We report the best out of last 4 epochs for all methods.

Backbone	# Parameters	mAP mAP (%)
ResNet-50 (Baseline)	25.6 M	77.90
ResNet-50 + CutMix [74]	25.6 M	77.74
ResNet-50 + CutMix [74] + CopyPaste	25.6 M	78.19
ResNet-50 + CopyPaste	25.6 M	<b>78.35</b>

learning rate and train for 2 more epochs. We use a batch size of 16 and report the best numbers out of the last two epochs.

The results show that MS-COCO mAP performance of CutMix is improved when CopyPasted samples are used instead of clean ImageNet samples as shown in table 3.5.

In the case of small objects, as shown in table 3.6, our approach is comparable to CutMix and provides the best improvement when a backbone trained with CutMix and CopyPaste is used.

Table 3.5: Object detection mean average precision values using ResNet-50 backbone and Faster-RCNN. We report the best out of last 2 epochs for all methods.

Backbone	# Parameters	mAP 0.50:0.95
ResNet-50 (Baseline)	25.6 M	31.3
ResNet-50 + CutMix [74]	25.6 M	31.8
ResNet-50 + CutMix [74] + CopyPaste	25.6 M	<b>32.3</b>
ResNet-50 + CopyPaste	25.6 M	31.7

Backbone	# Parameters	mAP 0.50:0.95 (small)
ResNet-50 (Baseline)	25.6 M	12.6
ResNet-50 + CutMix [74]	25.6 M	13.3
ResNet-50 + CutMix [74] + CopyPaste	25.6 M	<b>13.7</b>
ResNet-50 + CopyPaste	25.6 M	13.2

Table 3.6: Object detection mean average precision values using ResNet-50 backbone and Faster-RCNN for small objects in COCO. We report the best out of last 2 epochs for all methods.

In the next section, we discuss some visualizations that provide more insight into the localization ability of our model.

### 3.6.3 Qualitative results

We show qualitative results comparing various methods on ‘CopyPaste-d’ samples as well as regular samples. Our approach shows better localization characteristics, we use implementation of [80] to compute the class activation maps. These maps show the most discriminating regions that the classifier relies on to make its predictions. Class activation maps (CAMs) are extremely useful in debugging CNNs as they show whether the model is simply memorizing the samples or learning to localize pertinent objects and classify properly. As shown in figures 3.7 and 3.8, our approach pays attention to small objects and covers a greater extent of the pertinent object(s) compared to other methods.

Figure 3.9 shows that when multiple objects are present in a given image (last row), CopyPaste is more attentive compared to standard ImageNet training.

## 3.7 Conclusion

We show that bounding box annotations provided in the ImageNet dataset even for a subset of images (480k) can be used to CopyPaste using images of the same class to help improve classification and object detection performance.

Our approach however has a significant overhead and we use a network storage and separate node to augment our data on the fly.

Our approach shows promise in using bounding box annotations to train better classifiers. Our results on ImageNet and PASCAL VOC certainly show that our approach helps in training more discriminating classifiers.



Figure 3.3: We show the differences in recent augmentation methods and the corresponding labels for images generated using the different approaches. Because CutMix and RICAP have no localization information, they are more likely to assign the wrong label to a given crop (red bordered images indicate wrong labels and green bordered images indicate the correct labels), whereas our approach generates correct labels all the time. ImageNet column shows the standard images from the dataset without any augmentation so the labels are not changed during training like the other methods. (Note: The border colors are purely for illustrative purposes. The CNN does NOT receive the samples with the colored borders.)

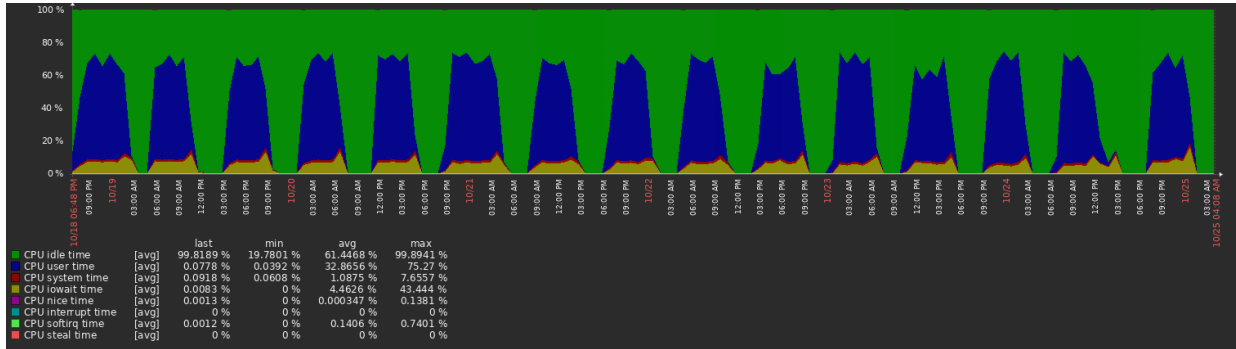


Figure 3.4: We use a compute node that is different from our main GPU node to handle the tremendous compute imposed by modifying the train set on the fly. The gaps in between the high CPU use show the times during which we utilize clean ImageNet samples.

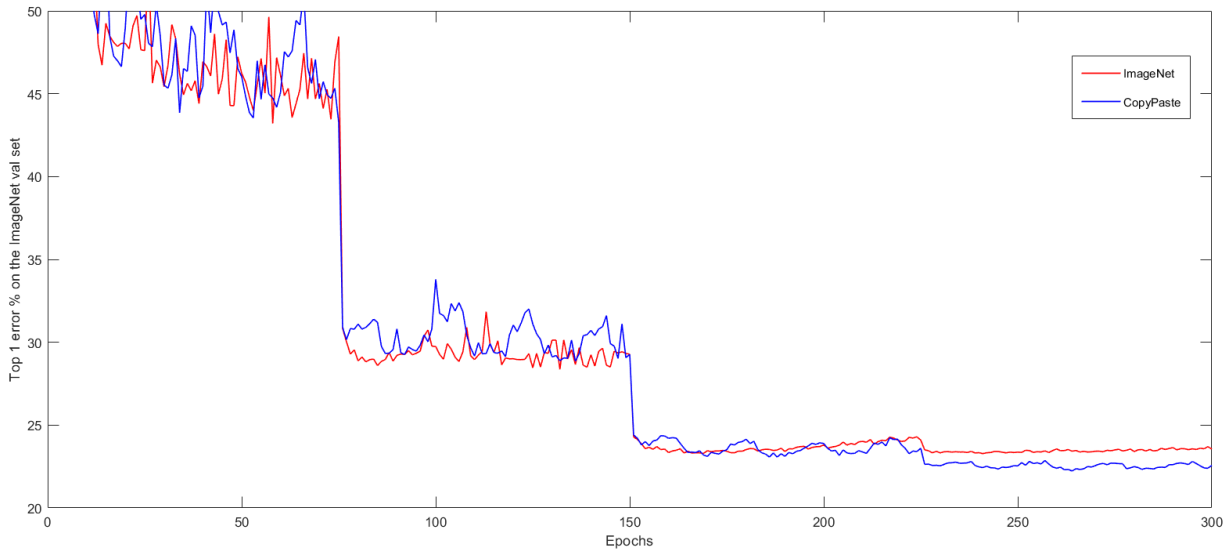


Figure 3.5: We plot the validation error during the training of the baseline approach as well as our approach

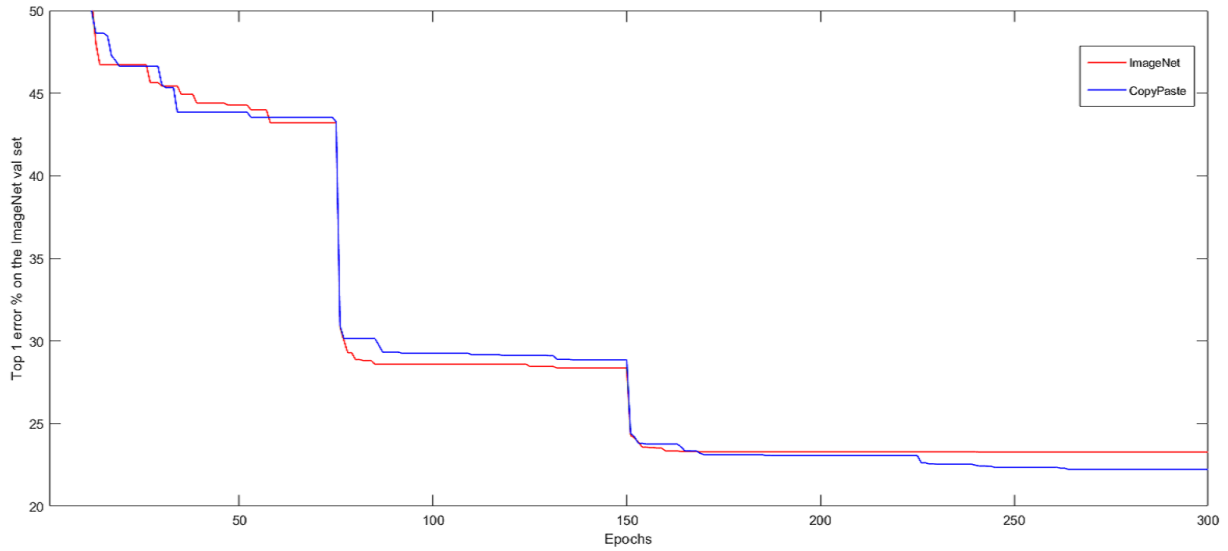


Figure 3.6: We plot the last lowest error (last good performance) during the course of training the baseline model as well as our approach

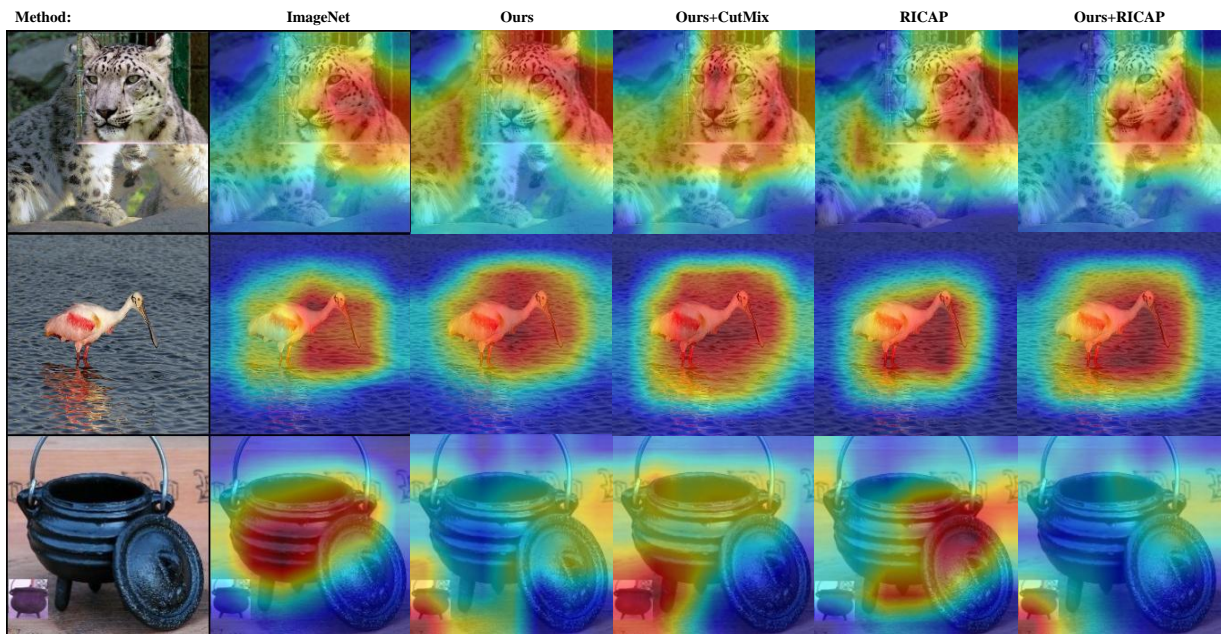


Figure 3.7: Qualitative results using class activation maps to show the most pertinent regions used by each method to make the prediction.

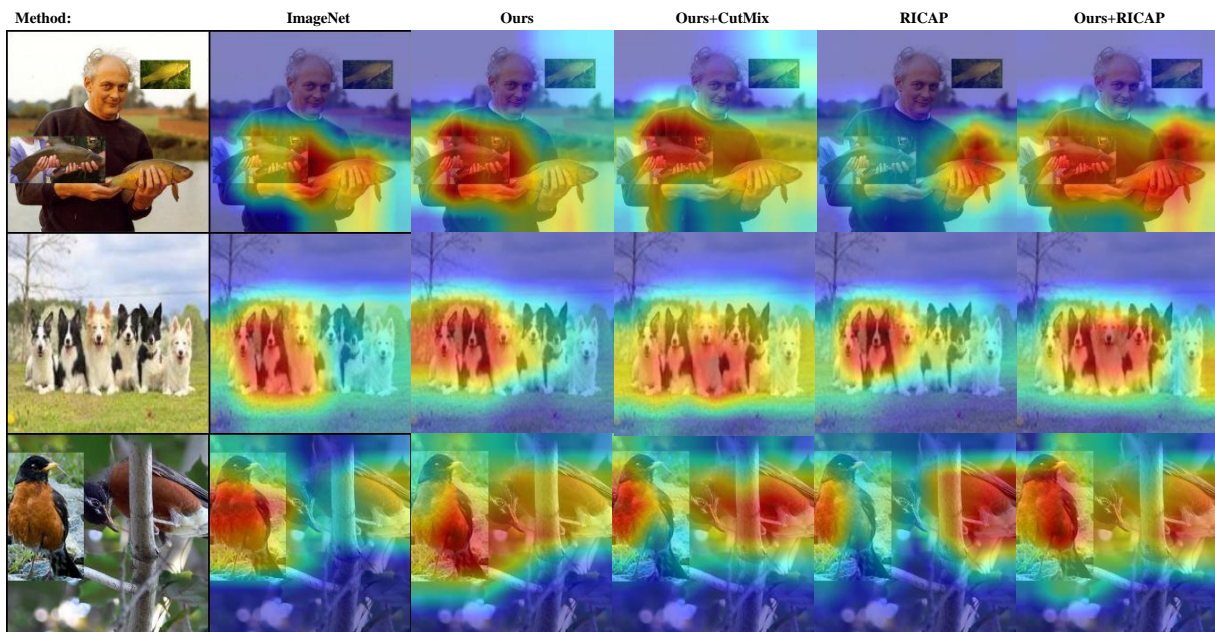


Figure 3.8: More qualitative results using class activation maps to show the most pertinent regions used by each method to make the prediction.

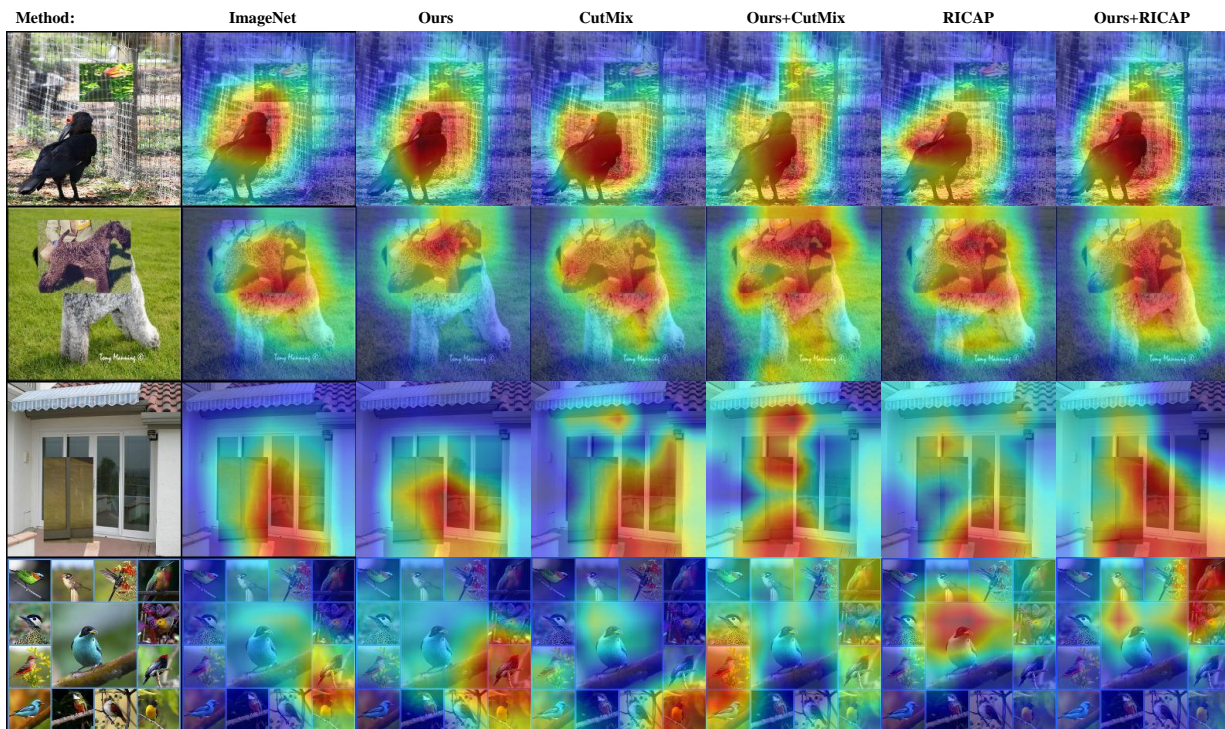


Figure 3.9: Qualitative results using class activation maps to show the most pertinent regions used by each method to make the prediction. Our approach also helps CutMix and RICAP localize the pertinent objects better.

# Chapter 4

## Adaptive Label Smoothing

The main contribution of this work is that we have developed a novel way to train classification CNNs using *adaptive* label smoothing. To demonstrate improved classification performance with less likelihood of overconfidence, we trained 20 classifiers and evaluated them on four popular datasets.

### 4.1 Related Work

Bias exhibited by machine learning models can be attributed to many underlying statistics present in datasets and model architectures [5, 78] including context, object texture [19], size, shape and color in the case of images. Various approaches to mitigate bias have been proposed [3, 9, 19] in recent years. Our approach produces high entropy predictions when context-only images are provided as input during inference, as we aim to learn the size of the relevant object within the image and classify it, instead of relying on context to produce a prediction.

Label smoothing was introduced by [66], and a more recent improvement is [15], correlations between the classes observed during training are used to provide a smooth label in an iterative fashion. This approach however has the problem of encouraging the model to make a mistake that is closer to the pertinent class. Also, using class similarity based on wordnet similarity may not always reflect image/feature similarity computed by a CNN. Knowledge

distillation [4] can be used to compute a soft label that is based on visual similarity, however this still doesn't mitigate the problem of encouraging class confusion especially for safety-critical applications. Both adaptive regularization and knowledge distillation involve changes to the network architecture.

The authors of AlexNet [32] employed random cropping and horizontal flipping methods when they surpassed the performance of traditional machine learning approaches in 2012. Traditionally, any label preserving transformation on an input image is considered to help regularize a CNN. Randomly cropping a given image during training prevents overfitting the scale or location of the object; flipping an image improves the generalization to view points. The random noise class of data augmentation methods [13, 79] mask random regions of an input image with zeros. Random noise based methods may accidentally erase the pertinent object in a given image and force the CNN to rely purely on context to make a prediction, this contributes to label noise. The authors of DropBlock [20] have used this technique (applying random noise to feature space) to obtain better generalization. Authors of AutoAugment [10] used reinforcement learning dynamically during training to learn the best combination of existing data augmentation methods.

In contrast to augmentation based approaches that manipulate the input but not the corresponding label, our approach regularizes classification CNNs by computing a label based on the proportion of the object being classified in a given random crop of the training sample. The latest work in the area of data augmentation uses samples from different classes and changes expected outputs to predict a probability distribution based on the number and intensity of pixels represented by each class. The authors of Mixup [70, 77] use alpha blending (weighted sum of pixels from two different classes), and apply blending weights to corresponding labels. The authors of CutMix and RICAP [69, 74] also use soft labels by cropping different regions and classes of images and 'mixing' the labels in matching proportions to

corresponding regions in the final augmented sample. The sample mixing based approaches above do not rely on object size when ‘mixing’ regions in images and computing the label. Conversely, our approach uses bounding box information to apply a smoothing factor based on the object’s size relative to the image size to produce a soft label without mixing the samples.

Calibration of CNNs is important as predictions need to be equally accurate and confident. Calibration and uncertainty estimation of predictors has been an ongoing interest to the machine learning community [12, 43, 52, 56, 75]. Bayesian binning into quantiles(BBQ) [53] was proposed for binary classification and beta calibration [33] employed logistic calibration for binary classifiers. In the context of CNNs, [24] proposed a temperature scaling approach to improve calibration performance of pre-trained models. Calibration has been explored in multiple directions; popular approaches to calibrate CNNs are to transform outputs of pre-trained models using approximate bayesian inference [46], or to use a special loss function to help regularize the model [35, 55] during training. Our approach is loosely related to the latter class of methods. Our work also relates to label smoothing proposed first by [66], with its applicability for many tasks explored by [55]; [72] applies dropout-like noise to the labels. Recently, [50] explored the benefits of label smoothing; apart from having a regularizing effect, label smoothing helps reduce intra-class distance between samples [50]. Another approach to calibrate CNNs was proposed by [49], using a special loss function and temperature scaling, the authors were able to obtain state-of-the-art calibration performance. Label smoothing also improves calibration performance of CNNs [49].

Contrasting previously discussed methods, our approach involves using hard labels multiplied by the objectness measure and obtaining a uniform distribution over all other classes when input images are devoid of pertinent objects. We do not change our loss function as opposed to [49]. Our approach can be described as a variant of label smoothing, employing an

*adaptive* label smoothing approach that is unique to every training sample as it accounts for object size. To our knowledge, we are the first to apply *adaptive* label smoothing to train image classification CNNs.

The objectness is computed using bounding box information during training. CNNs trained using hard labels produce ‘peaky’ probability distributions without considering the spatial size of the pertaining object. Our approach produces outputs that are softer and the peaks correspond to the spatial footprint of the object being classified as shown in figure 4.1.

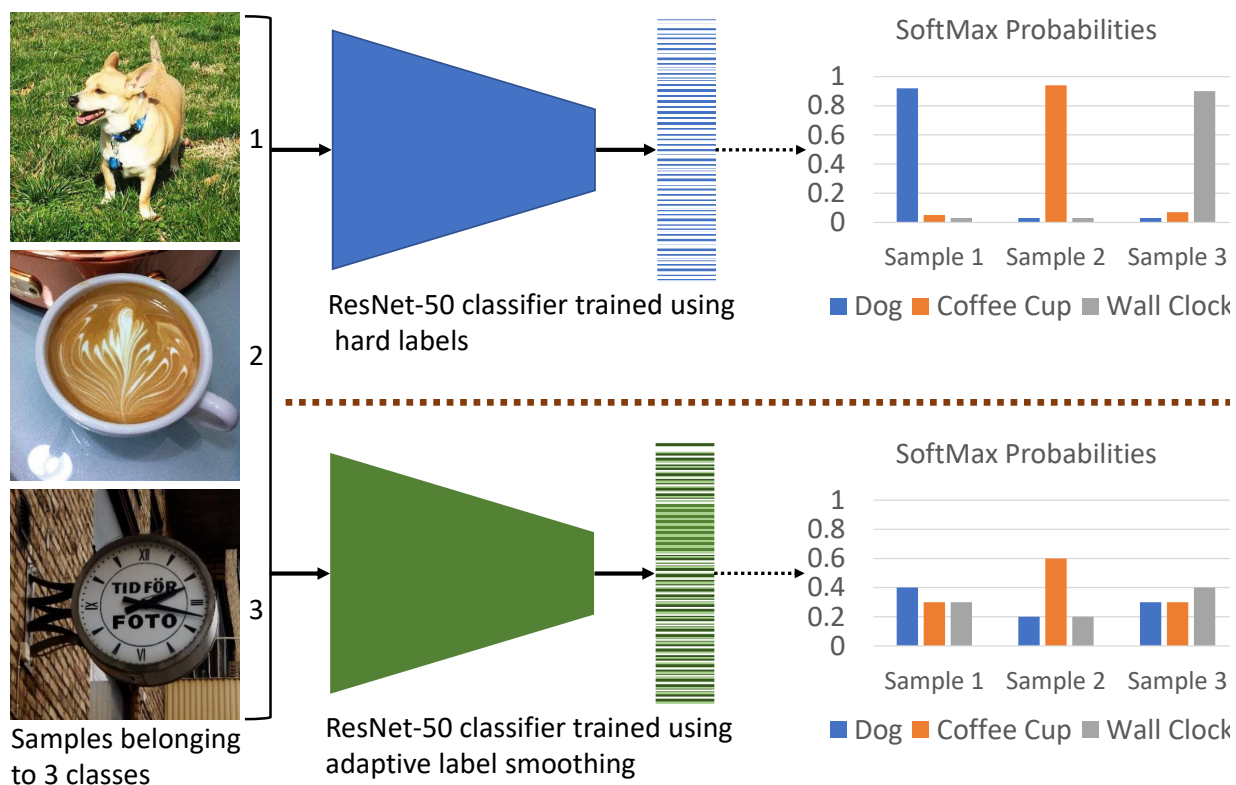


Figure 4.1: Hard-label and label-smoothing based approaches (top half of the figure) do not take into account the proportion of the object being classified. Our approach (bottom half) weights soft labels using the objectness measure to compute an *adaptive* smoothing factor.

## 4.2 Method

Consider  $D = \langle (\mathbf{x}_i, y_i) \rangle_{i=1}^N$  to be a dataset consisting of  $N$  independent and identically distributed real-world images belonging to  $K$  different classes. Let  $\mathcal{X}$  represent the set of images, and let  $\mathcal{Y}$  denote the set of ground-truth class labels. Sample  $i$  consists of the image  $\mathbf{x}_i \in \mathcal{X}$  along with its corresponding label  $y_i \in \mathcal{Y} = \{1, 2, \dots, K\}$ . Let  $f_\theta$  represent the CNN classifier with model parameters  $\theta$ . The predicted class is  $\hat{y}_i = \operatorname{argmax}_{y \in \mathcal{Y}} \hat{p}_{i,y}$ , where  $\hat{p}_{i,y} = f_\theta(y|\mathbf{x}_i)$  is the computed probability that the image  $\mathbf{x}_i$  belongs to class  $y$ .

Let  $z_i$  represent the one-hot encoding of label  $y_i$ . Following [66], the hard label  $z_i$  can be converted to soft label  $\tilde{z}_i$  using,

$$\tilde{z}_i = z_i(1 - \alpha) + (1 - z_i)\alpha/(K - 1) \quad (4.1)$$

Where,  $\alpha \in [0, 1]$  is a fixed hyperparameter. This is the standard procedure known as label smoothing.

A novelty of our approach is to make  $\alpha$  *adaptive*, calculating the value based on the relative size of an object within a given training image. Using the bounding box annotations available for the images in the dataset, we generate object masks. We apply the same augmentation transform (scale, crop) to the masks and compute the objectness score on the fly for every training image. Let the image width and height be denoted by  $(W, H)$  and the object width and height be denoted by  $(w, h)$ . The ratio  $\alpha$  is computed as

$$\alpha = 1 - \frac{wh}{WH} \quad (4.2)$$

The soft label  $\tilde{z}_i$  is computed as before.

We also explore a weighted combination of *adaptive* label smoothing and hard labels. To do

Table 4.1: Confidence and accuracy metrics on the validation set of ImageNet with all the objects removed using bounding box annotation provided by [8].

Method	# Train(N)	Accuracy Mean	Overconfidence Mean	Underconfidence Mean	Average confidence Mean
Hard Label	474 K	0.0633	0.2734	0.3362	0.2982
Label Smoothing	474 K	0.0618	0.1851	0.4816	0.2057
CutMix [74]	474 K	0.0921	0.1679	0.4696	0.2013
A. L. S. (Ours)	474 K	0.0473	0.0121	0.8409	0.0191

this, we introduce parameter  $\beta \in [0, 1]$  to determine the degree of *adaptive* label smoothing being applied. The setting  $\beta = 0$  corresponds to the case of classic hard labels. The soft label in this case is computed as,

$$\tilde{z}_i = (z_i(1 - \alpha) + (1 - z_i)\alpha / (K - 1))\beta + (1 - \beta)(z_i) \quad (4.3)$$

### 4.3 Experiments

In this section, we provide a description of the datasets used in our experiments, introduce some of the commonly used metrics for calibration of CNNs and describe our implementation details. We then discuss the merits of our approach and answer important questions related to applicability to transfer learning in an object detection setting, and we discuss the effect of using different types of labels during training in an ablative manner. We use ResNet-50 [25] for all our experiments.

Our classifier (A.L.S.) approach uses labels that are more accurate than any of the previous approaches when random cropping and scaling of images are applied during training. To our knowledge, almost all classifiers trained on ImageNet use random crop and scaling based augmentation to regularize. This ‘randomness’ forces the CNNs to rely on context rather than the pertinent object. Our approach uses bounding box labels to produce labels in an

adaptive way during training (Figure 1). To quantify context dependence, we used bounding box annotations on the 50K validation images, removed all objects and replaced the pixels with image mean [3,6]. Hard label approach had an accuracy of 6.3% with an average confidence of 0.29, label smoothing [42] predicted with an accuracy of 6.1% and an average confidence of 0.2, CutMix[47] had an accuracy of 9.2% with an average confidence of 0.2!. All these methods produced high confidence predictions on images with no objects present using pure context bias! Our approach had an accuracy of 4.7% and an average confidence of 0.02! We have an order of magnitude improvement in performance over recent baselines as our approach helps CNNs produce confidence based on the relative size of the pertinent object 4.1.

We provide detailed calibration metrics with mean and standard deviation for ImageNet and OpenImages classifiers in tables 4.2 and 4.3 respectively. We also provide AP (average precision) measures for different object sizes in table 4.4.

We provide more class activation maps to visualize the localization performance of baseline approaches, as well as our approaches in figures 4.3 and 4.4.

### 4.3.1 Datasets

As indicated in table 4.2, we have used different training datasets that are based on ImageNet-1K dataset [60]. ImageNet-1K consists of 1.2M training images and 50K validation images spanning 1K categories. As only 38% of ImageNet training images have bounding-box annotations, we distinguish these experiments from those trained on the full dataset. We use standard data-augmentation strategies for all methods and train all our models for 300 epochs starting with a learning rate of 0.1 and decayed by 0.1 at epochs 75, 150, and 225 using a batch size of 256.

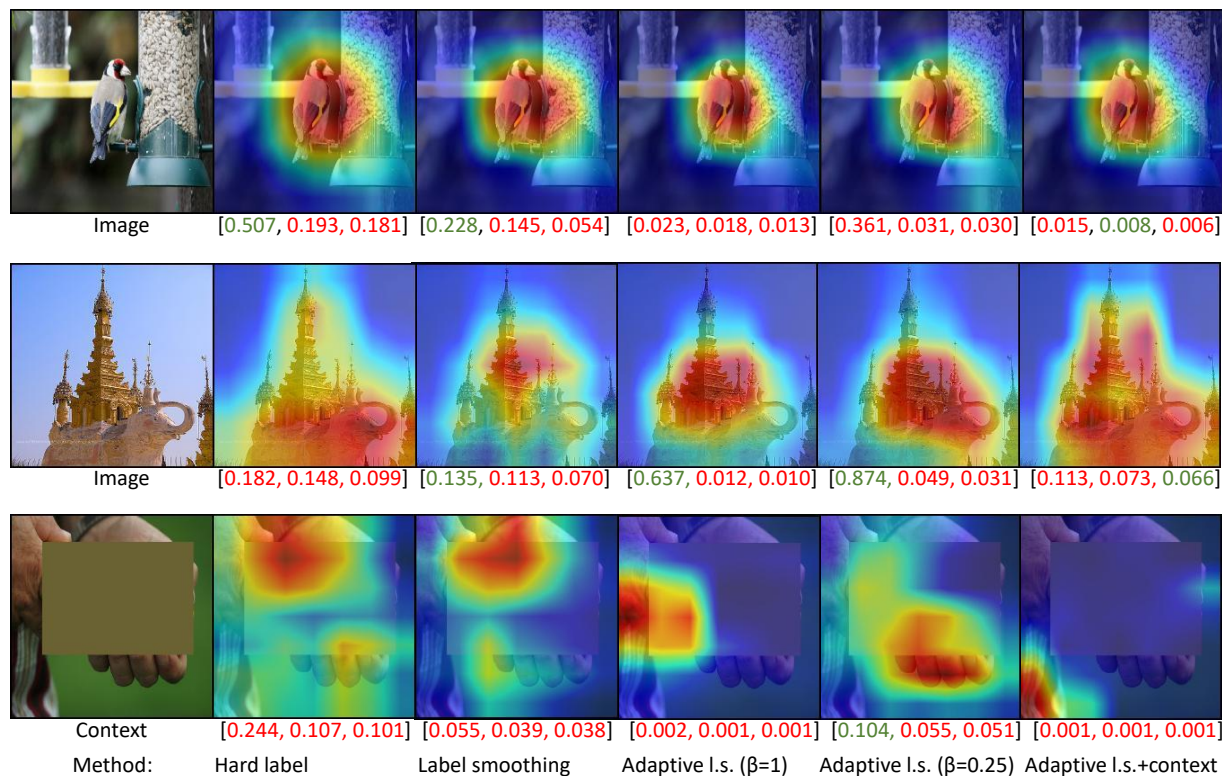


Figure 4.2: Examples of class activation maps (CAMs). These were obtained using the implementation of [6]. Two columns on the left show results for baseline CNNs using hard labels and standard label smoothing. Our approach, *adaptive* label smoothing (“adaptive l.s.”), is illustrated in the three columns on the right. Our technique produces high-entropy predictions and shows an improved localization performance. The values under each CAM represent the top three probabilities, with green indicating the pertinent class and red indicating an incorrect prediction.

The first 6 rows of the table employ the standard dataset for training. However, as our method needs object proportions, we use a subset of the standard ImageNet dataset that has bounding boxes (0.474M). These results are shown in the next 8 rows of table 4.2. To generate the ‘mask’ version, we make sure that only one object is present in a given image and ‘mask’ all other objects replacing them with pixel means. We use this version of the dataset derived from the 0.474M subset and identify the approach with ‘(mask)’ next to the method in table 4.2. We end up with about 54K more images as some ImageNet images have

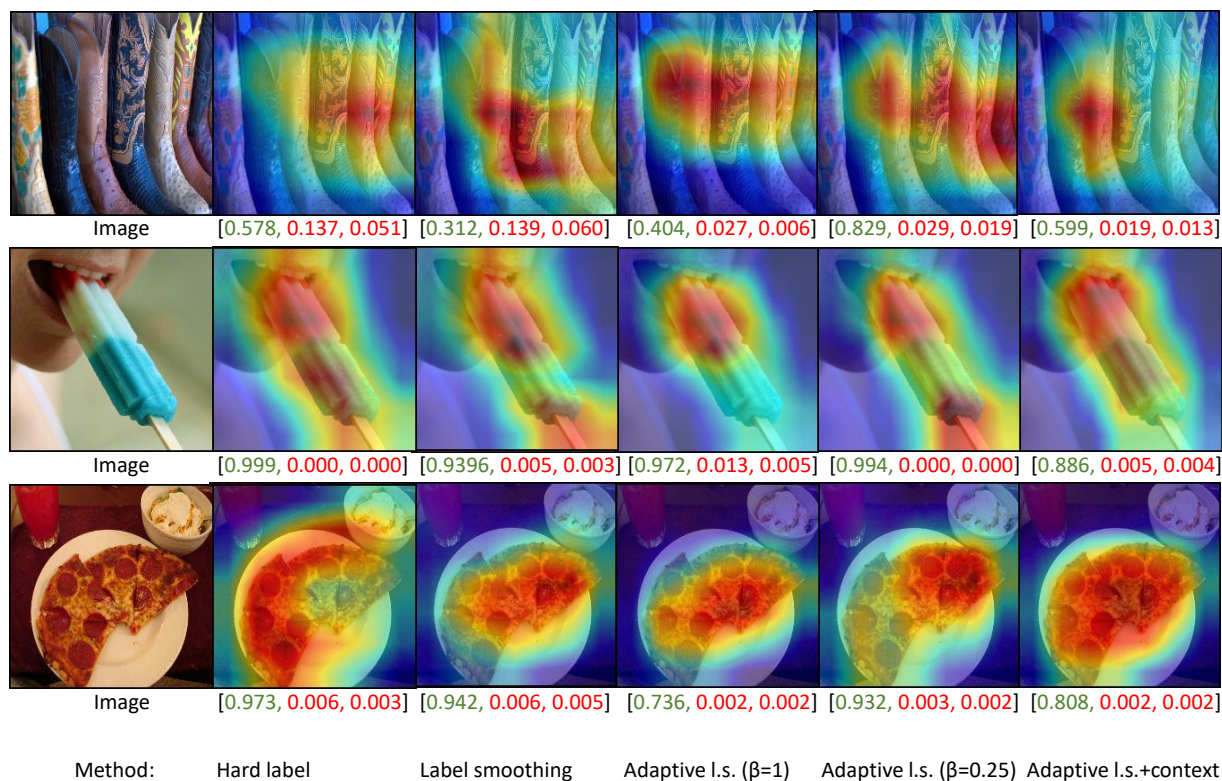


Figure 4.3: Examples of class activation maps (CAMs). These were obtained using the implementation of [6]. The second and third columns from the left show results for baseline CNNs using hard labels and standard label smoothing. Our approach, *adaptive* label smoothing (‘Adaptive l.s.’), is illustrated in the three rightmost columns. Our technique produces high-entropy predictions and shows an improved localization performance. The values under each CAM represent the top three probabilities, with green indicating the pertinent class and red indicating an incorrect prediction.

multiple annotated objects. Lastly, we generate another dataset that is devoid of any object altogether. We sample about 15% of the time from this dataset during training of one of our approaches, and the label generated for these methods is a vector of uniform probability distribution across 1000 classes. The idea is that when no objects are present in a sample, a CNN should produce a high-entropy prediction.

For validation, we use the validation set of [60] (V1) and the newly released ImageNetV2 set [57]. Specifically, we use the more challenging ‘MatchedFrequency’ set of images. The

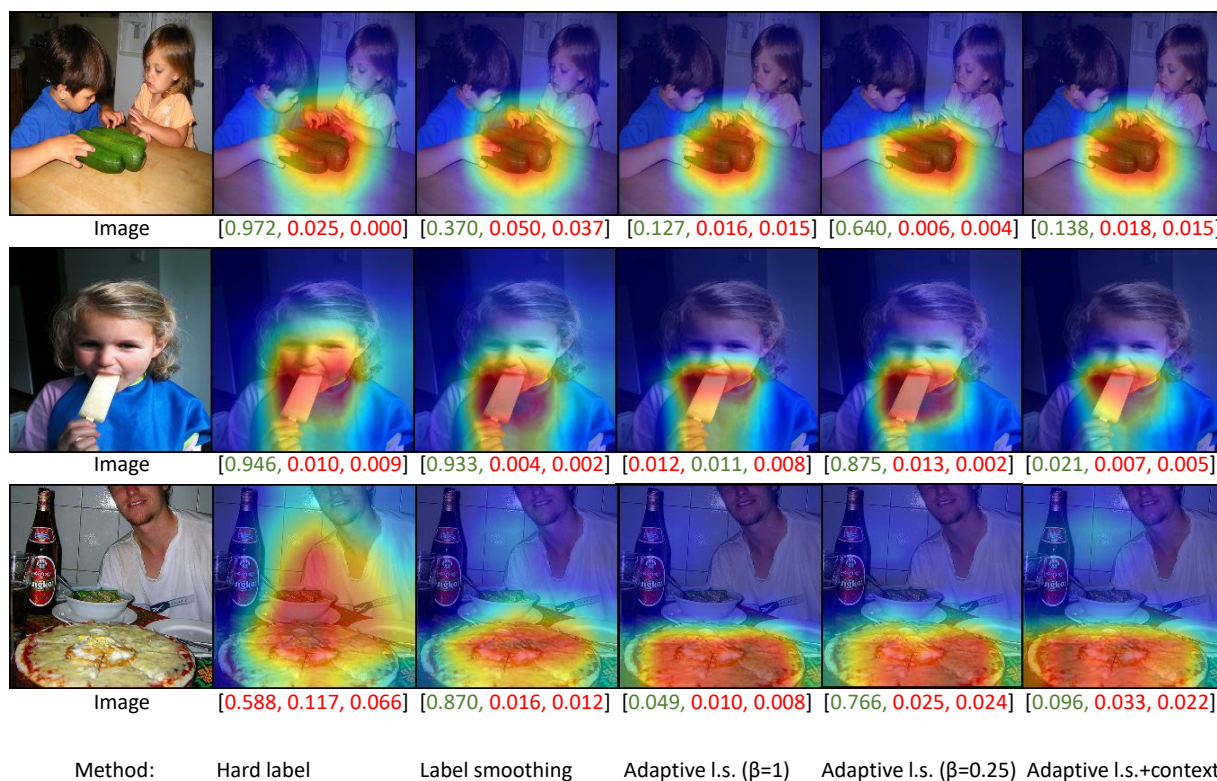


Figure 4.4: Examples of class activation maps (CAMs). These were obtained using the implementation of [6]. The second and third columns from the left show results for baseline CNNs using hard labels and standard label smoothing. Our approach, *adaptive* label smoothing (‘Adaptive l.s.’), is illustrated in the three rightmost columns. Our technique produces high-entropy predictions and shows an improved localization performance. The values under each CAM represent the top three probabilities, with green indicating the pertinent class and red indicating an incorrect prediction.

different validation sets are identified in the ‘Val.’ column of table 4.2.

We also used a portion of the OpenImages [36] dataset. More specifically, we used the object-detection version of the dataset, consisting of 600 classes and 1.7M images with bounding boxes. We selected a subset of these images and trained 5 classifiers. For a fair comparison with our ImageNet-based models, we matched the number of iterations and reduced the total epochs. We trained all our OpenImages models for 72 epochs starting with a learning rate of 0.1, and decayed by 0.1 at epochs 18, 36, and 54 using a batch size of 256.

To measure the transfer-learning ability of the representations learned by our classifiers, we used the challenging [44] dataset to obtain the results described in-4.4. The dataset consists of about 230K training images and we use the ‘minival’ validation set of 5K images.

## 4.4 Experimental setup

### 4.4.1 Datasets and splits

Our approach to create the different versions of ImageNet [60] to train our models are described in figure 4.5. We use the pixel means to mask all but one or all the objects using the same methodology as [3, 9]. We use the standard validation set along with ImageNet V2 [57] without any changes to the images.

In the case of OpenImages [36], we use the object detection dataset consisting of 600 classes and 1.7M images with 14M bounding boxes. However, the 600 classes also include many parent nodes and as this can contribute to label confusion. We remove all parent node classes and use only the leaf node classes. The dataset has bounding boxes for only a subset of images for commonly occurring objects and we remove these classes as well. Finally, we follow the approach of [45] and merge confusing classes. We end up with 480 classes and approximately 1.2M images. There are about 7 objects per image (average) in this subset and after applying the ‘mask’ method, we end up with approximately 6.8M images. Of these, about 1.3M images corresponded to the ‘man’ class and ‘women’ and ‘windows’ classes also had very high sample counts. We restrict the maximum number of images in a given class to around 50K and end up with roughly 2.2M images. We apply the same methodology to the val and test splits but we do not clip the sample counts per class.

Figure 4.6 is a visualization of the sample counts per class in the two datasets used for

training. Each rectangle represents a class and the size of a rectangle denotes the sample count for that class. Even after clipping the sample counts, the OpenImages dataset is very skewed compared to ImageNet as shown in figure 4.6, and we believe this imbalance along with the presence of about 7 objects per image makes OpenImages unsuitable for training good classifiers.

#### 4.4.2 Hardware and software

All our experiments were run on ‘Dell C4130’ nodes, equipped with 4 Nvidia V100 cards each. We used Docker to maintain the same set of libraries across multiple nodes. The host environment was running ubuntu 18.04 with cuda 10.2 installed. The docker environment used ubuntu 16.04 with cuda 9.0 and PyTorch 1.1 and Anaconda python 4.3. We will release all our code and pretrained models before the conference.

#### 4.4.3 Runtimes

Our *adaptive* label smoothing approach using the ‘mask’ version of ImageNet took approximately 74 hours and the hard label version took approximately 48 hours for 300 epochs. The object detection experiments took approximately 34 hours for 10 epochs.

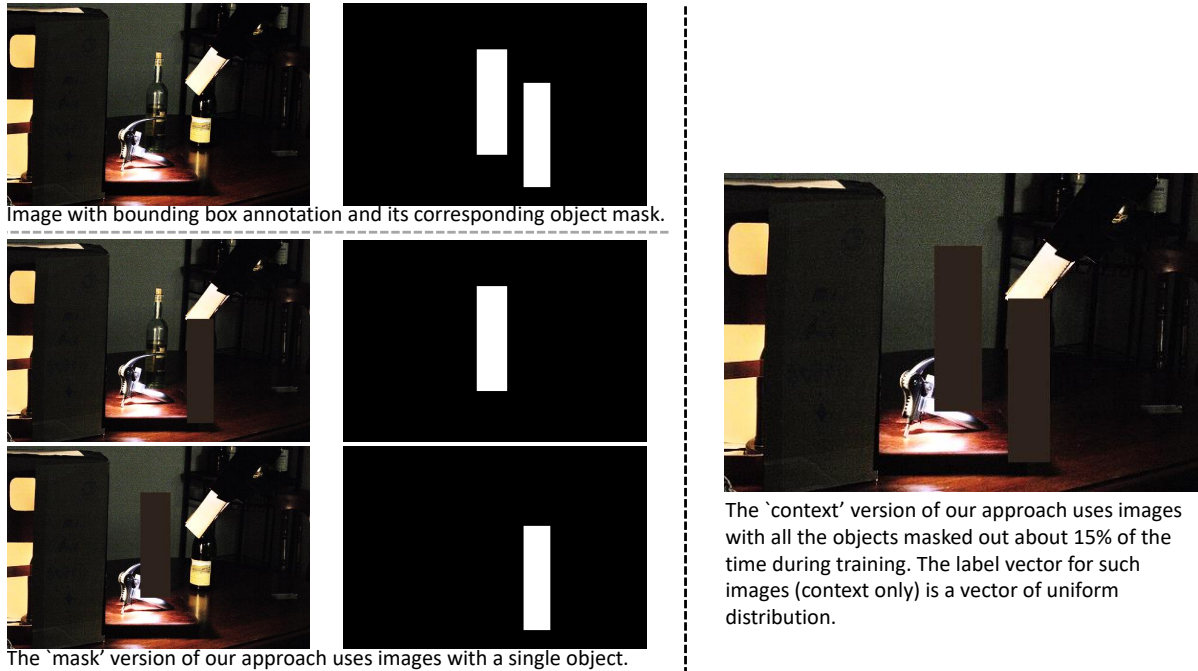
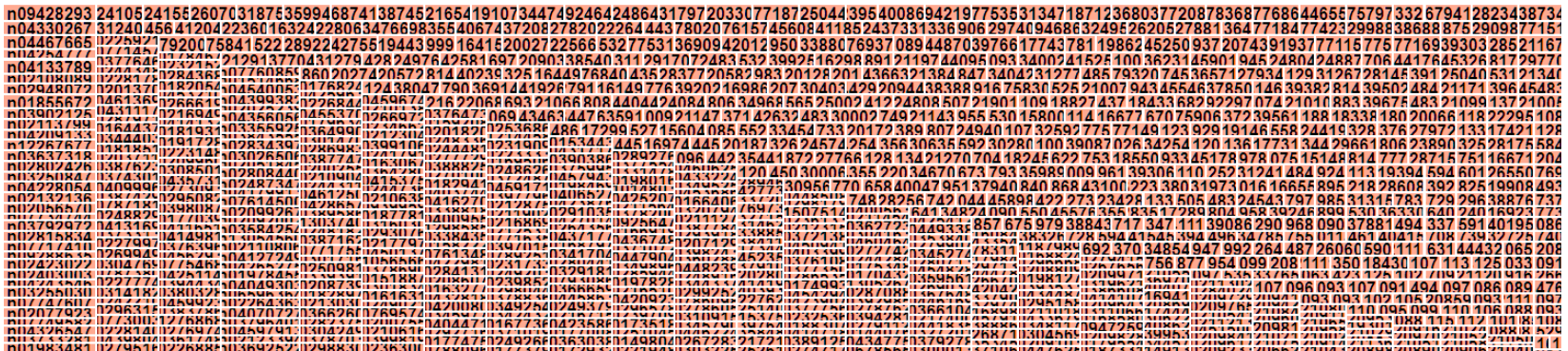


Figure 4.5: The first row of images in the left half of the figure are an example of the ImageNet dataset ( $N=0.474M$ ) that have bounding box annotations. We match the images from the training set of ImageNet-1K dataset with the corresponding '.xml' files included in the ImageNet object detection dataset. We then create object masks for each of the images. When applying any scaling and cropping operation to training samples, we apply the same transformation to the corresponding object masks as well. By counting the number of white pixels, we can determine the object proportion post transformation. We describe the two other approaches in the figure, the 'mask' version of our approach has a single object (for images with multiple bounding box annotations) and this version has 0.528M samples. Our approach helps generate accurate labels during training and penalizes low-entropy (high-confidence) predictions for context-only images like the example on the right half of the figure.

Table 4.2: Classification and calibration results with ImageNet. For a detailed explanation of the metrics please refer to the ‘Experimental setup’ section. ‘A.conf’, ‘O.conf’ and ‘U.conf’ refer to average confidence, overconfidence, and underconfidence scores. We provide ECE values for 100 bins and 15 bins mean scores along with their standard deviation (std).

Method	Val. Set	Train (N)	Acc. mean	Log-loss mean	ECE 100 mean	ECE 100 std	ECE 15 mean	ECE 15 std	MCE mean	MCE std	O.conf mean	U.conf mean	A.conf mean
Hard Label	V1	1.28M	0.769	0.963	0.062	0.003	0.045	0.003	0.284	0.069	0.582	0.100	0.826
Hard Label	V2	1.28M	0.647	1.643	0.131	0.006	0.099	0.006	0.664	0.166	0.538	0.131	0.752
CutMix	V1	1.28M	0.788	0.882	0.035	0.002	0.022	0.003	0.267	0.085	0.520	0.162	0.770
CutMix	V2	1.28M	0.661	1.499	0.094	0.007	0.051	0.005	0.817	0.183	0.485	0.192	0.699
RICAP	V1	1.28M	0.782	0.896	0.032	0.002	0.021	0.002	0.284	0.087	0.553	0.131	0.800
RICAP	V2	1.28M	0.663	1.533	0.108	0.010	0.072	0.013	0.697	0.168	0.516	0.165	0.728
Hard Label	V1	0.474M	0.669	1.568	0.104	0.005	0.093	0.005	0.347	0.068	0.558	0.123	0.771
Hard Label	V2	0.474M	0.543	2.365	0.171	0.014	0.148	0.012	0.759	0.162	0.520	0.154	0.697
CutMix	V1	0.474M	0.689	1.368	0.032	0.002	0.017	0.002	0.167	0.029	0.456	0.209	0.687
CutMix	V2	0.474M	0.577	2.021	0.100	0.008	0.050	0.008	0.517	0.169	0.421	0.248	0.612
Label Smoothing	V1	0.474M	0.691	1.428	0.055	0.002	0.051	0.002	0.354	0.241	0.401	0.248	0.643
Label Smoothing	V2	0.474M	0.558	2.107	0.102	0.006	0.047	0.010	0.512	0.111	0.368	0.283	0.563
A. L.S.	V1	0.474M	0.655	2.121	0.191	0.003	0.186	0.003	0.461	0.020	0.255	0.401	0.480
A. L.S.	V2	0.474M	0.532	2.839	0.185	0.011	0.158	0.012	0.661	0.071	0.217	0.441	0.399
Hard Label (mask)	V1	0.528M	0.680	1.451	0.088	0.003	0.076	0.004	0.259	0.025	0.549	0.132	0.766
Hard Label (mask)	V2	0.528M	0.559	2.194	0.155	0.007	0.127	0.011	0.686	0.106	0.507	0.163	0.691
CutMix (mask)	V1	0.528M	0.698	1.326	0.032	0.002	0.020	0.003	0.249	0.128	0.477	0.197	0.704
CutMix (mask)	V2	0.528M	0.576	1.999	0.110	0.008	0.067	0.008	0.614	0.100	0.449	0.228	0.635
Label Smoothing (mask)	V1	0.528M	0.687	1.447	0.051	0.002	0.046	0.003	0.430	0.311	0.407	0.244	0.647
Label Smoothing (mask)	V2	0.528M	0.563	2.135	0.108	0.005	0.048	0.007	0.524	0.074	0.374	0.281	0.568
A. L.S. (mask)	V1	0.528M	0.648	2.176	0.186	0.002	0.182	0.002	0.463	0.038	0.246	0.396	0.478
A. L.S. (mask)	V2	0.528M	0.528	2.914	0.185	0.005	0.160	0.006	0.687	0.074	0.209	0.441	0.394
A. L.S. (mask) (beta =0.75)	V1	0.528M	0.681	1.759	0.146	0.004	0.142	0.003	0.377	0.020	0.319	0.337	0.553
A. L.S. (mask) (beta =0.75)	V2	0.528M	0.556	2.478	0.146	0.009	0.113	0.013	0.572	0.119	0.274	0.375	0.469
A. L.S. (mask) (beta =0.25)	V1	0.528M	0.684	1.479	0.059	0.003	0.052	0.004	0.244	0.025	0.402	0.244	0.645
A. L.S. (mask) (beta =0.25)	V2	0.528M	0.561	2.191	0.109	0.009	0.059	0.009	0.627	0.264	0.369	0.285	0.563
A. L.S. + Context (mask)	V1	0.528M	0.637	2.197	0.174	0.004	0.169	0.004	0.431	0.023	0.251	0.390	0.480
A. L.S. + Context (mask)	V2	0.528M	0.515	2.954	0.177	0.009	0.147	0.007	0.682	0.069	0.221	0.437	0.397
A. L.S. + CutMix (mask)	V1	0.528M	0.442	4.569	0.349	0.004	0.332	0.004	0.559	0.029	0.047	0.843	0.095
A. L.S. + CutMix (mask)	V2	0.528M	0.346	4.952	0.292	0.011	0.265	0.012	0.902	0.070	0.049	0.851	0.083



Visualization of the count per each of the 1000 classes in the 'mask' version of ImageNet used by our approach.

482	256	484	56	320	82	436	42	470
	40000	40000	40000	40000	40000	40000	40000	40000
48	427	413	344	238	31	176	380	472
182	40000	40000	40000	39708	38947	29112	27922	25569
392	180	398	203	372	302	337	307	101
381	183	461	96	119	89	408	59	74
40	121	135	329	241	79	72	150	260
280	27	38	71	154	47	284	12	377
202	196	360	474	285	439	368	246	322
100	437	102	161	426	444	418	104	406
191	449	235	70	384	504	300	1	411

Visualization of the count per each of the 480 classes in the 'mask' version of OpenImages used by our approach. Class '256' for example, has 40k images.

Figure 4.6: Top half of the figure shows the count per class for the ImageNet dataset, the highest number of images in a given class is '1349' and the lowest count is '190'. The distribution in this case is not as skewed as the OpenImages (bottom half) dataset. About 60 classes in our subset of the OpenImages dataset account for half the dataset. The maximum and minimum counts are 55K and 28K respectively.

Table 4.3: Classification and calibration results with OpenImages. For a detailed explanation of the metrics please refer to the ‘Experimental setup’ section. ‘A.conf’, ‘O.conf’ and ‘U.conf’ refer to average confidence, overconfidence, and underconfidence scores. We provide ECE values for 100 bins and 15 bins mean scores along with their standard deviation (std).

Method	Val./Test size	Val. Set	Acc. mean	Log-loss mean	ECE 100 mean	ECE 100 std	ECE 15 mean	ECE 15 std	MCE mean	MCE std	O.conf mean	U.conf mean	A.conf mean
Hard Label (mask)	105978	Val	0.552	1.519	0.089	0.003	0.080	0.004	0.280	0.029	0.476	0.235	0.636
Hard Label (mask)	325098	Test	0.549	1.522	0.089	0.002	0.083	0.002	0.262	0.073	0.479	0.238	0.634
Label Smoothing (mask)	105978	Val	0.554	1.573	0.044	0.003	0.032	0.004	0.220	0.061	0.410	0.312	0.564
Label Smoothing (mask)	325098	Test	0.550	1.577	0.033	0.002	0.029	0.002	0.196	0.148	0.408	0.315	0.561
A. L.S. (mask)	105978	Val	0.392	4.725	0.389	0.008	0.372	0.008	0.779	0.117	0.032	0.908	0.055
A. L.S. (mask)	325098	Test	0.388	4.749	0.346	0.004	0.328	0.004	0.579	0.018	0.031	0.912	0.053
A. L.S. (mask) + Context	105978	Val	0.383	4.049	0.219	0.005	0.203	0.005	0.464	0.028	0.092	0.626	0.200
A. L.S. (mask) + Context	325098	Test	0.371	4.092	0.193	0.003	0.178	0.002	0.415	0.022	0.089	0.624	0.195
A. L.S. (mask) (beta =0.25)	105978	Val	0.556	1.667	0.058	0.003	0.051	0.003	0.226	0.094	0.371	0.362	0.519
A. L.S. (mask) (beta =0.25)	325098	Test	0.554	1.670	0.052	0.002	0.049	0.002	0.127	0.010	0.370	0.364	0.517

Table 4.4: Fine-tuning on MS-COCO using FRCNN for object detection. For a detailed explanation of the results please refer to the ‘Experimental setup’ section. AP refers to average precision and AR refers to average recall at the specified Intersection over union (IoU) level. We also provide AP values for small, medium, and large objects using ‘S’, ‘M’, and ‘L’ respectively.

Method	Pre-train dataset	Pre-train size	AP 0.5:0.95	AP 0.5	AP 0.75	AP (S) 0.5:0.95	AP (M) 0.5:0.95	AP (L) 0.5:0.95	AR 0.5:0.95
Hard Label	ImageNet	1.28M	0.323	0.519	0.345	0.136	0.367	0.481	0.438
CutMix	ImageNet	1.28M	0.329	0.528	0.353	0.139	0.376	0.490	0.445
RICAP	ImageNet	1.28M	0.331	0.528	0.354	0.138	0.376	0.493	0.447
Hard Label	ImageNet	0.474M	0.290	0.479	0.309	0.112	0.325	0.437	0.415
Adaptive L.S.	ImageNet	0.474M	0.311	0.501	0.332	0.119	0.352	0.470	0.429
Hard Label (mask)	ImageNet	0.528M	0.290	0.482	0.307	0.114	0.329	0.435	0.415
CutMix (mask)	ImageNet	0.528M	0.312	0.509	0.329	0.125	0.353	0.470	0.428
Label Smoothing (mask)	ImageNet	0.528M	0.304	0.500	0.324	0.122	0.346	0.455	0.424
Adaptive L.S. (mask)	ImageNet	0.528M	0.311	0.501	0.333	0.124	0.351	0.477	0.428
Adaptive L.S. (mask) (beta =0.75)	ImageNet	0.528M	0.309	0.498	0.331	0.123	0.348	0.467	0.427
Adaptive L.S. (mask) (beta =0.25)	ImageNet	0.528M	0.298	0.492	0.315	0.122	0.340	0.449	0.419
Adaptive L.S. + Context (mask)	ImageNet	0.528M	0.303	0.490	0.323	0.115	0.339	0.465	0.421
Adaptive L.S. + CutMix (mask)	ImageNet	0.528M	0.273	0.449	0.289	0.098	0.300	0.423	0.403
Hard Label (mask)	OpenImages	1.20M	0.295	0.484	0.313	0.115	0.330	0.453	0.416
Label Smoothing (mask)	OpenImages	1.20M	0.301	0.493	0.320	0.119	0.339	0.457	0.420
Adaptive L.S. (mask)	OpenImages	1.20M	0.243	0.415	0.250	0.083	0.263	0.376	0.371
Adaptive L.S. + Context (mask)	OpenImages	1.20M	0.289	0.471	0.308	0.111	0.321	0.448	0.408
Adaptive L.S. (mask) (beta =0.25)	OpenImages	1.20M	0.304	0.494	0.324	0.118	0.340	0.462	0.422

#### 4.4.4 Classification and calibration

This section identifies various calibration metrics used by the community and discusses our results obtained on the popular [57, 60] datasets. We use the implementation of [71] on all of our classifiers to generate the results in table 4.2. To evaluate the performance of *adaptive* label smoothing we use five metrics that are very common: *accuracy*, *expected calibration error* (ECE) [53], *maximum calibration error* (MCE) [53], *overconfidence* [51], and *underconfidence* [51]. We computed ECE using 100 bins and 15 bins. The authors of [34, 71] discuss the advantages of using 100 bins in greater detail.

ECE is defined as the expected absolute difference between a classifier’s confidence and its accuracy using a finite number of bins [53]. MCE is defined as the maximum absolute difference between a classifier’s confidence and its accuracy of each bin [53]. Overconfidence is the average confidence of a classifier’s false predictions and underconfidence as the average uncertainty on its correct predictions [51, 71].

The results in table 4.2 indicate our approaches based on *adaptive* label smoothing using the abbreviation ‘A. L. S.’ In general, these results have a low overconfidence score, which is highly desirable. These results demonstrate that adaptive label smoothing based CNNs seldom produce high confidence scores when they make incorrect predictions. In fact, our models are under confident as they pay attention to the spatial footprint of the pertinent object. It is important to note that our methods outperform all baselines for the overconfidence metric.

#### 4.4.5 Transfer learning for object detection

We adopt the architecture of Faster RCNN [58] adapted to use the ResNet-50 backbone. Specifically, we train all of our classifiers using the implementation of <https://github.com/jwyang/faster-rcnn.pytorch>. We train all ImageNet pre-trained models with a batch size of 16 and initial

learning rate of 0.01 decayed after every 4 epochs for a total of 10 epochs. We train all OpenImages pre-trained models with a batch size of 16 and initial learning rate of 0.0075 decayed after every 4 epochs for a total of 10 epochs. We employ the standard metrics for average precision (AP) and average recall [44] at different intersection over union (IoU) levels. As shown in table 4.2, our approach outperforms hard label and label smoothing based approaches on this downstream task. The better localization performance is also shown without fine-tuning using class activation maps in figure 4.2. Specifically, our approach performs almost as well as CutMix [74] using AP measures.

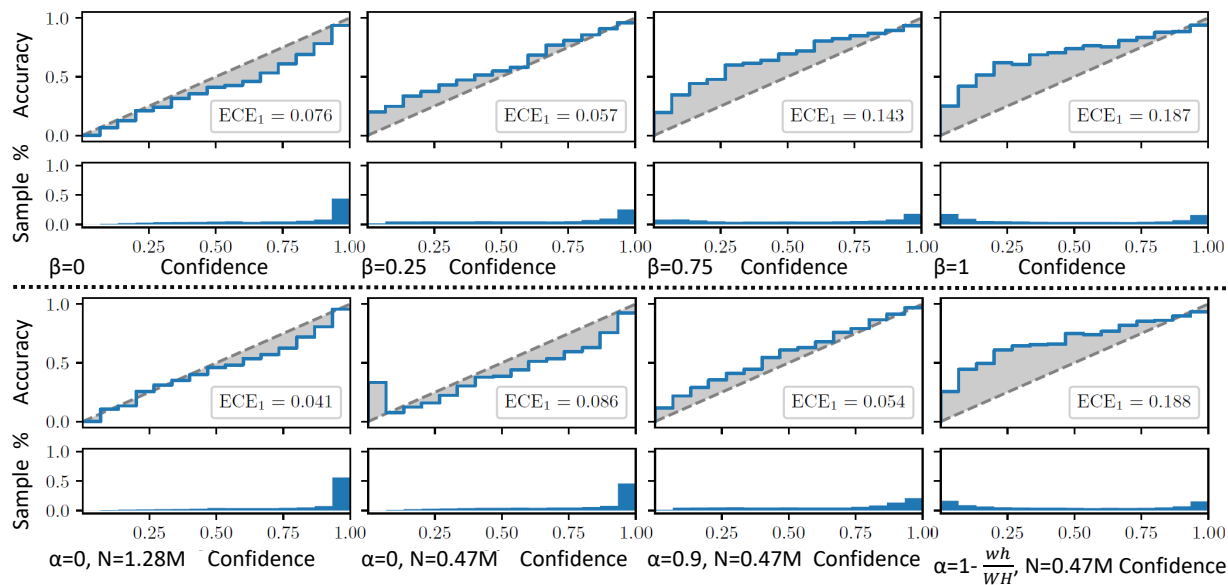


Figure 4.7: Reliability diagrams help understand the calibration performance [12, 54] of classifiers. We compute  $ECE_1$  using the implementation of [71] on the validation set of ImageNet. The deviation from the dashed line (shown in gray), weighted by the histogram of confidence values, is equal to Expected Calibration Error [71]. The top half of the figure shows classifiers trained using the same dataset ( $N=0.528M$ ), but with different values of  $\beta$ . The leftmost reliability diagram is the classic hard label setting and the rightmost reliability diagram is the *adaptive* label setting. The bottom half of the figure compares classifiers trained on the complete ImageNet (leftmost) with 3 classifiers trained on the subset of ImageNet with bounding box labels using different values of the  $\alpha$  hyperparameter.

### 4.4.6 Ablation studies

We compare our approach with standard baselines and provide results in an ablative manner to understand the benefits and limitations of applying *adaptive* label smoothing to classification and transfer learning for object detection tasks. As shown in figure 4.7, increasing the value of  $\beta$  helps reduce model overconfidence and produces predictions that are less ‘peaky’ compared to label smoothing and hard label settings. Another interesting trend can be observed by changing the value of the  $\beta$  hyperparameter. As  $\beta$  decreases in value, the overconfidence rate goes up along with it as shown in table 4.2. Average confidence of a model describes the mean confidence of a model. As our model predictions are grounded in the spatial size of the object, our average confidence values on ‘V1’ and ‘V2’ are 0.48 and 0.39, respectively; in the case of hard labels the values are 0.77 and 0.69, respectively.

In case of transfer learning, we observe that decreasing  $\beta$  causes the object localization performance to drop. Using implicit object size information helps CNNs localize and detect objects for downstream tasks as well.

## 4.5 Conclusion

This work has addressed the problems of contextual bias and calibration using a novel approach called *adaptive* label smoothing. We believe that our approach addresses significant problems that are associated with current training techniques. In particular, random cropping of images is a common augmentation technique during training of ResNet, but occasionally the crop misses the object entirely. In such a case, the equivalent of a one-hot label is typically provided, with the result that the system is steered toward increased dependence on background (context) portions of the image. We argue that one-hot representations

are too limiting, and our adaptive approach to label smoothing makes it possible for the classifier to avoid overconfidence in many cases. In particular, our approach accomplishes the following: 1) Our labels not only indicate the presence of an object but also tell the classifier the gross proportion of the object in a given image. This implicit regularization guides the classifiers to avoid producing high confidence values when the object pixels are lower in proportion. On the other hand, if a random crop contains mostly object pixels, then the classifier will be encouraged to produce higher-confidence predictions. 2) A traditional classifier will tend to generate decisions with high confidence values even when images containing only background (no objects) are presented. Formally, classifiers often produce overconfident predictions. Overconfidence is particularly a problem for safety critical applications. With our approach, the system is trained to produce lower confidence predictions with out-of-distribution samples or background-only images are presented. Low confidence predictions from our approach are meaningful for rejecting false positives. High confidence approaches are hard to threshold as most predictions have high confidence even when they are wrong. 3) Traditional classifiers “cheat” by relying heavily on context (see RICAP Figure 9 row 1). Although context helps increase computed accuracy for a given dataset, such reliance is not viable for real-world applications. During training, we assume that every class is equiprobable when only background is provided. We show that bounding box information pertaining to objects can be used to compute a smoothing factor *adaptively* during training to improve the localization and calibration performance of CNNs. We use bounding box information for a portion of ImageNet [60] and OpenImages datasets to train 20 different classifiers. We show that our approach can be combined with traditional label smoothing approaches to train CNNs that are calibrated and have better localization performance on the challenging MS-COCO [44] dataset after fine-tuning, compared to approaches that use hard labels or traditional label smoothing approaches. Our labels capture the object proportion in an implicit manner during training, a significantly more challenging task when compared

to training with hard labels. Although our methods do not improve upon the accuracy of traditional label smoothing for the classification task, we show better regularization and calibration performance on the newly released ImageNetV2 [\[57\]](#) dataset.

# Chapter 5

## Out of Distribution Detection

The main contribution of this chapter is that we have developed a novel way to adapt *adaptive* label smoothing that was described in the previous chapter for out of distribution detection. To demonstrate improved out of distribution detection we have trained 14 classifiers and plot the confidence and entropy histograms over the validation samples.

### 5.1 Related Work

Classification convolutional neural networks always output in the space of the learnt classes while predicting the class of a given image regardless of what the image consists of. For example an ImageNet 1-K trained CNN can not say if the given image has no objects that it was trained on if it is provided with an image of a dinosaur (not an ImageNet category) or if the image has the main object cut out of it (context only). To build robustness, many approaches have been proposed [14, 42, 68]. Our goal is to train with images belonging to out of distribution and context using soft labels (a uniform distribution of probability over the set of target classes) for such images. This is a novel way to use soft labels as it allows the model to learn better confidence bounds.

## 5.2 Method

We train classifiers as before in chapter 4, however we validate on novel classes. A good classifier would produce a low confidence/high entropy prediction when presented with images from novel (unseen during training) classes. We use the wordnet embeddings to create an alternative to ImageNet-1K from the larger ImageNet-22K dataset and refer to this dataset as OImageNet-1K. OImageNet-1K consists of about 800K images in training and 50K images in validation sets. We refer to the validation set of ImageNet-1K as ‘In distribution’ and the validation set of OImageNet-1K as ‘Out of distribution’.

We compute entropy based on the information theory definition. Entropy can be used to measure the uncertainty of the output probability distribution. For a dataset consisting of  $K$  classes and output probability distribution denoted by  $p_i$ , we can compute entropy per sample  $x_i$  using the equation below.

$$H(x_i) = - \sum_{i=1}^K p_i \ln p_i \quad (5.1)$$

## 5.3 Experiments

We show the confidence histogram plots for each of the models as this is one the standard approaches for plotting in distribution vs. out of distribution predictions. For the confidence plots, we compute the bincounts for samples having a confidence value greater than 0.25 and for the entropy plots we compute bin counts having entropy under 2 natural units (NATs).

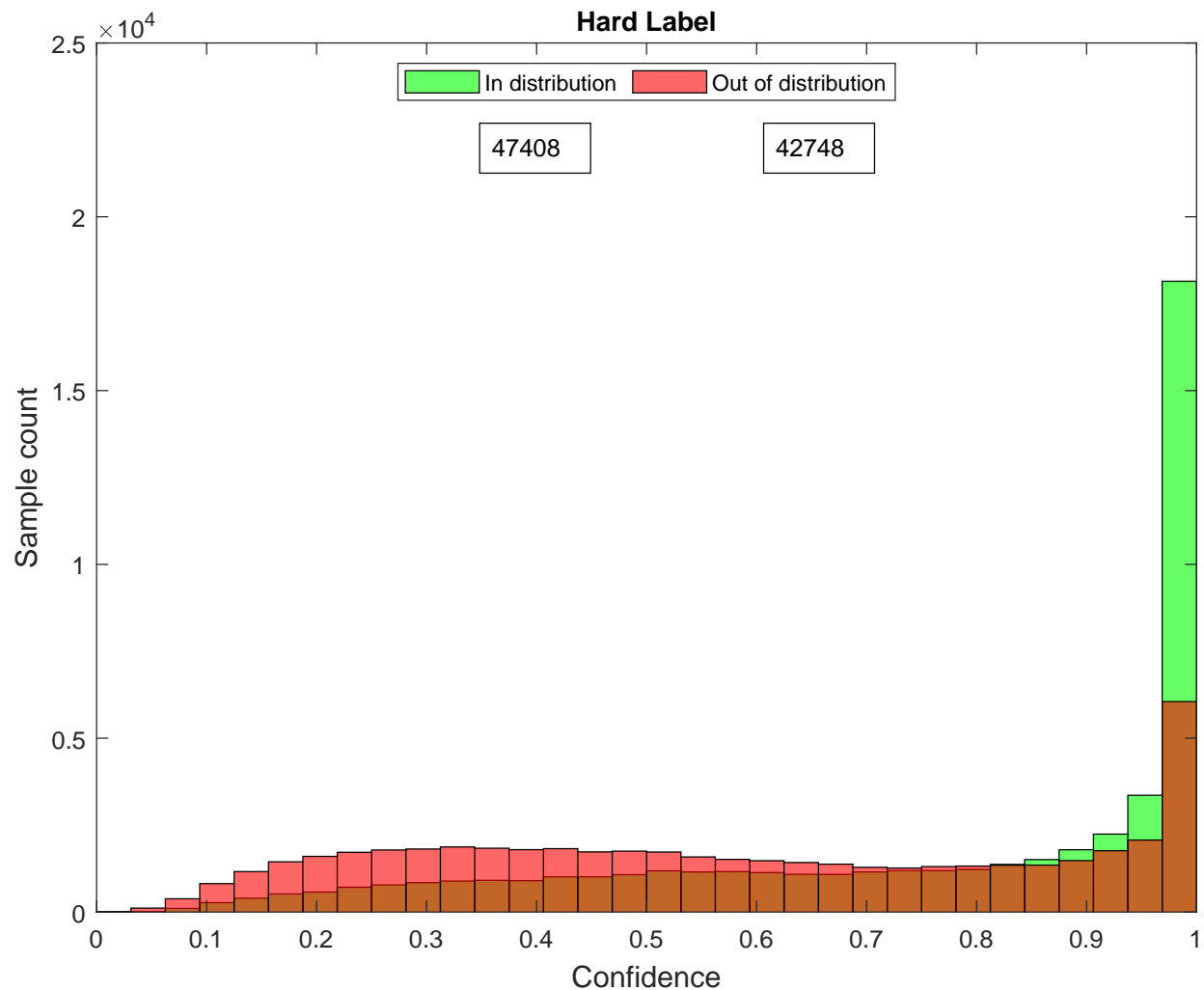


Figure 5.1: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for the hard label case. Clearly, there is no distinct way to threshold the different distributions as they severely overlap with one another.

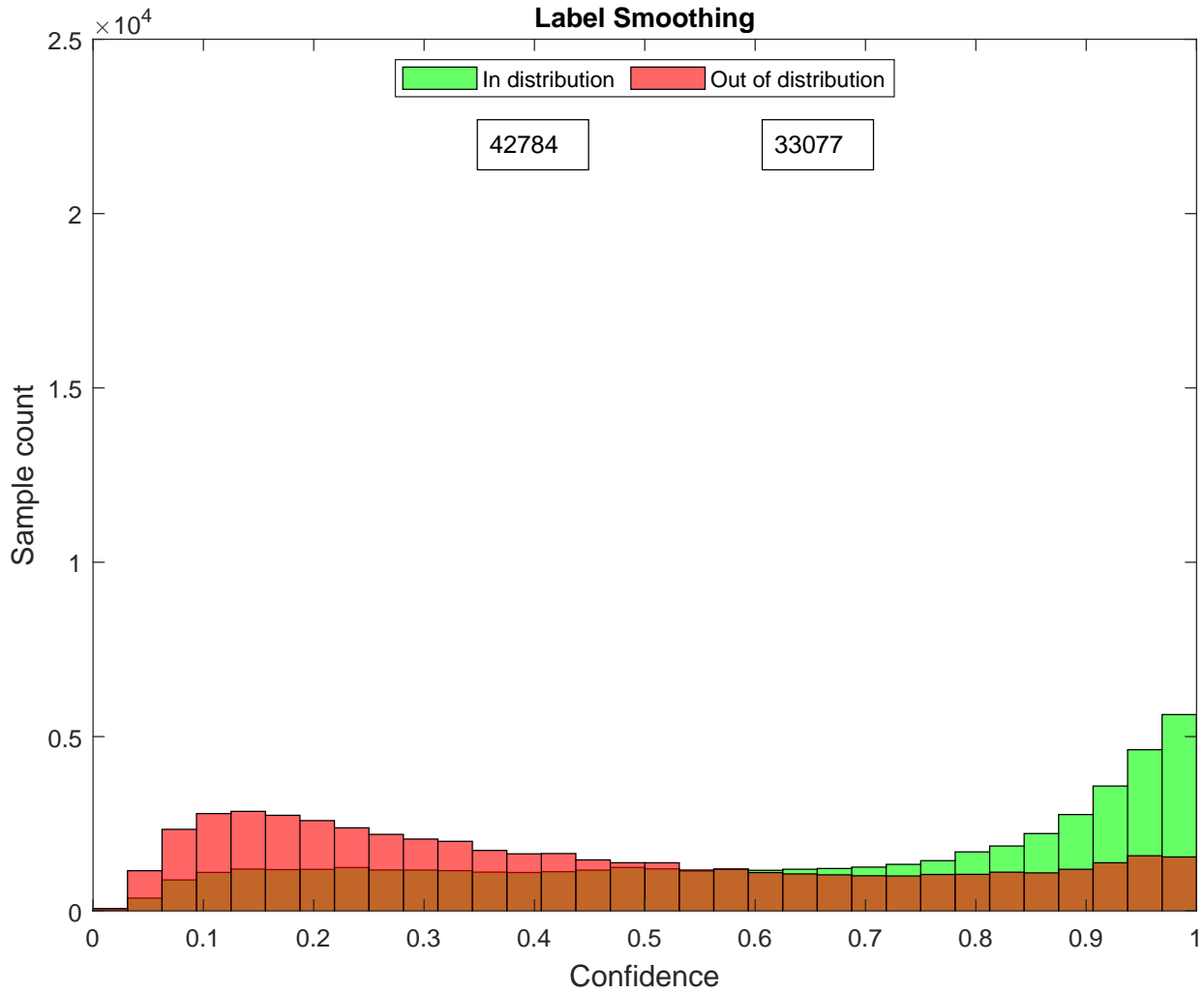


Figure 5.2: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for the standard uniform label smoothing case. The separation is better than the hard label case, but there is plenty of overlap.

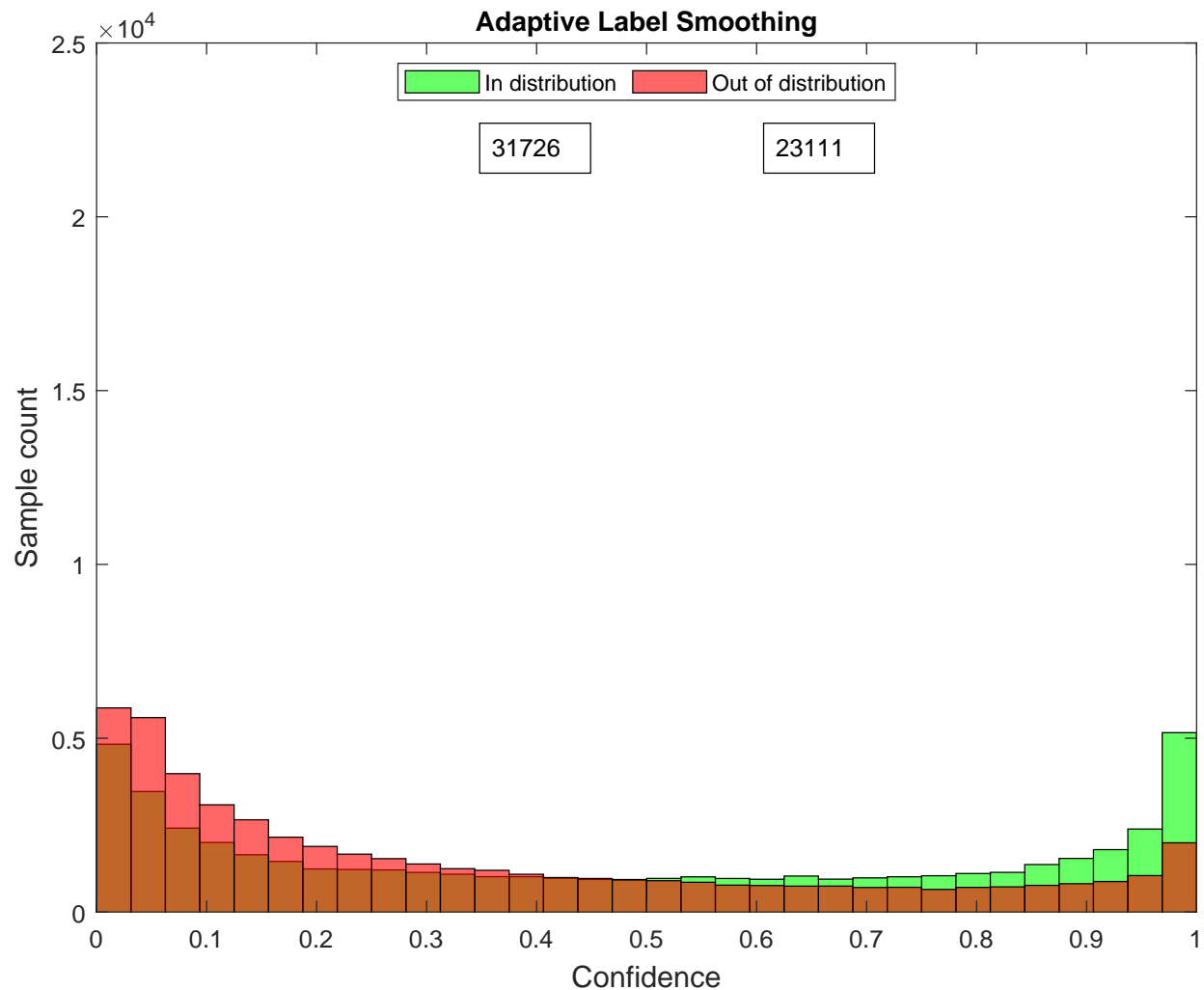


Figure 5.3: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method. The separation is better than the hard label and label smoothing cases, as we produce low confidence scores for a much larger number of out of distribution samples.

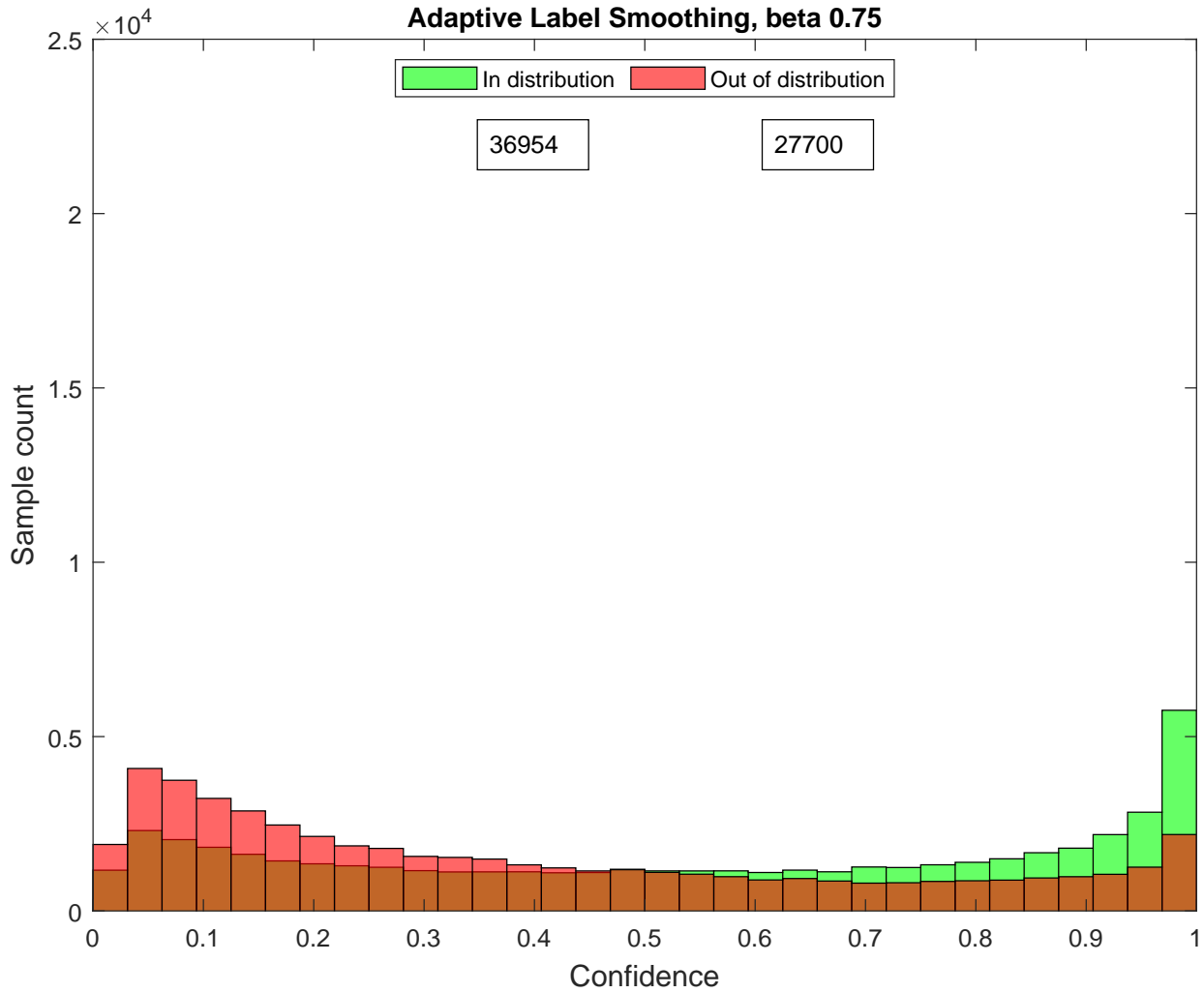


Figure 5.4: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ .

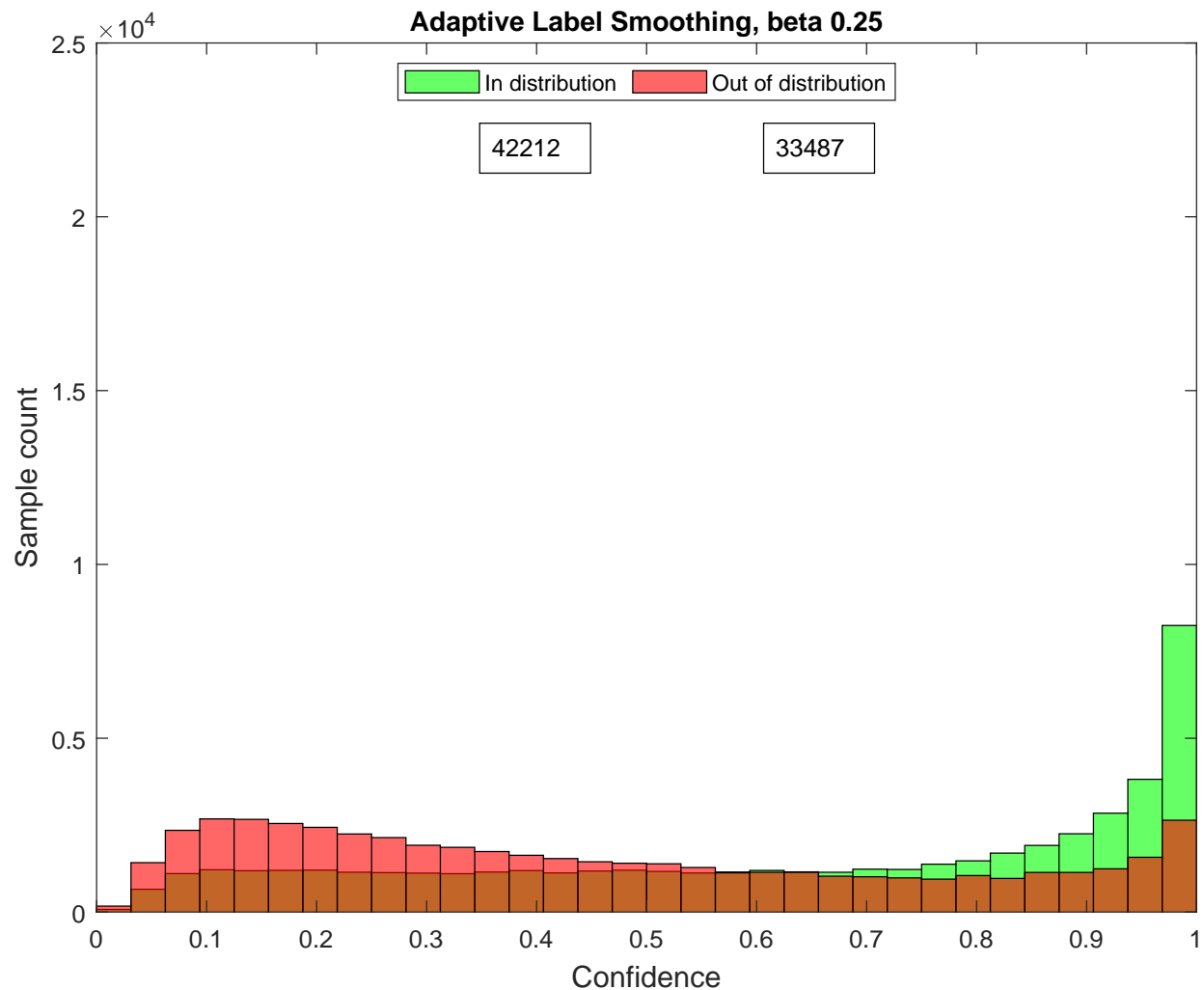


Figure 5.5: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.25$ .

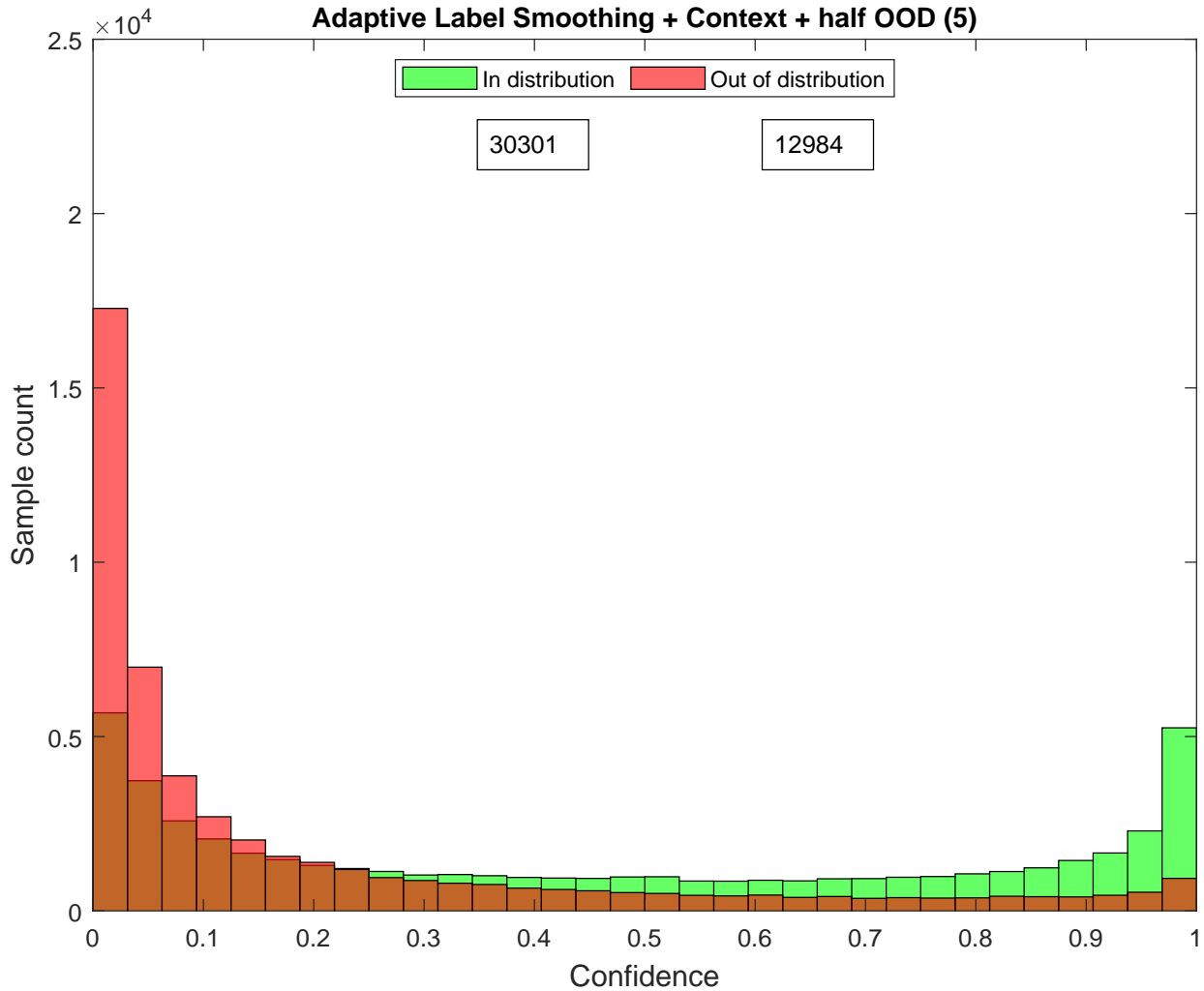


Figure 5.6: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with 5 percent of training samples are context only and 5 percent of samples are from 500 classes of OImageNet training set.

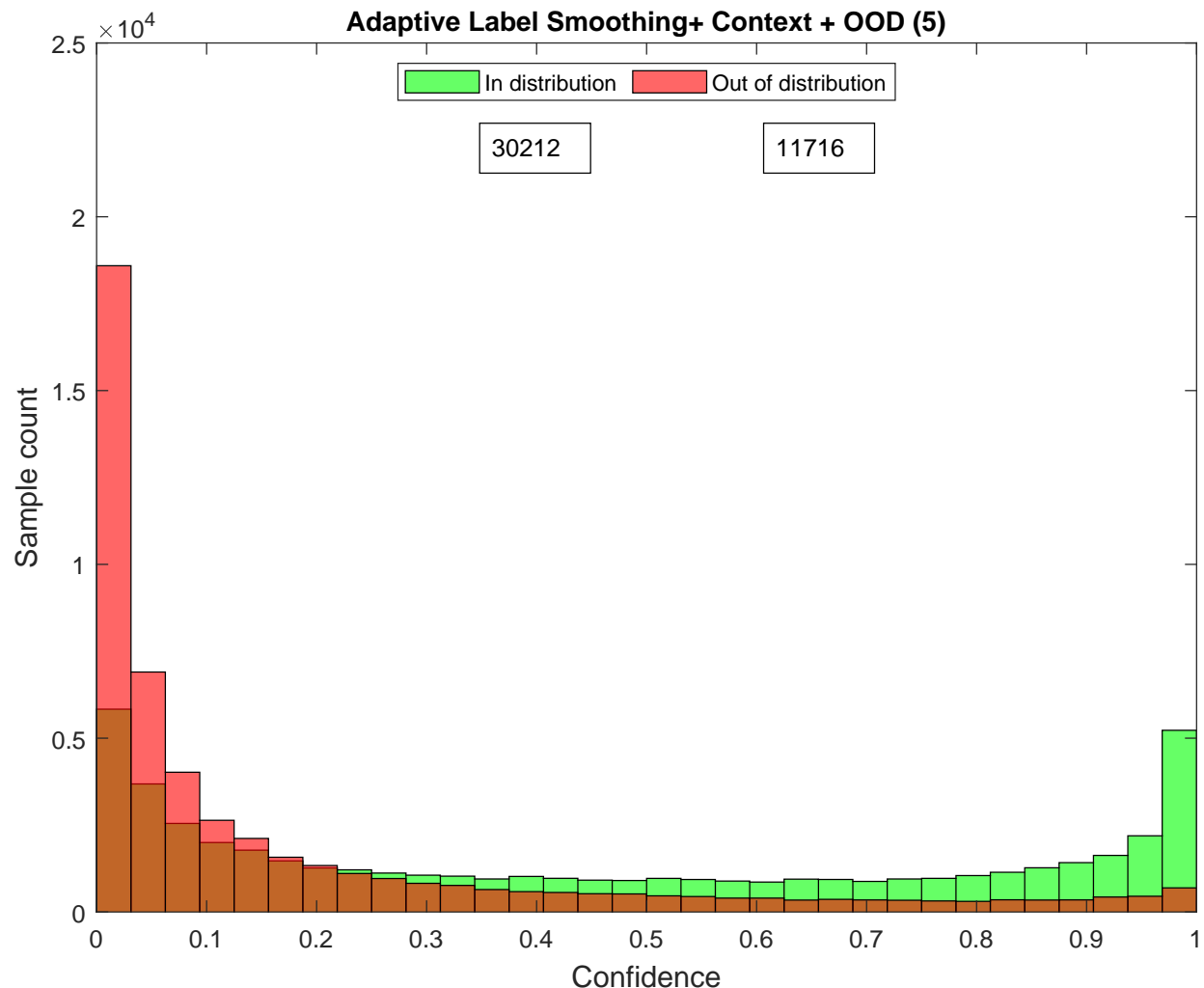


Figure 5.7: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with 5 percent of training samples are context only and 5 percent of samples are from 1000 classes of OImageNet training set.

As shown in the confidence histogram plots [5.1](#), [5.2](#), [5.3](#), [5.4](#), [5.5](#), [5.6](#) and [5.7](#) *adaptive* label smoothing helps identify novel classes even though they were never presented during training. Predictions under a certain threshold can be discarded or communicated to the

end user or a safety critical system using our approach. This helps improve precision for real world applications. We also show similar results with our entropy plots 5.8, 5.9, 5.10, 5.11, 5.12, 5.13 and 5.14.

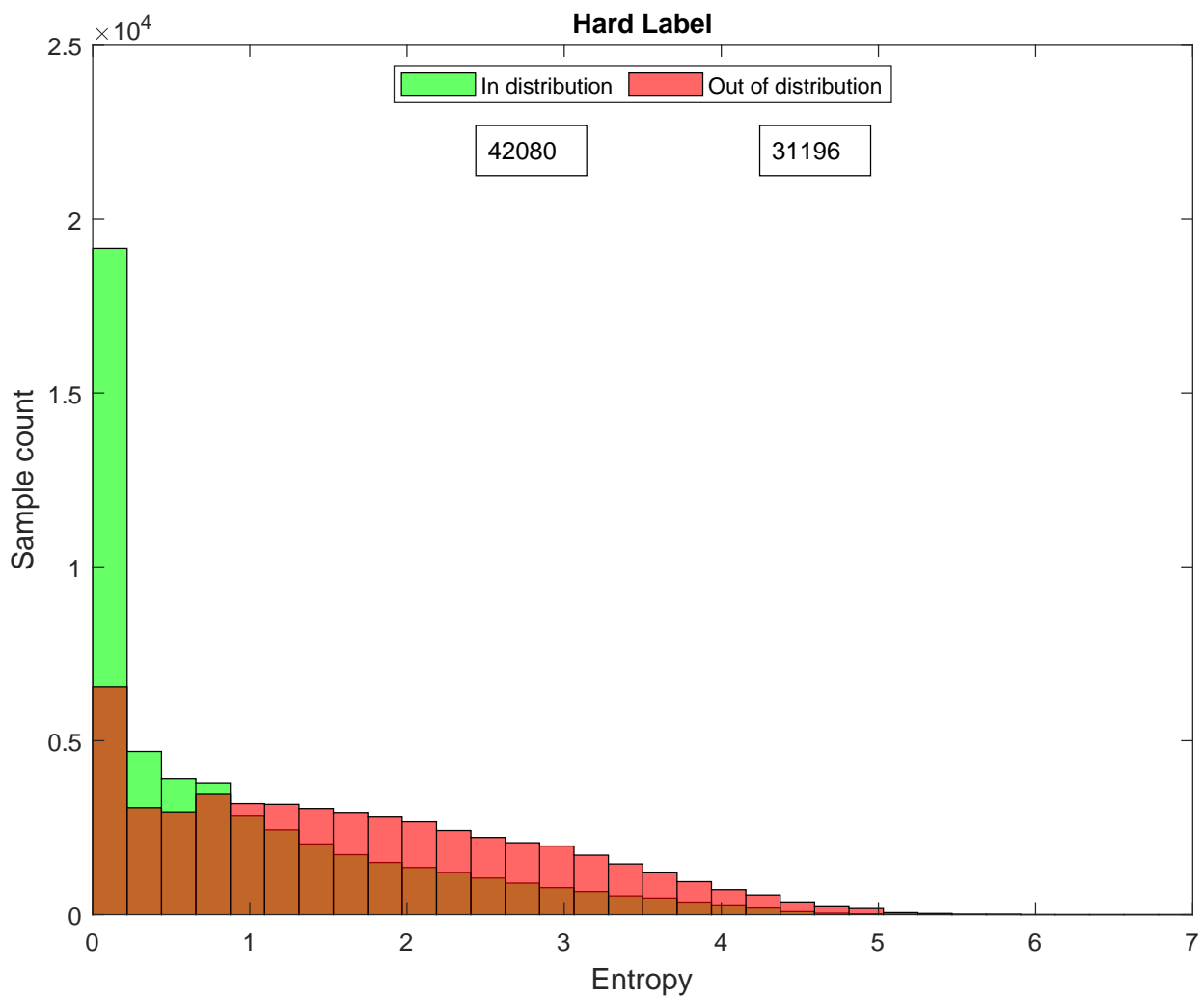


Figure 5.8: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for the hard label case. Clearly, there is no distinct way to threshold the different distributions as they severely overlap with one another.

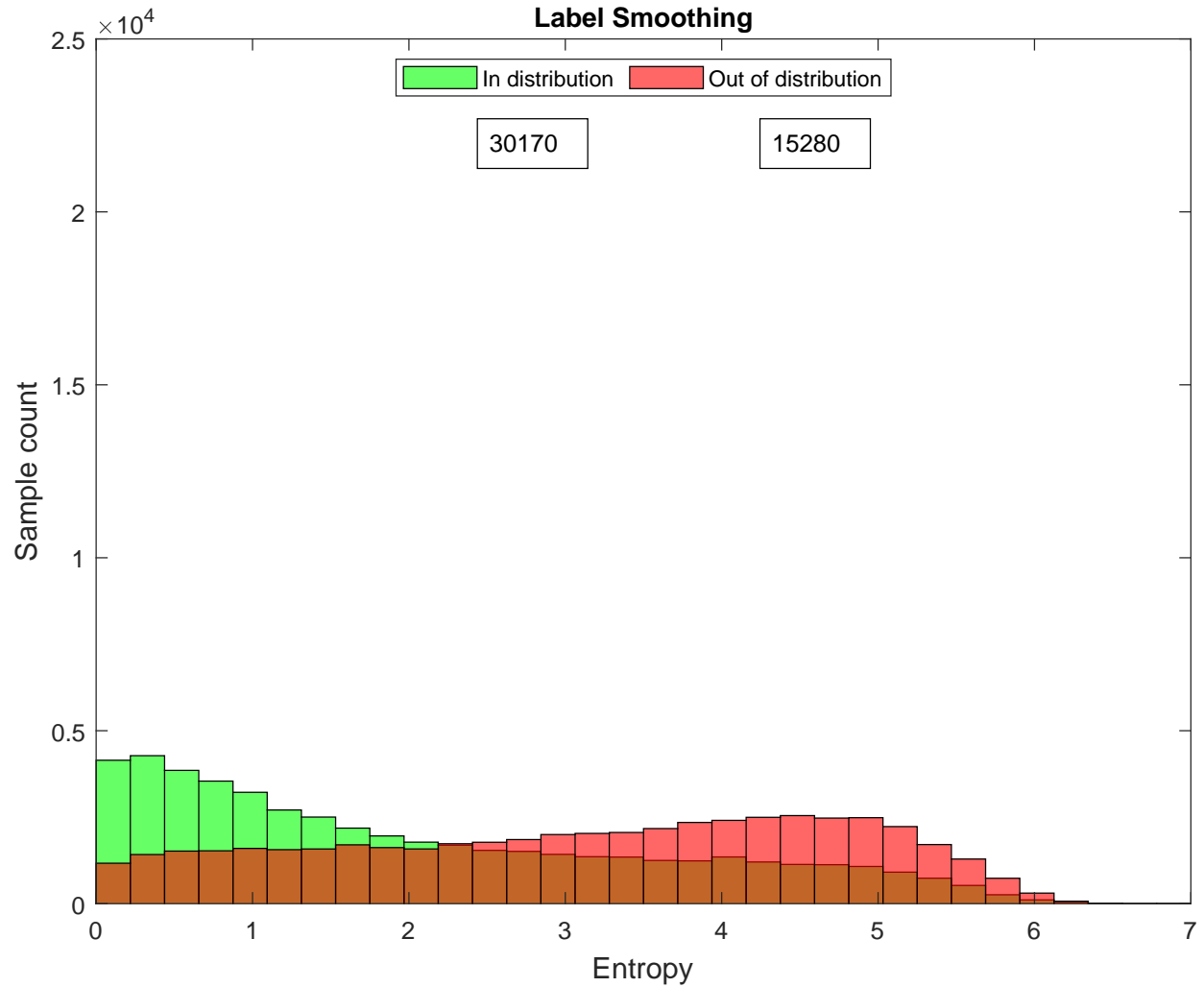


Figure 5.9: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for the standard uniform label smoothing case. The separation is better than the hard label case, but there is plenty of overlap.

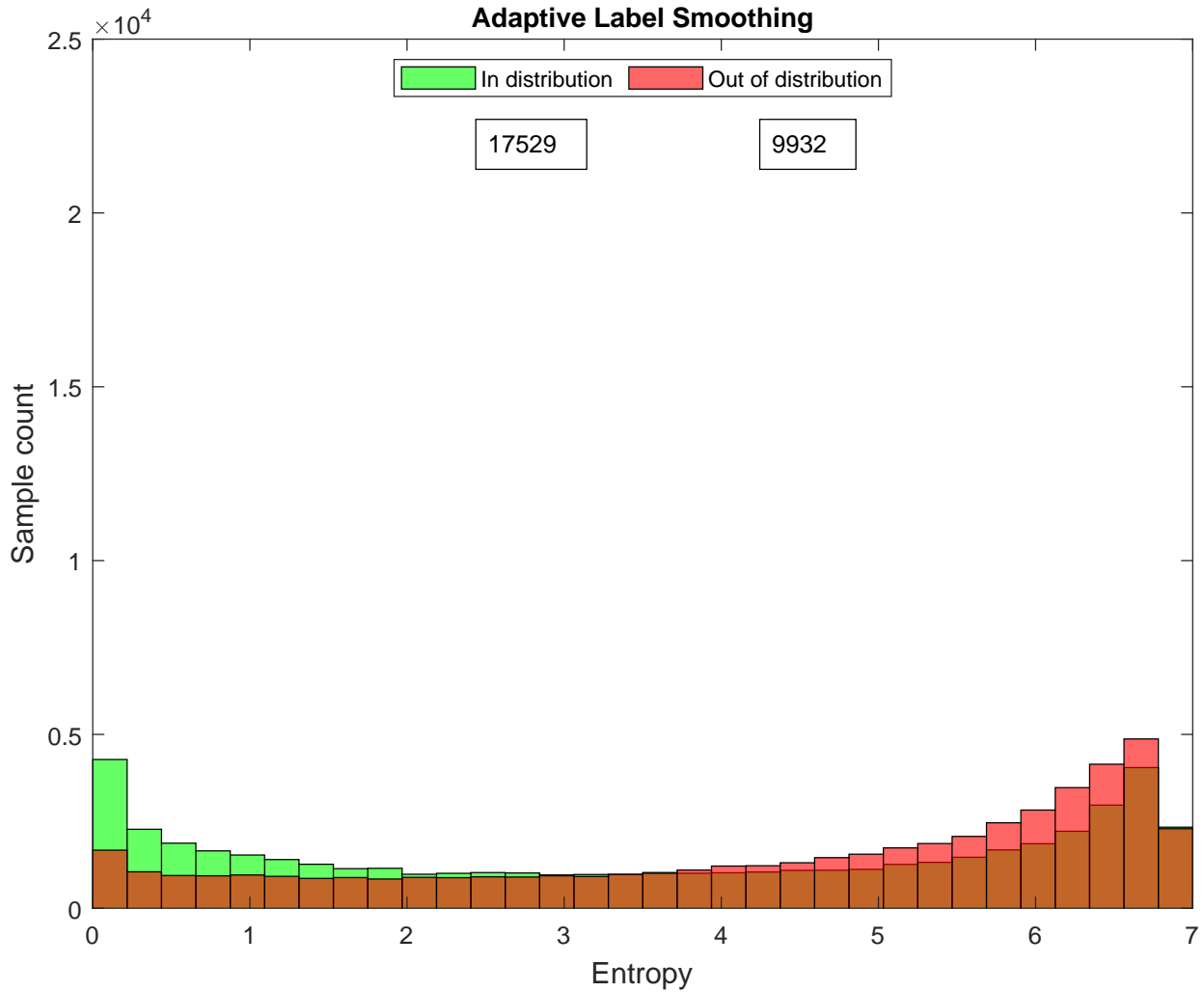


Figure 5.10: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method. The separation is better than the hard label and label smoothing cases, as we produce low Entropy scores for a much larger number of out of distribution samples.

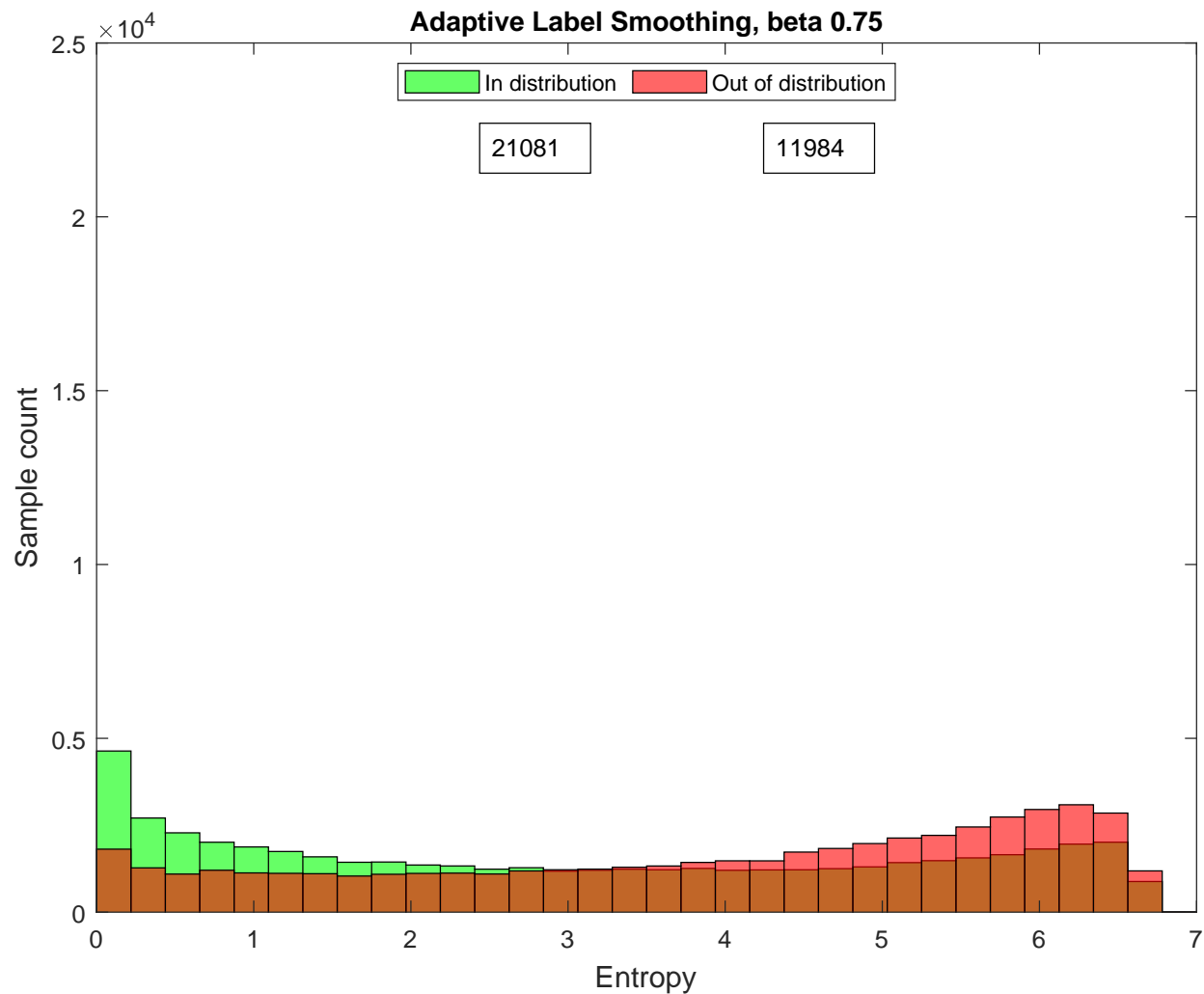


Figure 5.11: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ .

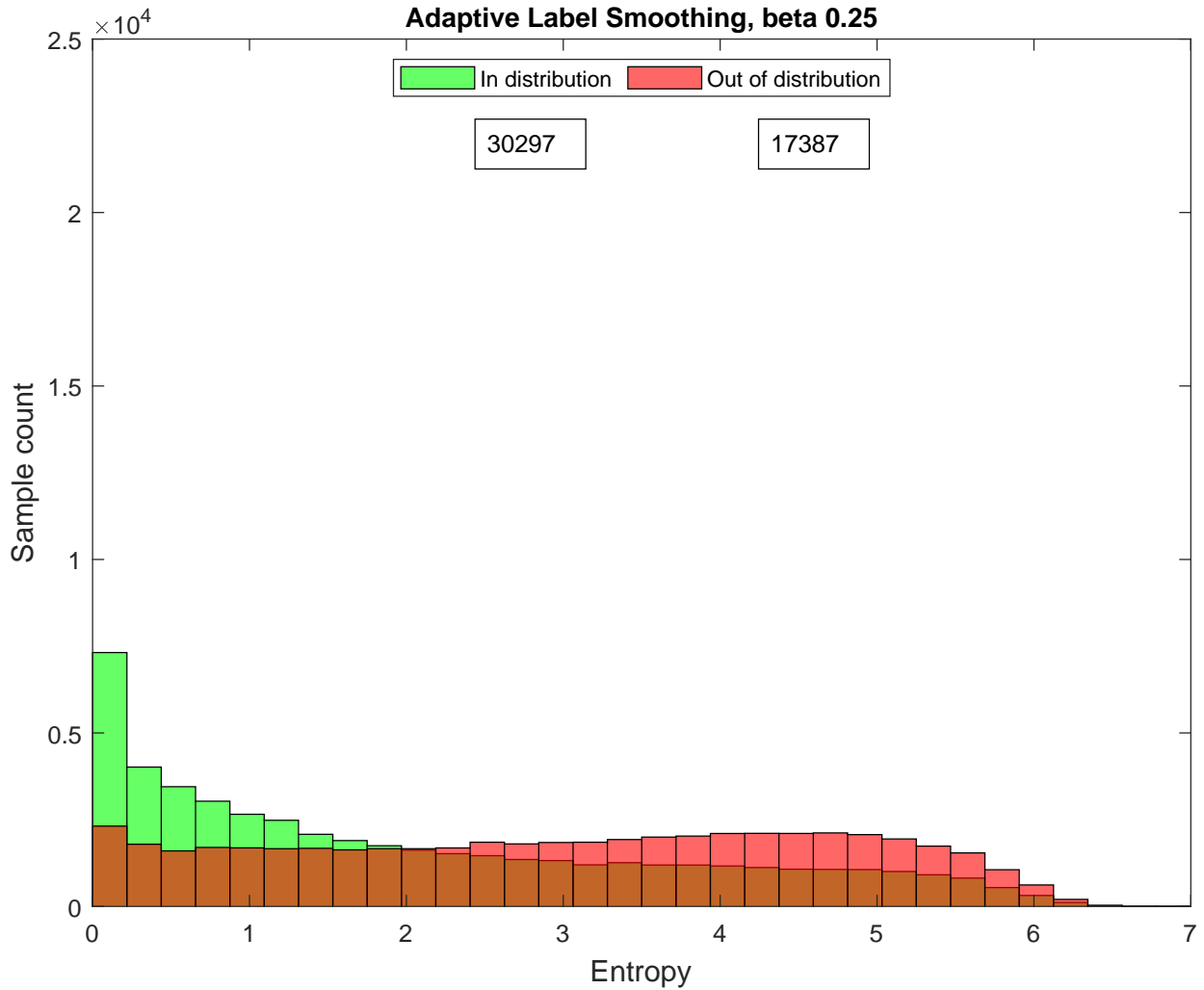


Figure 5.12: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.25$ .

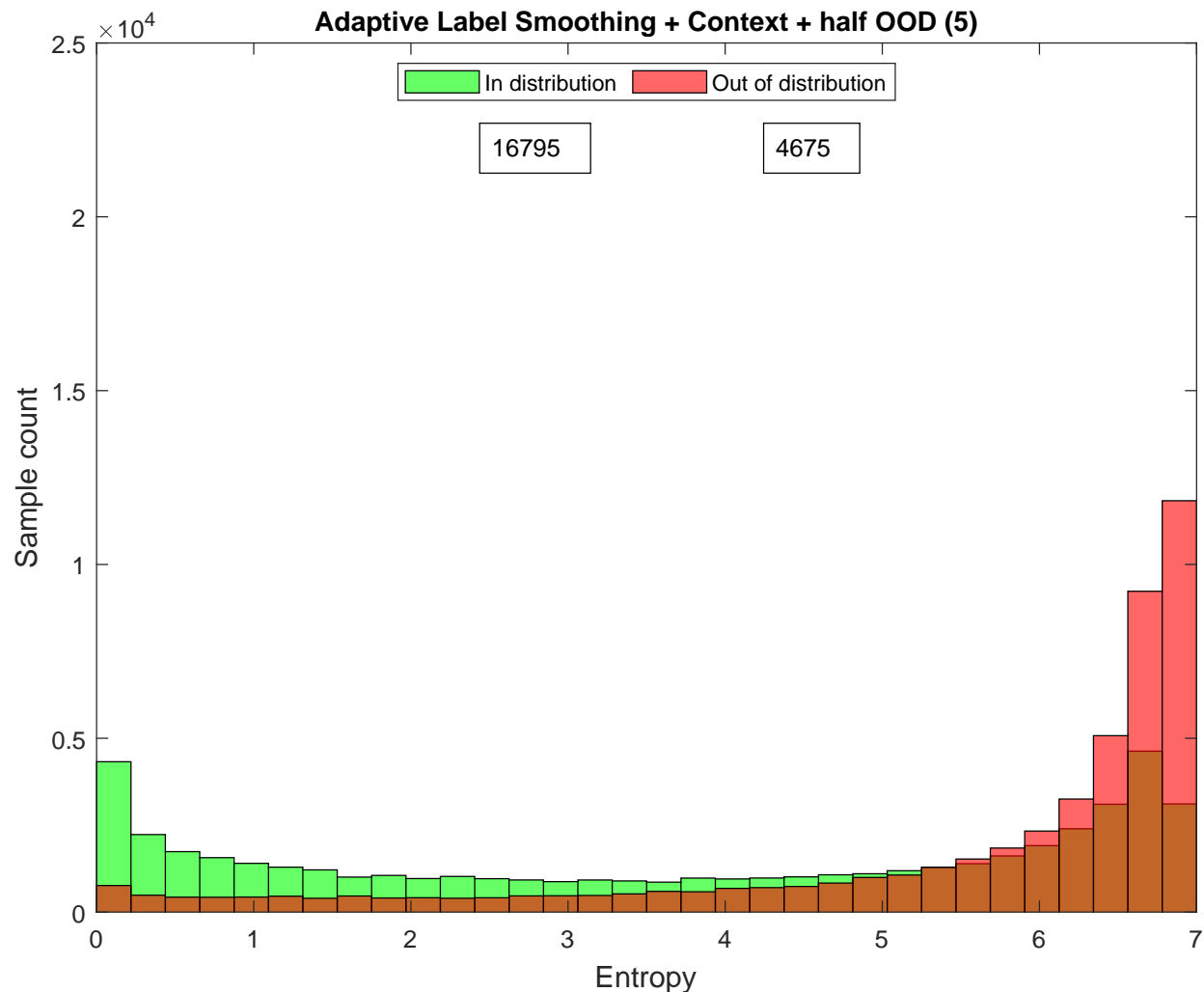


Figure 5.13: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with 5 percent of training samples are context only and 5 percent of samples are from 500 classes of OImageNet training set.

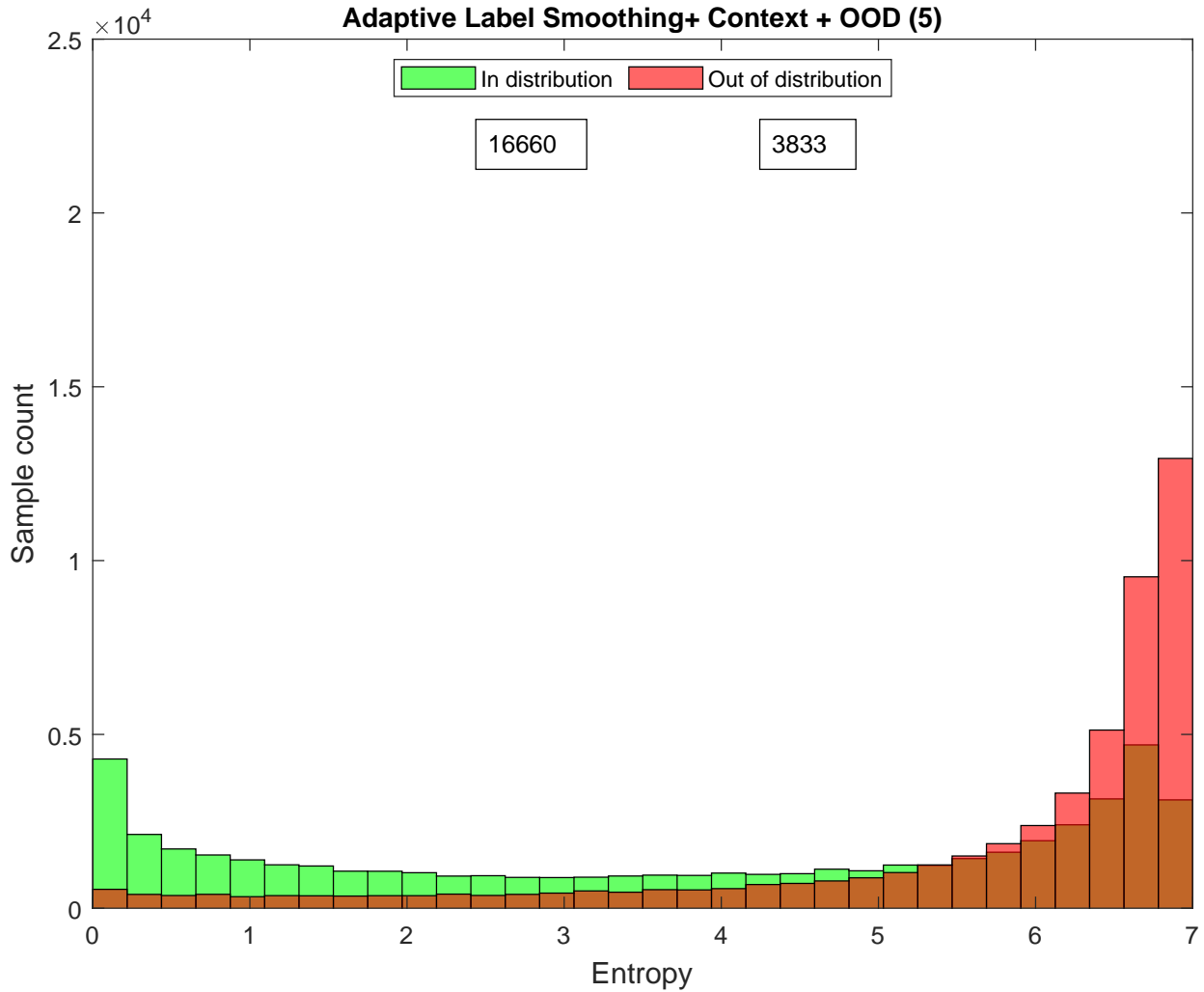


Figure 5.14: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with 5 percent of training samples are context only and 5 percent of samples are from 1000 classes of OImageNet training set.

## 5.4 Results

To further understand the effect of increasing the fraction(5 percent being used in [5.6](#) and [5.7](#)) of out of distribution samples (with uniform smooth labels) we train classifiers with  $\beta = 0.75$  with different amounts of out of distribution samples supplied per training epoch. For the confidence plots, we compute the bincounts for samples having a confidence value greater than 0.25 and for the entropy plots we compute bin counts having entropy under 2 natural units (NATs).

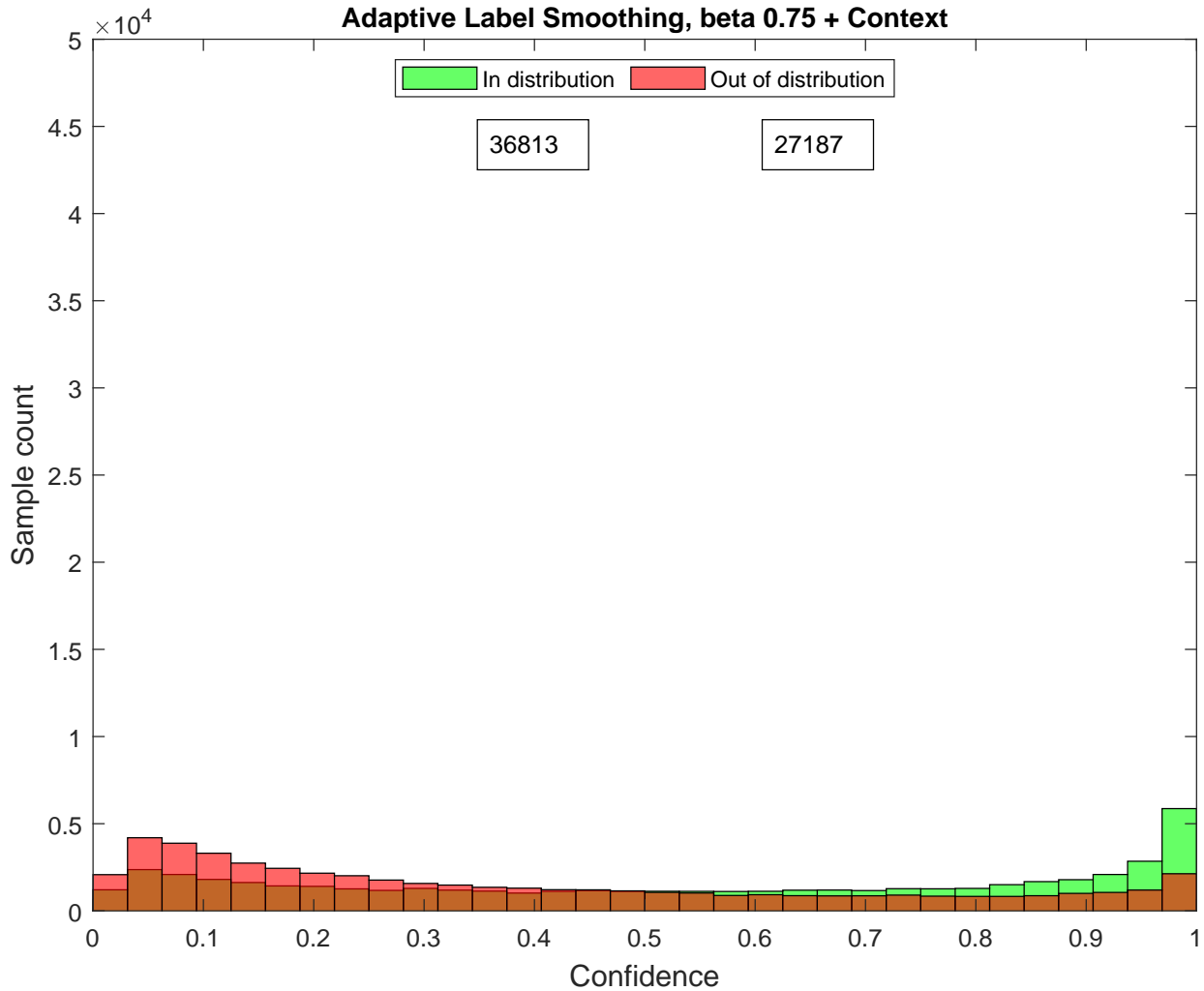


Figure 5.15: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 0 percent of samples are from 1000 classes of OImageNet training set.

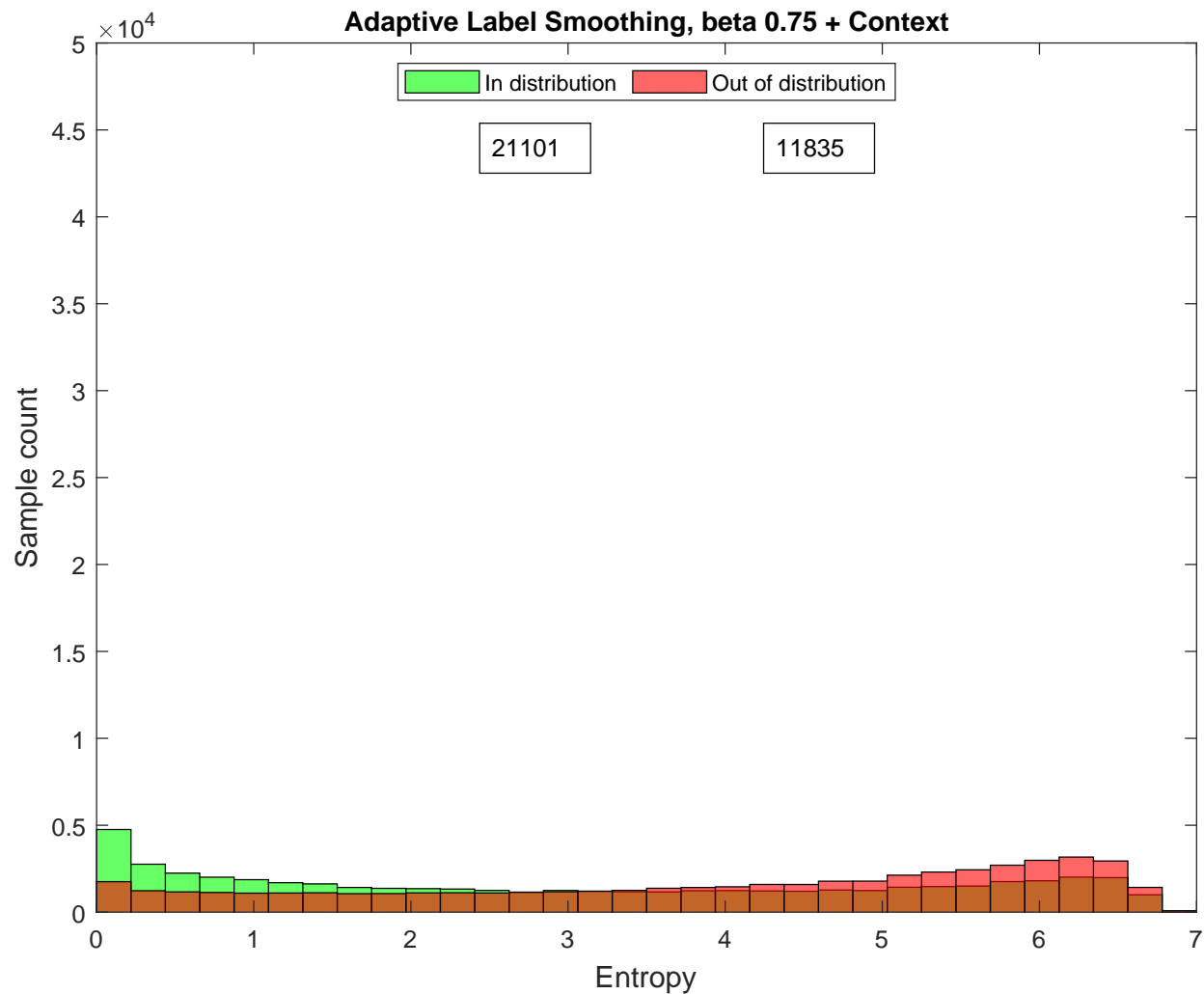


Figure 5.16: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 0 percent of samples are from 1000 classes of OImageNet training set.

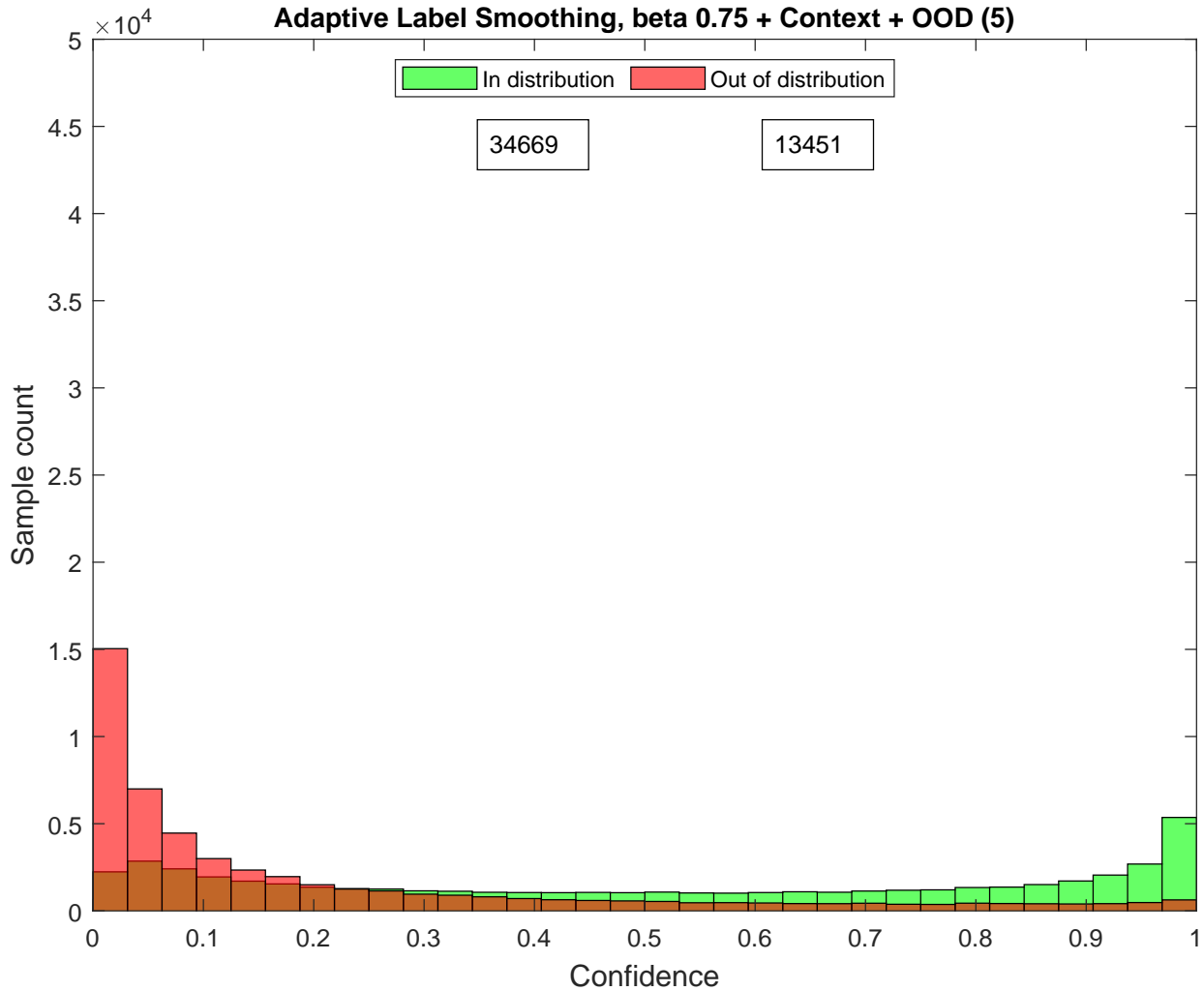


Figure 5.17: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 5 percent of samples are from 1000 classes of OImageNet training set.

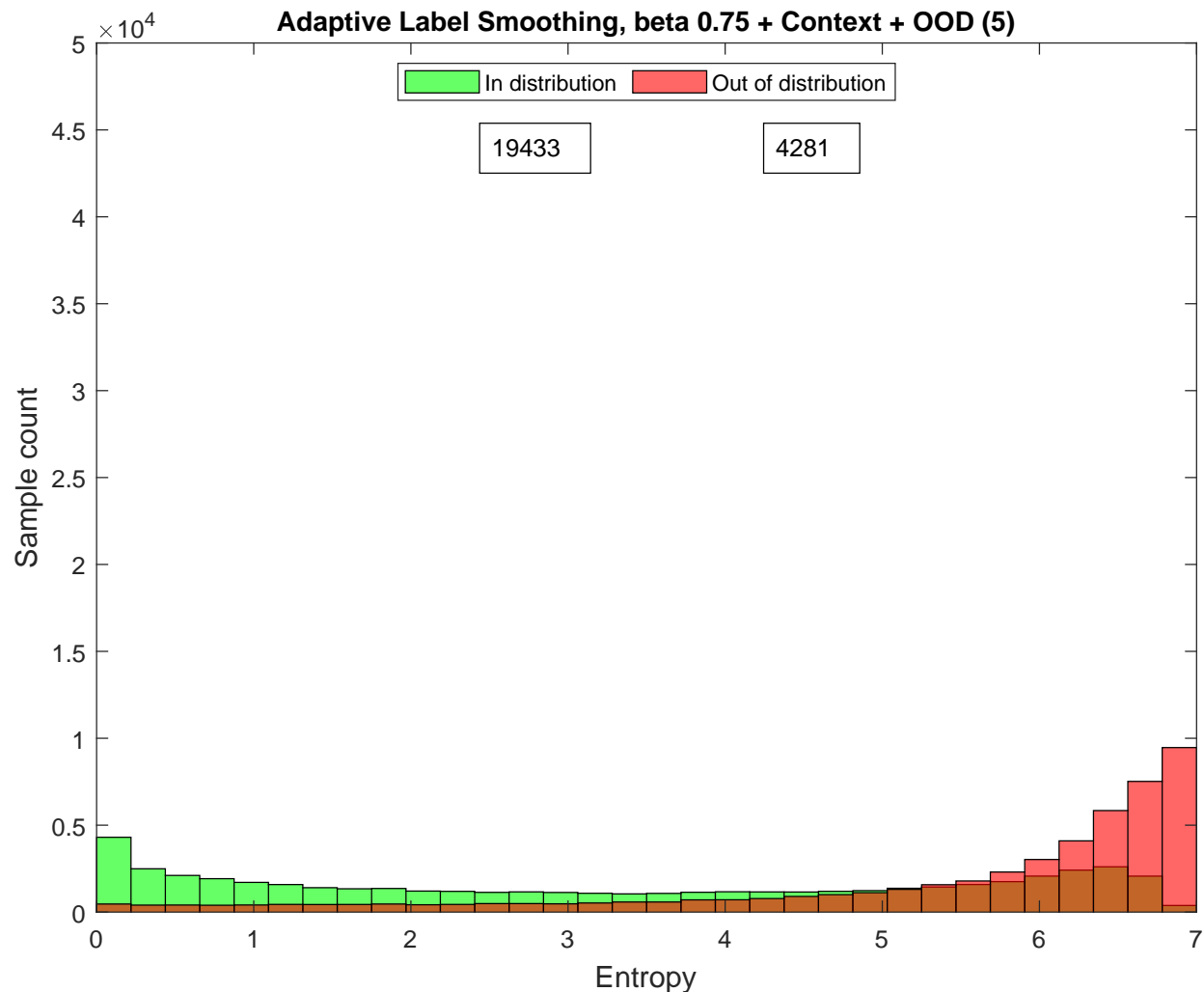


Figure 5.18: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 5 percent of samples are from 1000 classes of OImageNet training set.

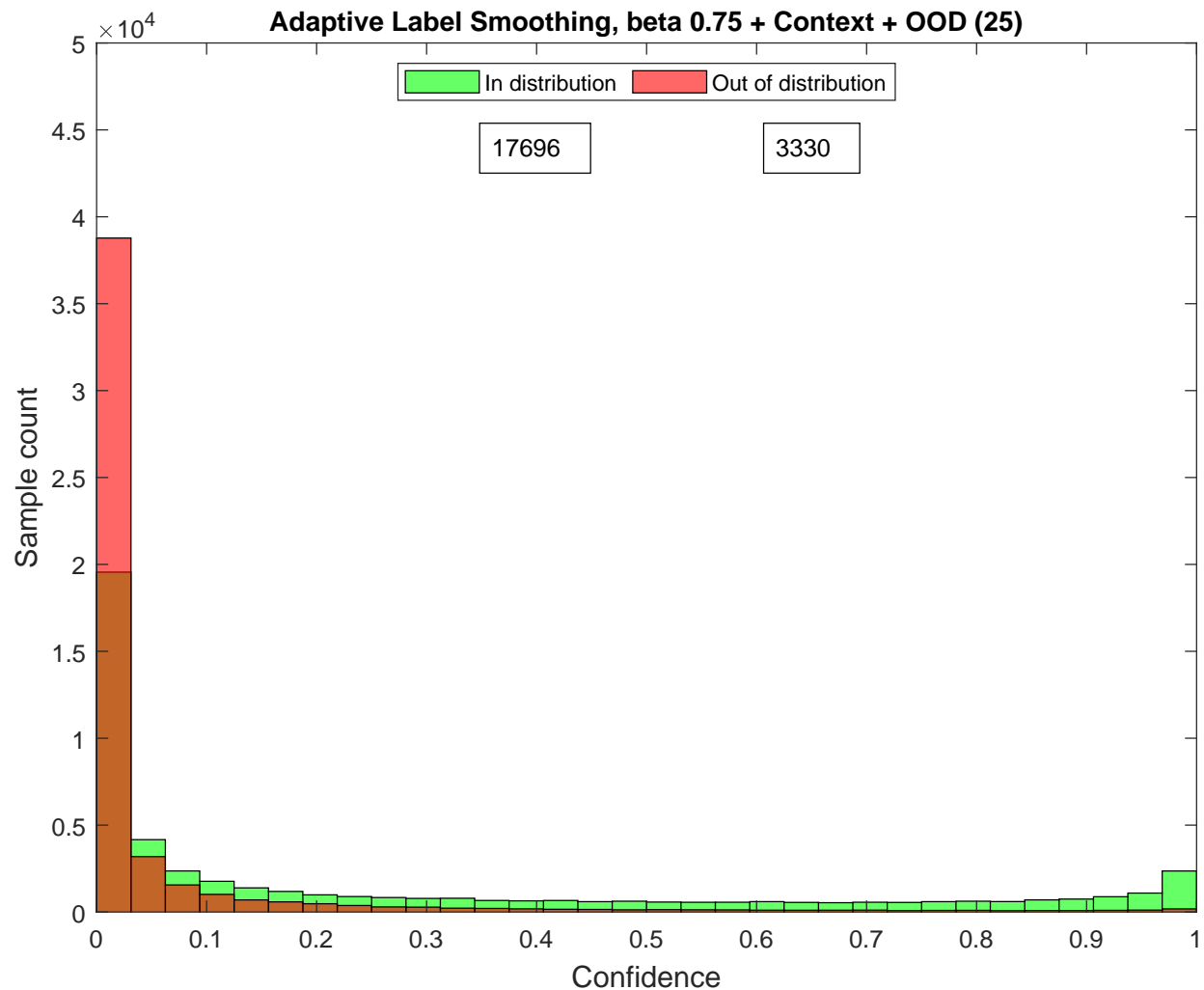


Figure 5.19: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 25 percent of samples are from 1000 classes of OImageNet training set.

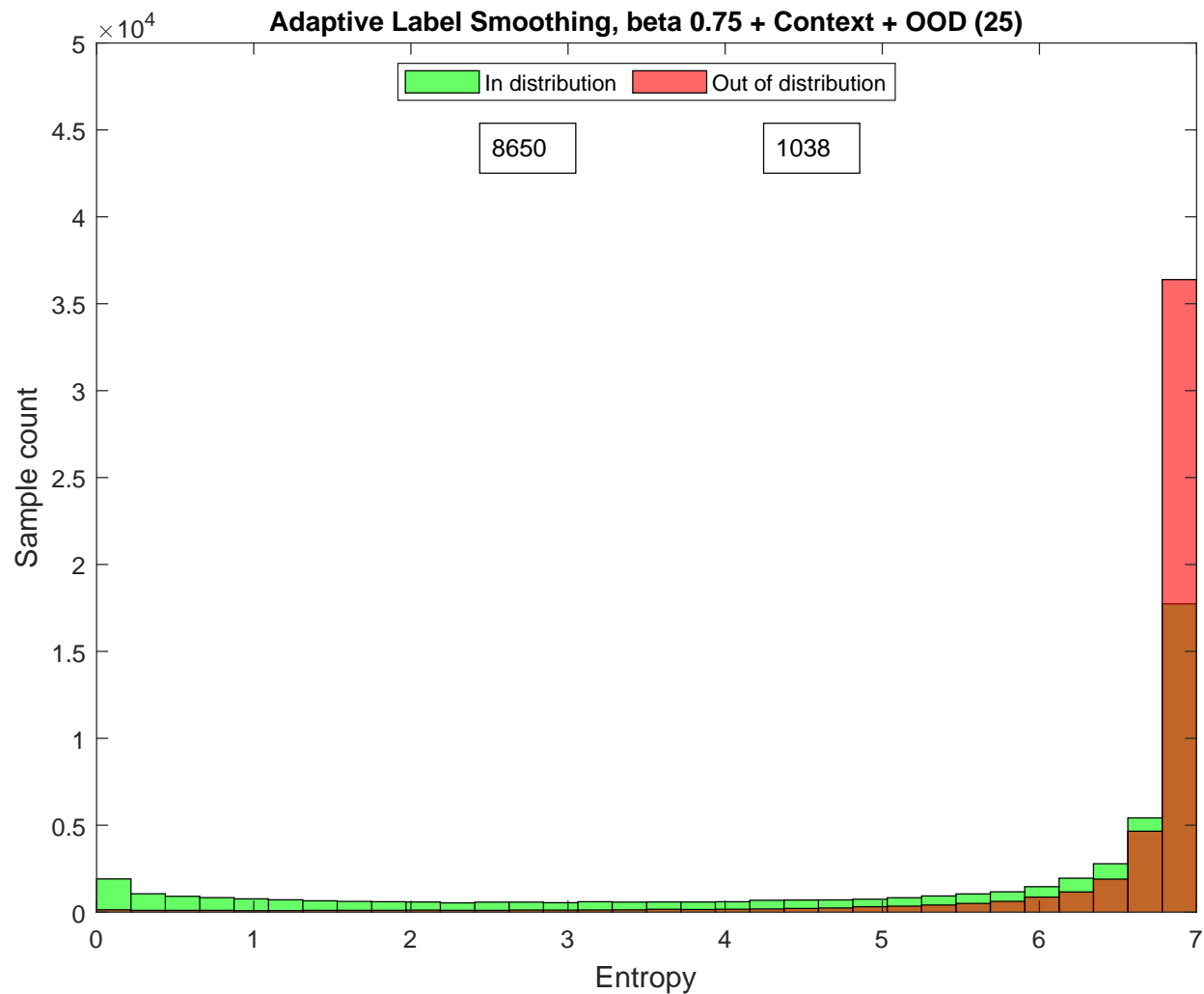


Figure 5.20: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 25 percent of samples are from 1000 classes of OImageNet training set.

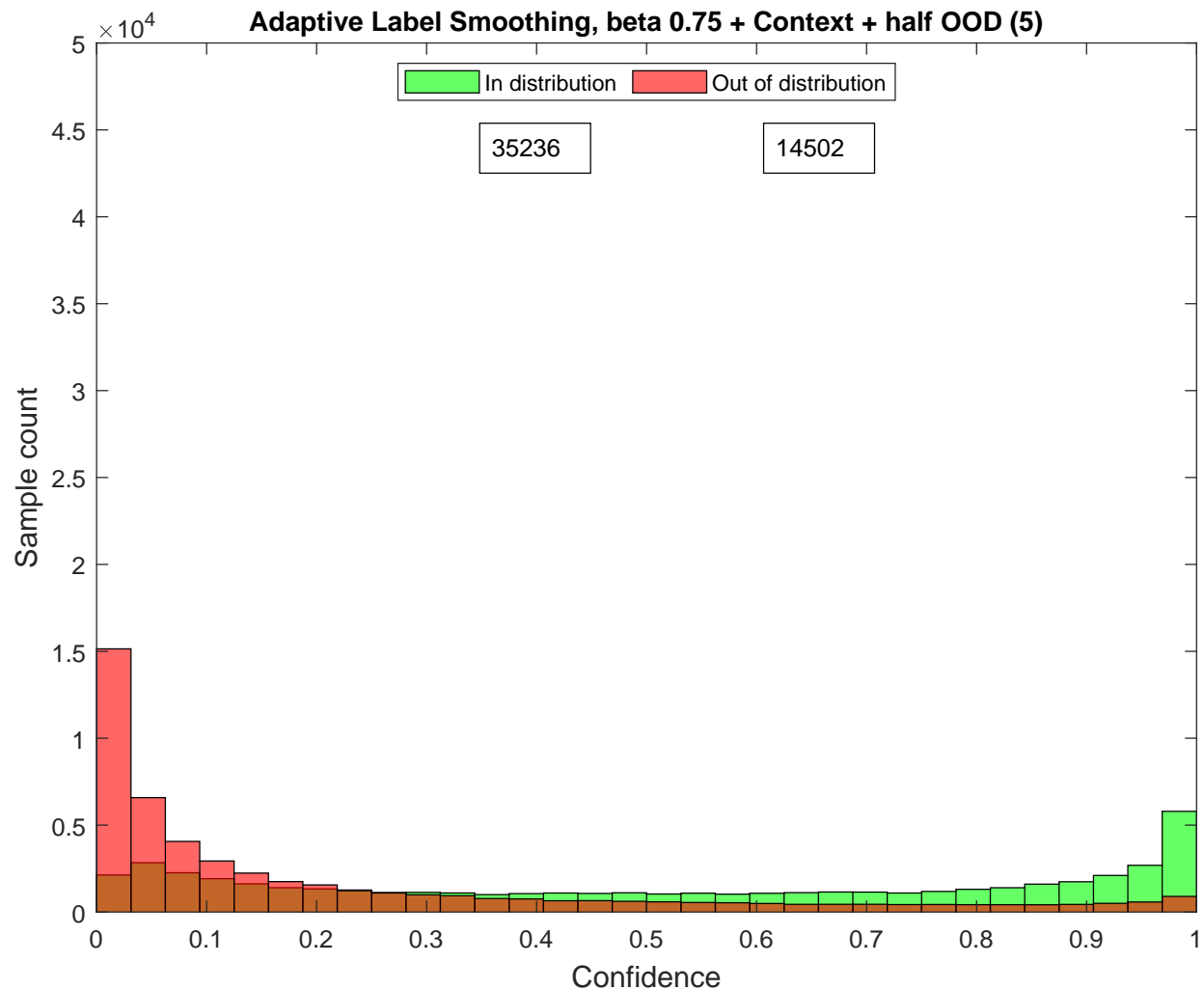


Figure 5.21: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 5 percent of samples are from 500 classes of OImageNet training set.

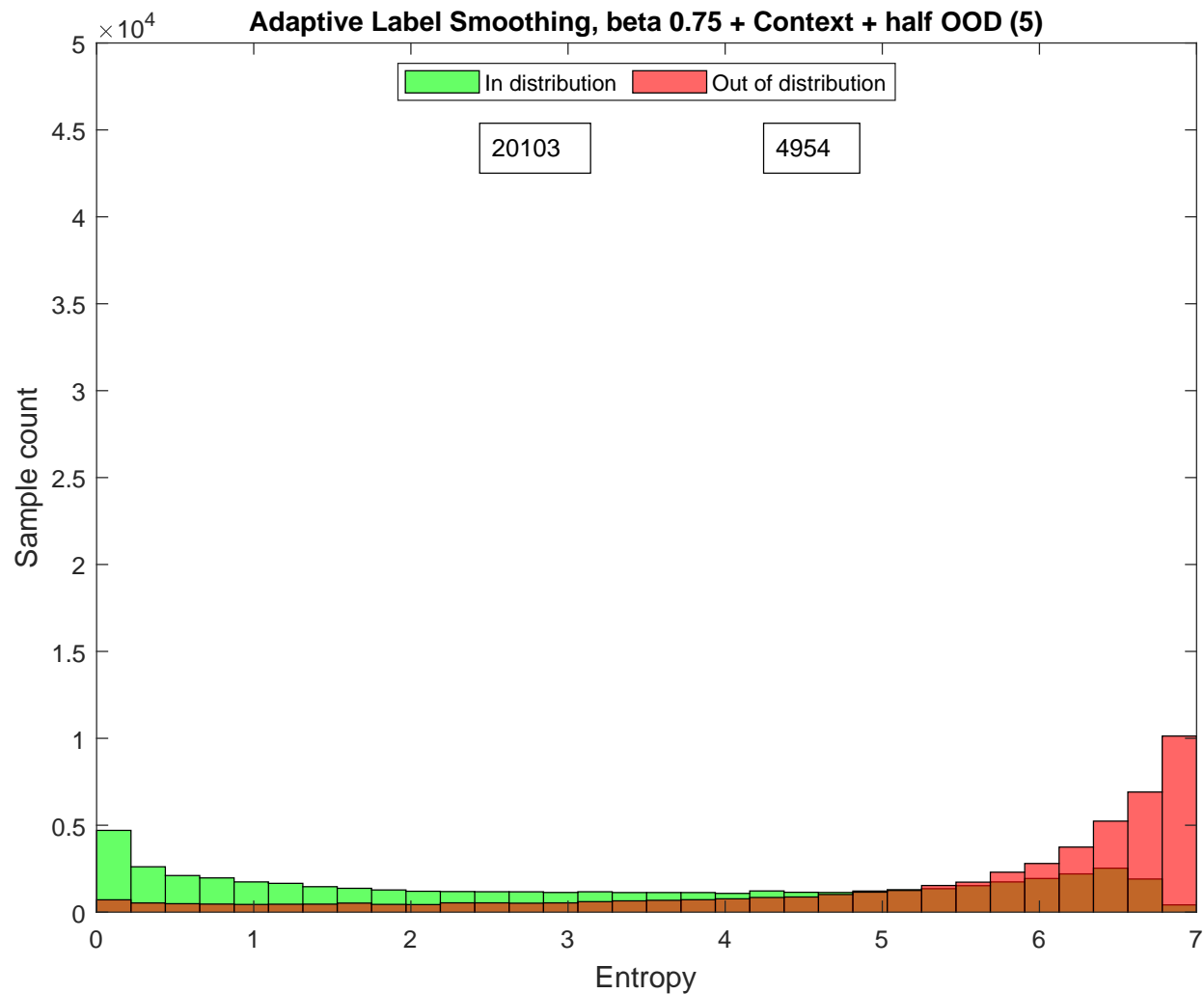


Figure 5.22: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 5 percent of samples are from 500 classes of OImageNet training set.

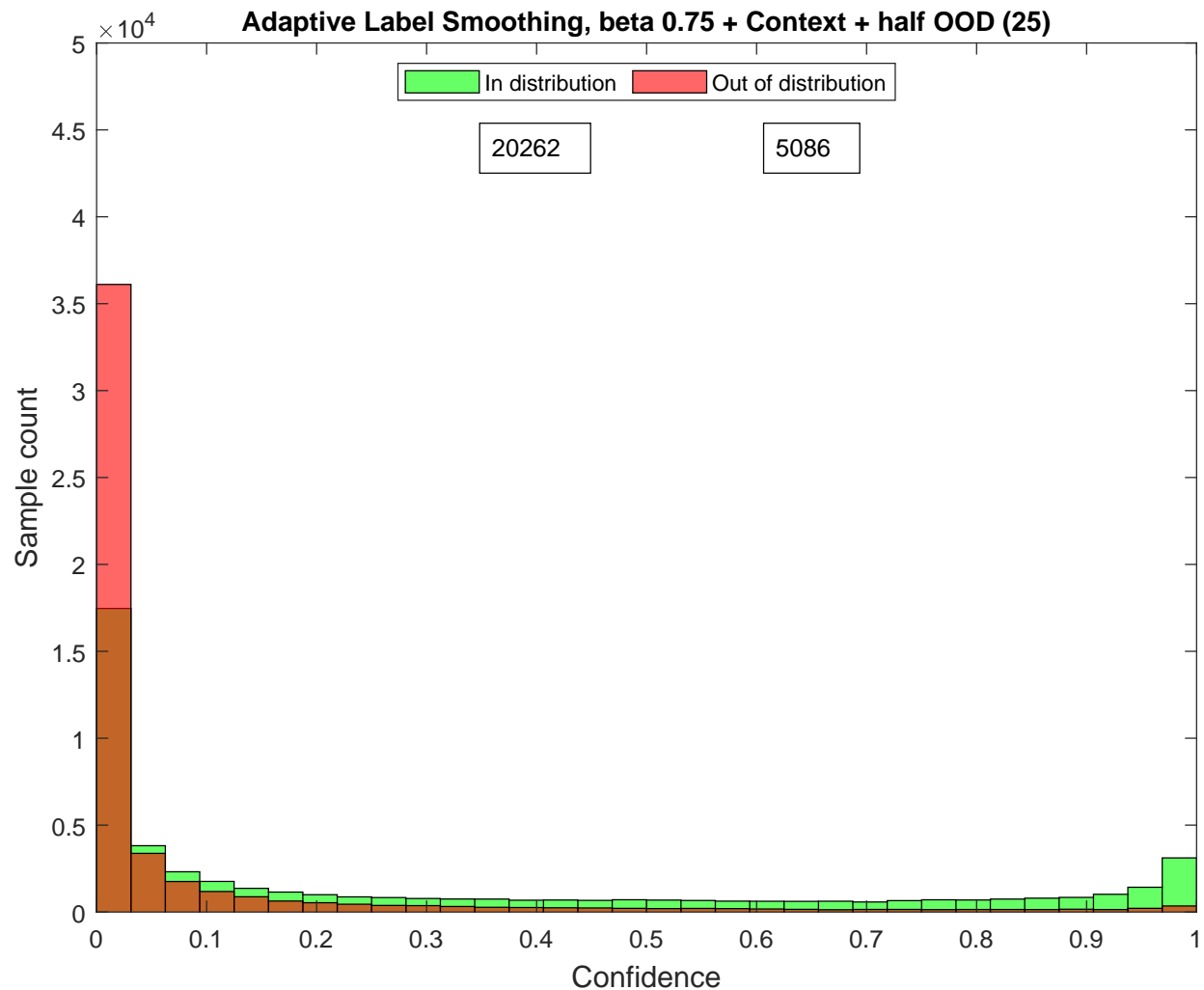


Figure 5.23: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 25 percent of samples are from 500 classes of OImageNet training set.

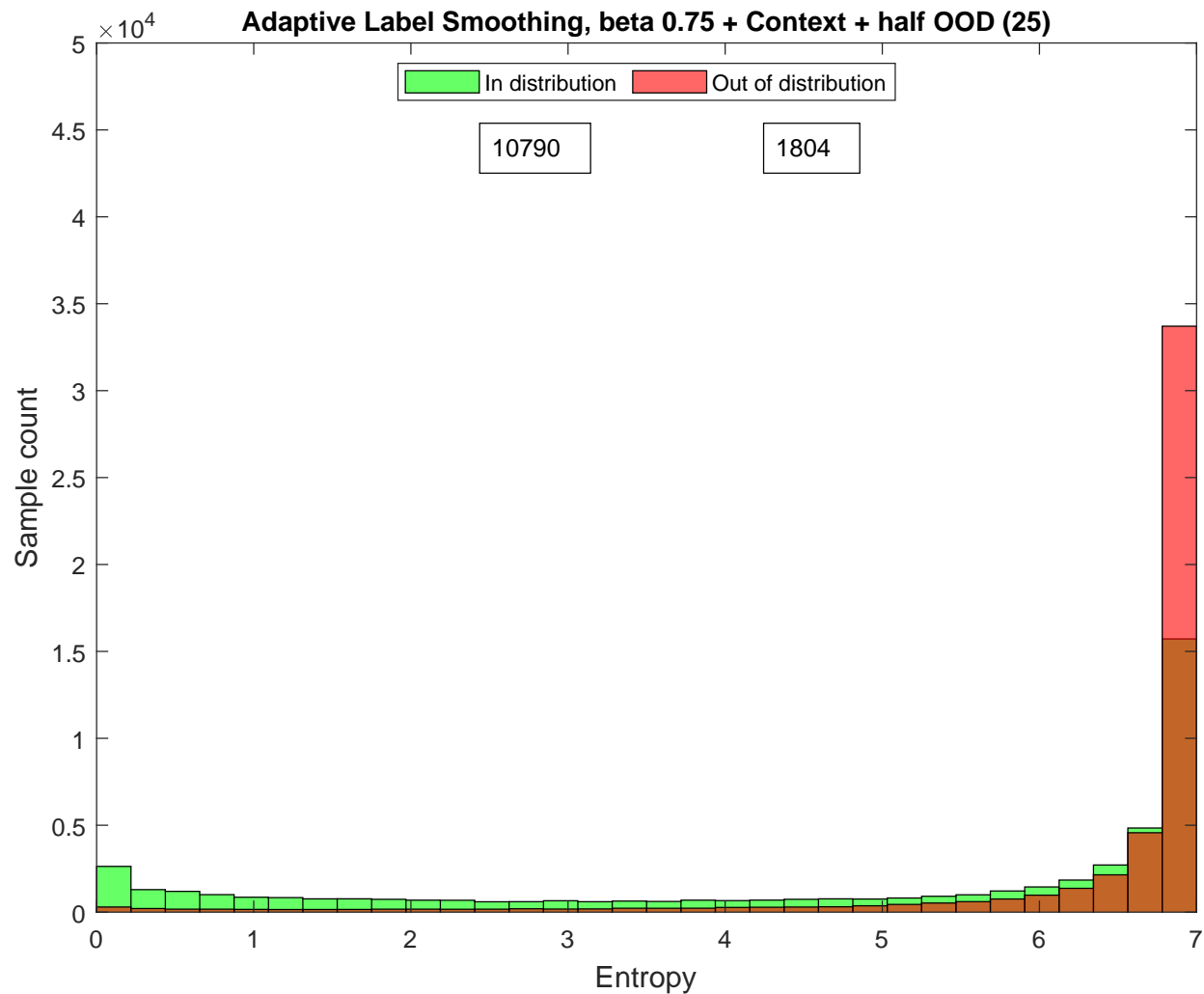


Figure 5.24: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 25 percent of samples are from 500 classes of OImageNet training set.

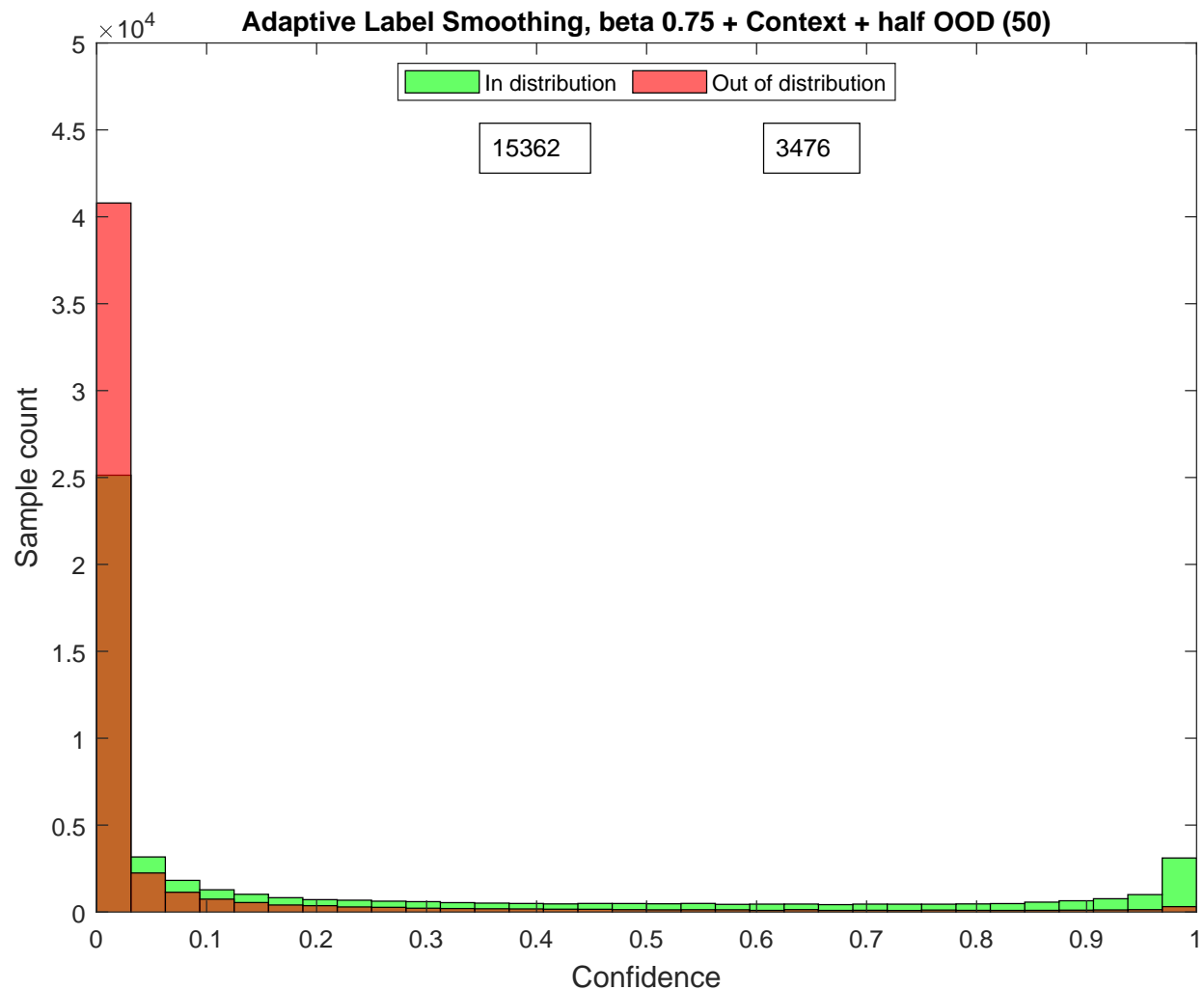


Figure 5.25: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 50 percent of samples are from 500 classes of OImageNet training set.

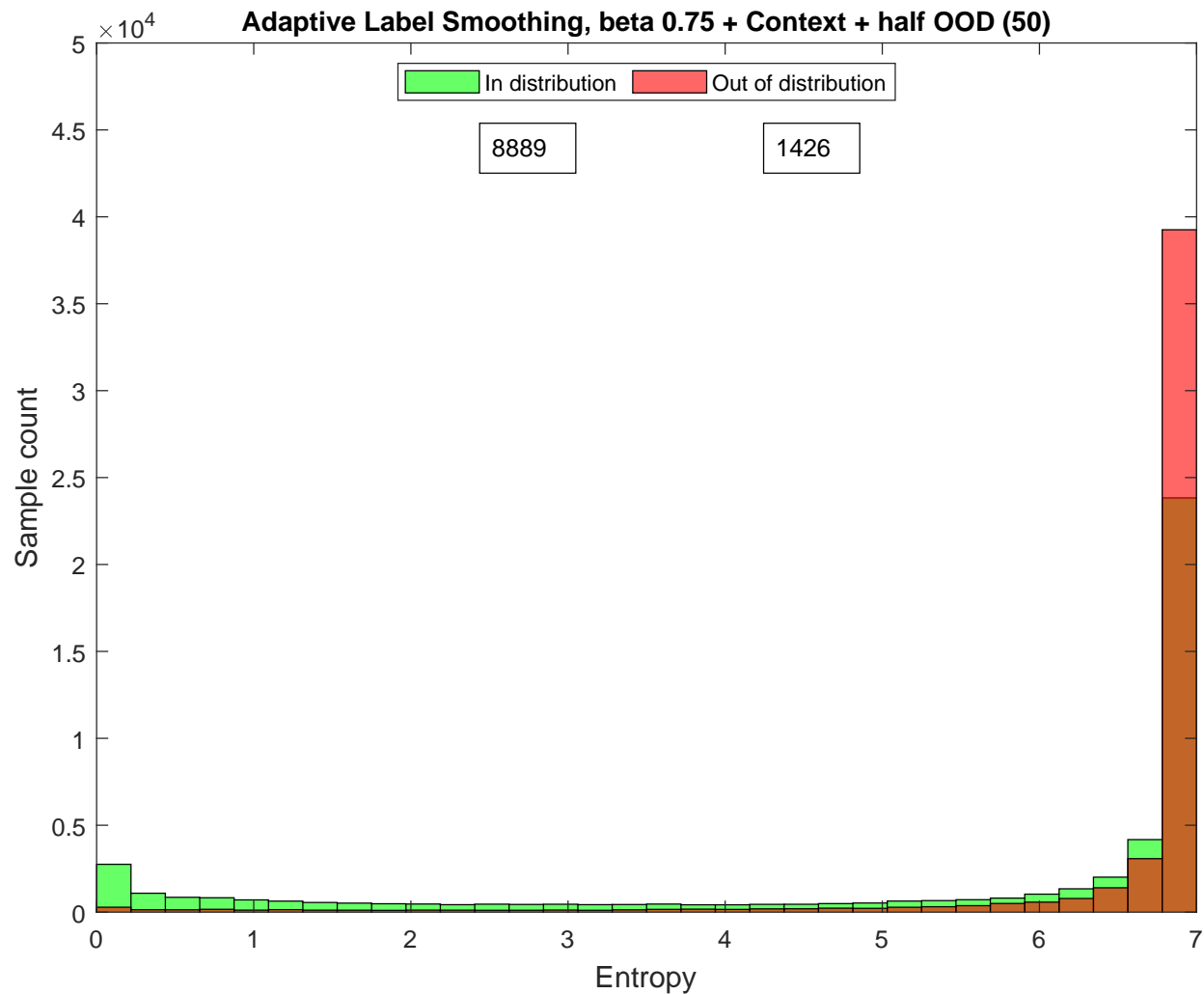


Figure 5.26: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 50 percent of samples are from 500 classes of OImageNet training set.

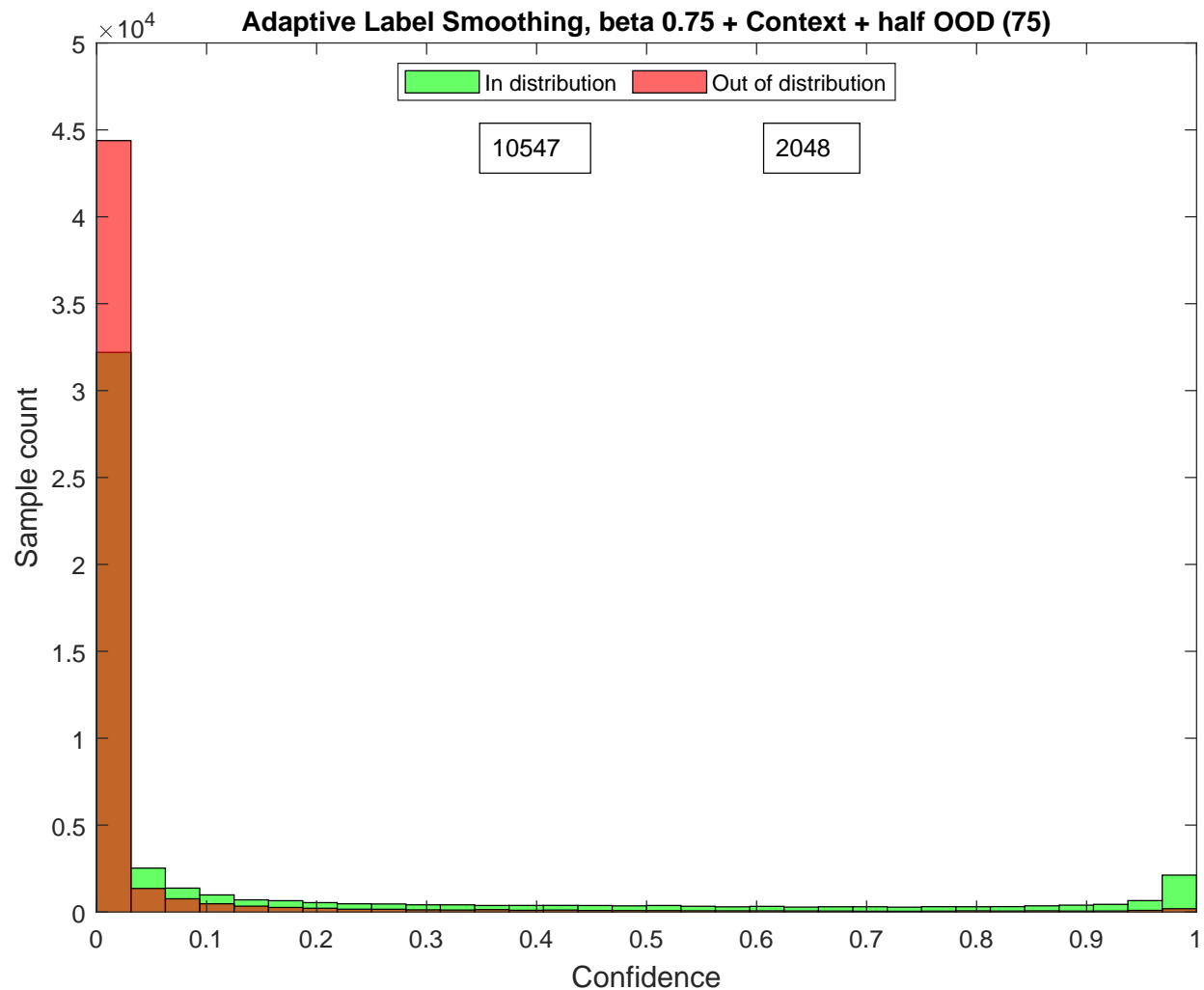


Figure 5.27: Confidence values (maximum value of the predicted output) of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 75 percent of samples are from 500 classes of OImageNet training set.

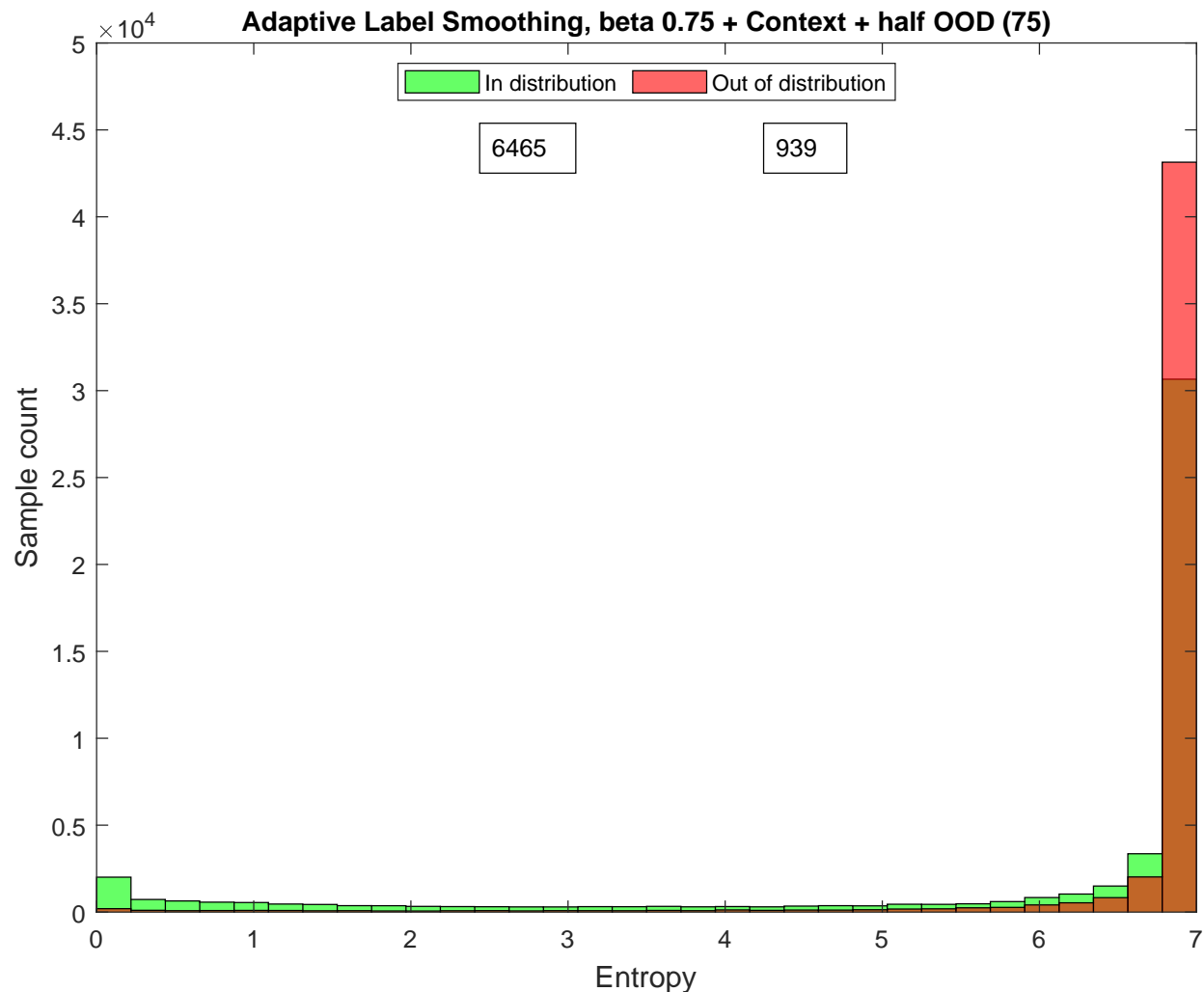


Figure 5.28: Entropy values of 50K ImageNet-1K and 50K OImageNet-1K validation images plotted as a histogram with 32 equispaced bins for our *adaptive* label smoothing method with  $\beta = 0.75$ , 5 percent of training samples are context only and 75 percent of samples are from 500 classes of OImageNet training set.

As we show in figures 5.15, 5.16, 5.17, 5.18, 5.19, 5.20, 5.21, 5.22, 5.23, 5.24, 5.25, 5.26, 5.27 and 5.28 the increase in the fraction of out of distribution samples supplied during training increases the proportion of in distribution samples after thresholding. However, the overall

number of samples drops drastically, this also results in an accuracy drop on the target dataset (in distribution). In the case of figure 5.15 the mean accuracy on the in distribution dataset is 68.1 percent, for figure 5.17 the accuracy is 67.7 percent and for figure 5.19 the accuracy is 62 percent. In the case of figure 5.27, the mean accuracy dropped to 47 percent.

## 5.5 Conclusion

This work has addressed the problems of out of distribution detection using a novel approach called *adaptive* label smoothing. Our approach helps train classifiers that pay attention to the more discriminative regions and learn to recognize out of distribution images better than standard hard label or label smoothing approaches. Our approach can be used to produce high entropy predictions when context-only or out of distribution images are provided as input to the classifier.

# Chapter 6

## Conclusions

The tremendous growth in the field of machine learning in the past two decades can be attributed to high accuracies and widespread real-world deployment of the models trained using large public datasets like ImageNet. Even though the modern day machine learning models have shown great improvement in performance, they suffer from a myriad of problems. Some of the problems commonly seen are poor generalization, overconfidence and poor out of distribution performance. These problems can be attributed to the datasets being used, algorithms being used to train and the model architectures as well. In this dissertation we developed approaches to mitigate some of the problems pertaining to bias, overconfidence and out of distribution detection.

Specifically, in Chapter 3 we developed an augmentation approach that is able to use bounding box information and augment data on the fly. This approach provides control over scale, size and position over multiple objects and maintains the class context at the same time. This is the first approach that ever utilized bounding box information to augment classifiers trained on ImageNet 1-K. In tests with ImageNet 1-K dataset, our approach when combined with CutMix using the ResNet-50 architecture outperforms the standard baseline by 1.8 percent. After fine tuning on the challenging MS-COCO, using Faster R-CNN with a ResNet-50 backbone, our approach when combined with CutMix yielded a performance of 32.3 mAP, which is an improvement of 0.5 mAP when just CutMix was used for the classifier regularization.

Chapter 4 addresses the algorithmic deficiencies in modern day image classifiers using a novel approach called adaptive label smoothing. We produce class probabilities that are grounded in the spatial footprint of the object being classified. This approach is novel and has a positive impact in improving the localization ability of the classifiers. Our approach also results in better interpretability of the outputs produced by the CNNs and reduces the overconfidence score. That is, we encourage our model to produce high entropy predictions on images that barely contain any relevant objects. When compared to other baseline approaches on ImageNet 1-K, our approaches produce the lowest overconfidence scores. After fine tuning on the challenging MS-COCO, using Faster R-CNN with a ResNet-50 backbone, our approach matches the performance of CutMix in terms of mAP. Our approach provides lower overconfidence scores and improved localization at the same time. Our confidences/predictions are based on the size of an object and we are over the diagonal because of this (under the diagonal is undesirable). *We are more accurate than we are confident compared to all baselines as we show using reliability charts in figure 4.7.* The average objectness of images in the validation set of ImageNet is 0.49 and the mean confidence of our approach is 0.60 compared to the mean confidence of 0.86 using the hard label approach. The mean deviation computed as the mean of the absolute difference between confidence and objectness for the hard label case is 0.42 and for our approach is 0.24. Using these metrics we show that our confidences are explainable as they closely match the objectness statistics much better, our predictions are explainable and we ground our labels/confidences in the object size as opposed to making correct predictions using contextual information only as shown in table 4.1.

Lastly, Chapter 5 extends the techniques developed in Chapter 4 to improve the out of distribution of CNNs. We supply images belonging to novel unseen classes during training, but provide a vector with a uniform probability distribution over all classes as the label for all of them. That is, we encourage our model to produce high entropy predictions on

unseen categories. Using bin counts we show that our approach is better and rejecting out of distribution samples when the confidence is thresholded at values over 0.25 and the entropy is thresholded under 2.

# Bibliography

- [1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11):2189–2202, 2012.
- [2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.
- [3] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 771–787, 2018.
- [4] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, pages 2654–2662, 2014.
- [5] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.
- [6] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.
- [7] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised

- object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2219–2228, 2019.
- [8] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3133–3142, 2020.
- [9] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In *Advances in Neural Information Processing Systems*, pages 851–863, 2019.
- [10] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.
- [11] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.
- [12] Morris H. DeGroot and Stephen E. Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 32:12–22, 1983.
- [13] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
- [14] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.
- [15] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Shu-Tao Xia. Adaptive regularization of labels. *arXiv preprint arXiv:1908.05474*, 2019.

- [16] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018.
- [17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338, June 2010.
- [18] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4): 193–202, 1980.
- [19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.
- [20] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pages 10750–10760, 2018.
- [21] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. <https://github.com/facebookresearch/detectron>, 2018.
- [22] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [23] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017.

- [24] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.
- [25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [26] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.
- [27] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–50, 2019.
- [28] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646–661. Springer, 2016.
- [29] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.
- [30] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.
- [31] Haris Iqbal. Plot neural net. <https://github.com/HarisIqbal88/PlotNeuralNet>, 2020.

- [32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- [33] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 623–631, 2017.
- [34] Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems*, pages 3792–3803, 2019.
- [35] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 2810–2819, 2018.
- [36] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Mallocci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pages 1–26, 2020.
- [37] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6402–6413, 2017.
- [38] Y. LeCun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W.E. Hubbard, and L.D. Jackel. Handwritten digit recognition with a back-propagation network. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan-Kaufmann, 1990.
- [39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to

- document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.
- [40] Yann LeCun. The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- [41] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pages 396–404, 1990.
- [42] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.
- [43] Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C Weng. A note on Platt’s probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276, 2007.
- [44] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.
- [45] Yu Liu, Guanglu Song, Yuhang Zang, Yan Gao, Enze Xie, Junjie Yan, Chen Change Loy, and Xiaogang Wang. 1st place solutions for openimage2019–object detection and instance segmentation. *arXiv preprint arXiv:2003.07557*, 2020.
- [46] Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *arXiv preprint arXiv:1902.02476*, 2019.

- [47] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.
- [48] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.
- [49] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip HS Torr, and Puneet K Dokania. Calibrating deep neural networks using focal loss. *arXiv preprint arXiv:2002.09437*, 2020.
- [50] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4696–4705, 2019.
- [51] D. Mund, R. Triebel, and D. Cremers. Active online confidence boosting for efficient object classification. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1367–1373, 2015.
- [52] Allan H. Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology*, 12(4):595–600, 1973.
- [53] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 2901–2907, 2015.
- [54] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 625–632, 2005.

- [55] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.
- [56] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large-Margin Classifiers*, pages 61–74. MIT Press, 1999.
- [57] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400, 2019.
- [58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.
- [59] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.
- [60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [61] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [62] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3544–3553. IEEE, 2017.

- [63] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.
- [64] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in Neural Information Processing Systems*, pages 2377–2385, 2015.
- [65] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [66] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.
- [67] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [68] Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. In *Advances in Neural Information Processing Systems*, pages 6414–6425, 2019.
- [69] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Ricap: Random image cropping and patching data augmentation for deep cnns. In *Asian Conference on Machine Learning*, pages 786–798, 2018.
- [70] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5486–5494, 2018.

- [71] Jonathan Wenger, Hedvig Kjellström, and Rudolph Triebel. Non-parametric calibration for classification. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, Proceedings of Machine Learning Research, 2020. URL <https://github.com/JonathanWenger/pycalib>.
- [72] Lingxi Xie, Jingdong Wang, Zhen Wei, Meng Wang, and Qi Tian. Disturblabel: Regularizing cnn on the loss layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4753–4762, 2016.
- [73] Jianwei Yang, Jiasen Lu, Dhruv Batra, and Devi Parikh. A faster pytorch implementation of faster r-cnn. <https://github.com/jwyang/faster-rcnn.pytorch>, 2017.
- [74] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.
- [75] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 694–699, 2002.
- [76] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87. URL <https://dx.doi.org/10.5244/C.30.87>.
- [77] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

- [78] Quanshi Zhang, Wenguan Wang, and Song-Chun Zhu. Examining cnn representations with respect to dataset bias. *arXiv preprint arXiv:1710.10577*, 2017.
- [79] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.
- [80] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2921–2929. IEEE, 2016.