Proxy Attention : Comparing and Combining Augmentation with Attention

Graduation Project
(Computational Intelligence and Robotics)

Subhaditya Mukherjee (s4747925)
April 4, 2023

Internal Supervisor: S.H. Mohades Kasaei, PhD
Second Internal Supervisor: Matias Valdenegro, PhD
(Artificial Intelligence, University of Groningen)
Artificial Intelligence
University of Groningen, The Netherlands

CONTENTS

LIST OF FIGURES

LIST OF TABLES

KEY

1. $\odot$ denotes element-wise multiplication

CHAPTER 1

INTRODUCTION

1.1 Context and Novelty

1.2 Motivation

1.3 Challenges

1.4 Problem Statement

1.5 Research Questions

1.6 Thesis Outline

CHAPTER 2

BACKGROUND

2.1 Interpretability

- Need for Interpretability

2.2 Gradient Based Explanations

- Taxonomy

2.3 Augmentation

- Taxonomy

2.4 Datasets

To test Proxy Attention, the following datasets were used. Note: Images are resized to 224x224 pixels for consistency. These batch visualizations are generated by the author using the torchvision and matplotlib libraries.

2.4.1 CIFAR 100

The CIFAR 100 dataset, introduced by [1] is an image dataset with 60000 color images with dimensions 32x32 pixels. As the name suggests, the dataset has 100 unique classes. Each of these classes have 500 training images. Some of the classes are - airplane, bird, truck, ship, deer and dog. This dataset is used as a coarse grained classification dataset in this project.

2.4.2 Stanford dogs

2.4.3 Imagenette

2.4.4 ASL

2.4.5 Food-101

Plant Village

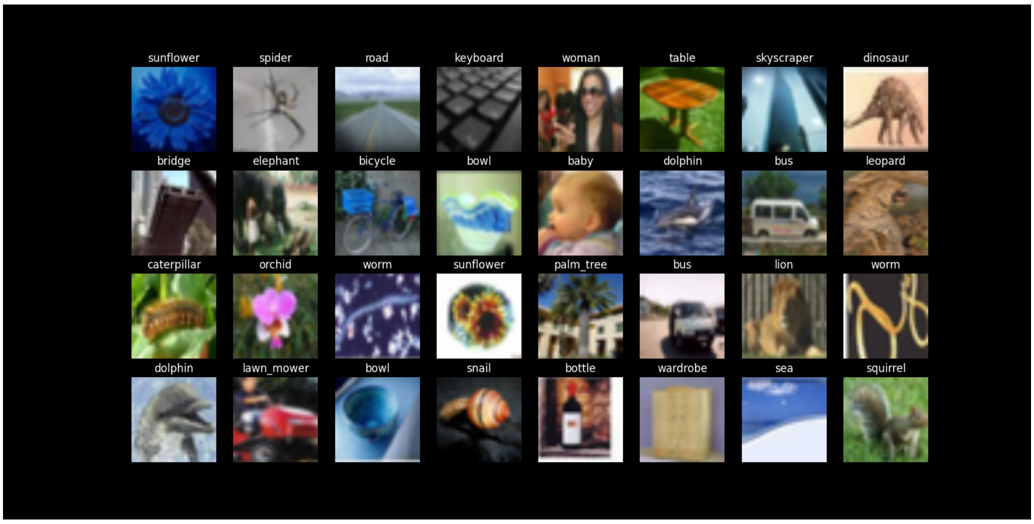

Figure 2.1: CIFAR100 Sample

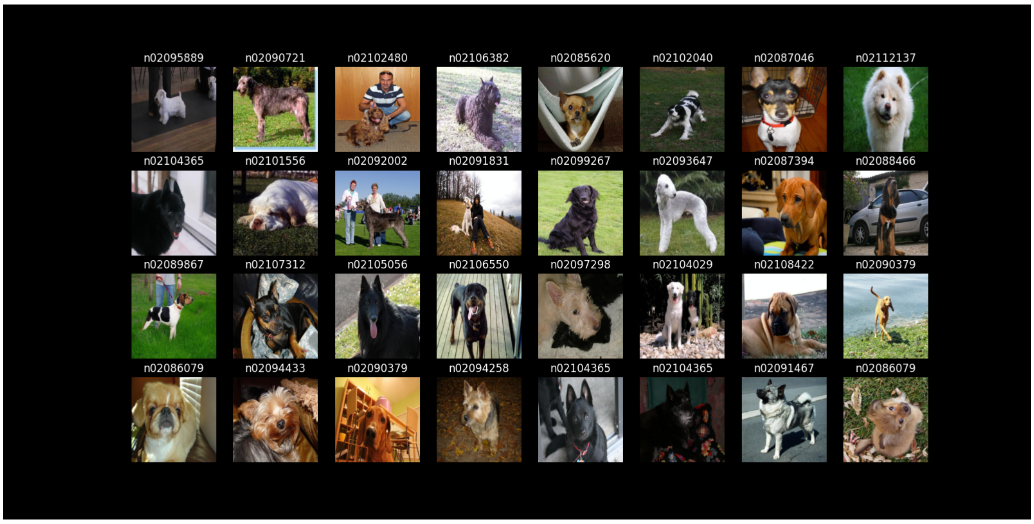

Figure 2.2: Stanford Dogs Sample

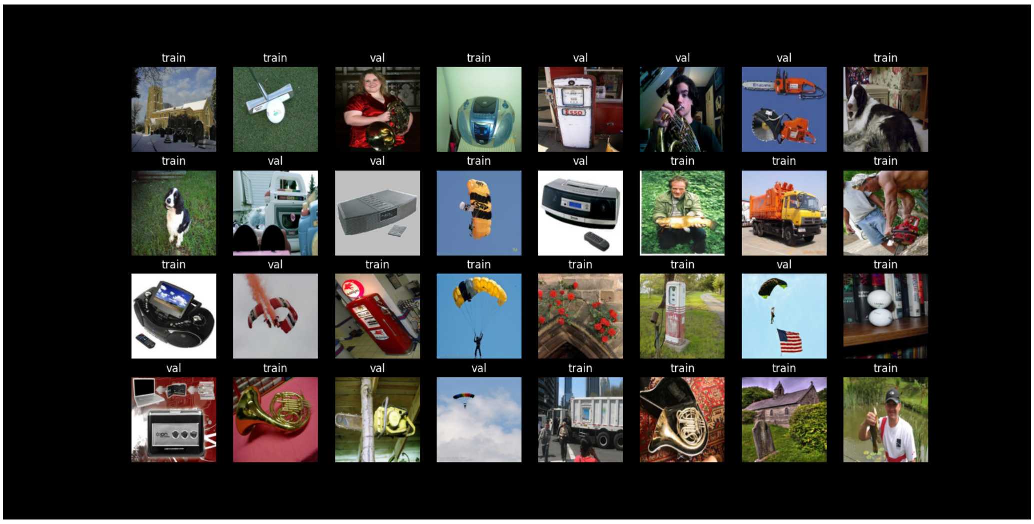

Figure 2.3: Imagenette Sample

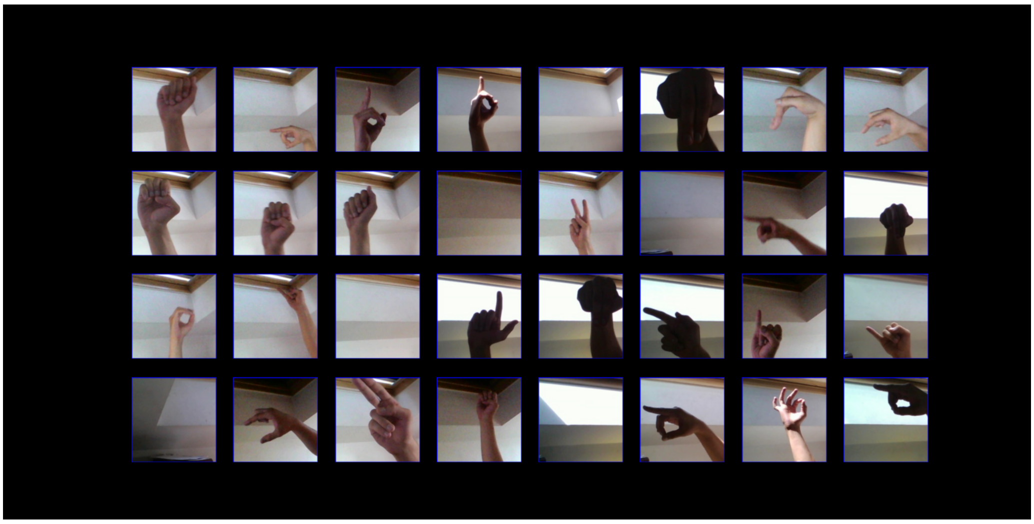

Figure 2.4: ASL Sample

CHAPTER 3

STATE OF THE ART

3.1 Gradient Based Explanations

Beware Of Inmates
Interpretation Is Fragile
Sanity Checks
The Unreliability Of Saliency Methods
There And Back Again
Influence Of Image Class Acc On Saliency Map Estimation
Deconvnet
Deep Inside Conv Nets
Cam
Gradcam++
Guided Backprop
In another paper, the authors propose a score weighted approach (ScoreCAM) to create saliency maps [2]. Like many other methods, the images are first passed through the network and the corresponding activations are obtained from the final convolutional layer. These activation maps are then upsampled and normalized to the range of $[0,1]$. The portions of the activation maps that were highlighted are then passed through a CNN with a SoftMax layer to obtain the score for each of the current classes. These scores are used to find the relative importance of all the activation maps. Finally the sum of all these maps is computed using a linear combination with the corresponding target score and then passed through a ReLU operation. These operations can be mathmaticaly represented as $L^c_{ScoreCAM} = ReLU(\sum_k w^c_k A^k)$, where k represents the index considered, c represents the current class and S_k represents the outputs of the aforementioned SoftMax layer. The authors find that the maps obtained using ScoreCAM are less noisy and using this method removes dependancy on unstable gradients as compared to other methods.
Guided Gradcam A variant of GradCAM [3] was proposed by Selvaraju et al. [4] where unlike GradCAM that finds the parts of the image that influence the model's decision, Guided GradCAM takes the positive gradients into account. These gradients are used to obtain an even more fine-grained representation of the outputs of the saliency map. While GradCAM backpropagates both positive and negative gradients, Guided Backprop only propagates the positive gradients and is defined as a pointwise multiplication of the results of GradCAM and Guided Backpropagation [5].
Salience Map
In combination with attribution methods, Noise Tunnel [6] is an algorithm that improves the accuracy of the masks obtained by these methods. Noise Tunnel was proposed to counter noisy and irrelevant attributions obtained by some of the gradient based methods by adding a Gaussian Noise and then averaging the predictions over sampled attributions. Since all the samples are considered, this method has a significant computational overhead. For Smooth Grad [7], the new attribution is defined as $\hat{M}_c(x) = \frac{1}{n}\Sigma^n_1 M_c(x+\mathcal{N}(0,\sigma^2))$. Where M_c is the attribution calculated by SmoothGrad, $\mathcal{N}(0,0.01^2)$ is the Gaussian Noise with $\sigma = 0.01$ and n is the number

of samples. Similarly for Smooth Grad Square, $\hat{M}_c(x) = \frac{1}{n}\Sigma_1^n \sqrt{M_c(x + \mathcal{N}(0,\sigma^2))}$. Noise Tunnel can also be used on Var Grad [8] with the equation $\hat{M}_c(x) = \frac{1}{n}\Sigma_{k=1}^n \{M_c(x + \mathcal{N}(0,\sigma^2))\}^2 - \{\hat{M}_c(x)\}^2$

For a model F, the attribution method Integrated Gradients [9] computes the contribution of each pixel in the image towards the final prediction. The output of the model is used to calculate a pixel wise partial derivative that is then integrated along a path starting from the baseline and ending at the input. Each of the steps are scaled according to the partial derivative obtained in the previous step. For every step k with m total steps over the path, the IG equation is defined as $IntegratedGrads_i^{approx}(x) ::= (x_i - x_i') \times \Sigma_{k=1}^m \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m}$. Where $(x_i - x_i')$ is the pixelwise difference between the two images, $\frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i}$ is the partial derivative of the model output F with respect to pixel i at the k-th step of the path and $\frac{1}{m}$ is the scaling factor that ensures that each of the steps taken contribute equally to the final result.

Sam Resnet
Conductance
Deep Fool
Deep Lift
Generalizing Adversarial Exp With Gradcam
Shap
Smooth Grad
Smooth Grad Square
Lime
Sp Lime
Summit

Petsiuk et al. propose RISE [10], a saliency method that randomly alters the input images by applying random noise to each of them. After model predictions are obtained, the saliency map is generated by a combination of the partial maps over each of the modified images. RISE improves accuracy but needs a lot of computation time considering that multiple models need to be trained for each of the random noise samples.

Lrp
Var Grad
Visualizing Impact Of Feature Attribution Baselines
Adaptive Whitening Saliency
Bayesian Rule List
Deep Visual Explanations
Dynamic Visual Attention
Embedding Knowledge Into Deep Attention Map
Graph Based Visual Saliency

3.2 Augmentation

Another augmentation strategy proposed by [11] first applies multiple transformations randomly and in parallel chains to each image. These transformations can include combinations of Translation, Rotation, Shearing etc. The outputs of these combinations are then mixed to form a new image, which is then further mixed with the original image to form the new image. This combination is done to improve performance in cases where data shifts are encountered in production. Once the images are mixed, a skip-connection is used to combine the results of the chains. AugMix also uses the Jensen-Shannon Divergence consistency loss [12] to ensure that the images are stable across a range of inputs. Considering KL to be Kullback-Leibler Divergence, the Jensen-Shannon Divergence can be defined as $JS(p_{orig}; p_{augmix1}; p_{augmix2}) = \frac{1}{3}(KL[p_{orig}||M||] + KL[p_{augmix1}||M||] + KL[p_{augmix2}||M||])$, where M is the mean of the three distributions $p_{orig}, p_{augmix1}, p_{augmix2}$.

Devries et al. in their paper [13] propose an augmentation method they call Cutout. In this method, random sized square patches are removed from the images by replacing the corresponding pixels with a constant value (usually 0). Selecting the region involves picking a random pixel value and then creating a uniform sized square around the chosen pixel. The authors also find that Cutout performs better in combination with other methods rather than just being used by itself. Cutout can be expressed as an element-wise multiplication operation $x_{cutout} = x \odot M$, where x is the original image, M is a binary mask of the same size as x with randomly chosen coordinates of a square patch of pixels to be cut out, and $\odot$ denotes element-wise multiplication.

Unlike Cutout [13], where the chosen patch is replaced with zero pixels, in CutMix [14] the chosen patch is replaced with a randomly chosen patch from a different region of the same image. Yun et al. propose this approach as multiple class labels can be learned with a single image. CutMix can be defined by the following operations $\widetilde{x} = M \odot x_A + (1-M) \odot x_B$; $\widetilde{y} = \lambda y_A + (1-\lambda) y_B$. where x is an RGB image, y is the respective label, M is a binary mask of the patch of the image that will be dropped and $\odot$ represents element wise multiplication. The new training sample $\widetilde{x}, \widetilde{y}$ is created by combining two other training samples x_A, y_A and x_B, y_B. To control the combination ratio λ, a sample from the $\beta(1,1)$ distribution is chosen. This combination is quite similar to [15] but differs in the sense that CutMix focuses on generating locally natural images. Building up on [14], Walawalkar et al. propose an alternative method of replacing patches in an image they call Attentive CutMix [16]. In this method, instead of randomly pasting patches in the image, a pre-trained network is used to identify attentive regions from the image. Similar to the earlier approach, these patches are then mapped back to the original image. Doing so allows the network to select backround regions that are important for the task while also updating the label information.

Many of the algorithms use rectangular or square shaped masks. While they are effective, French et al. propose Cow Mask [17], a new method of masking that uses irregularly shaped masks with a Gaussian filter to reduce noise. The authors also propose two methods of mixing, one that builds up on Random Erasing [18], and another that uses Cut Mix [14]. A pixel wise mixing threshold is also chosen, and either mixing or erasing is applied to the image based on this threshold. This augmentation technique is shown to be effective in semi-supervised learning.

Another approach involving a cut-paste methodology was proposed by [19]. In their paper, the authors propose a new method of augmentation that extracts instances of objects from the images and instead of pasting them on other images, they are pasted on randomly chosen backgrounds. This method leads to pixel artifacts in the images as selecting the objects is a noisy process. To overcome the drop in performance as a result of this, the authors apply a Gaussian blur and poisson blending to the boundaries of the pasted objects. Further augmentaion is applied before pasting the objects by rotation, occlusion and truncation. The authors also find that this approach makes the network more robust to artifacts in the images.

In their paper Singh et al. [20] propose a data augmentation method that takes an image as an input, and divides it into a grid. Each of the sub-grids are then turned off with a given probability. These sub-grids can be connected or independant of each other and the turned off grids are replaced by the average pixel value of all the images in the dataset.

One of the major drawbacks of algorithms that rely on modifying image patches (such as [20, 13, 18]) is that they sometimes delete parts of the image that might be useful to the network. To overcome this problem Chen et al. propose a new method Grid Mask [21] that uses evenly spaced grids to find a balance between the amount of information that is deleted and stored. Using the number of grids and their respective sizes as a hyperparameter, the authors find that Grid Mask is effective in preserving important parts of the image.

Zhang et al. propose another method of data augmentation that uses a CAM [22] to identify the most important regions of an image. These parts are then thresholded, scaled, translated and pasted onto the target image. A similar process is also applied to the target image and the attentive parts of the original image are used to replace the corresponding attentive parts of the target image. Similar to previous methods, the labels are also updated to reflect the changes in the image.

While Cutout augmentation [13] is applied to every image in the dataset, Zhong et al. propose a new method, Random Erasing, that takes a probability of being applied into account [18]. In Random Erasing, contiguous rectangular regions are selected and replaced at random with random upper and lower limits chosen for both region area and aspect ratio. For object detection tasks, a region aware detection algorithm is applied to make the network more robust to occlusion. Note that Cutout removes square patches, while Random Erasing either removes square or rectangular patches.

Many of the augmentation methods that rely on randomly choosing regions to cut and paste from sometimes fail to work well with regions that lack object information. ResizeMix [23] tackles this problem by replacing the patch with a proportional resized version of the selected image. This method is similar to CutMix [14] but differs in the sense that ResizeMix uses a resized version of the entire image instead of a randomly chosen patch.

Another augmentation technique that applies random cropping and pasting is RICAP [24]. In this method, four regions are cropped from different images and then pasted together to form a new image. The created image thus has multiple mixed labels. A uniform distribution is used to determine the area of each cropped region in the final image. The authors propose multple variants of RICAP that use different points of origin for cropping. They find that the method works best when the cropped regions use the corners as the origin as it allows the network to see more of the image.

While algorithms like Mixup [15] modify the labels of the image proportional to the amount of mixing between the original and the target images, Sample Pairing [25] maintains the same training labels. In their paper, Inoue et al. propose a method that merges images not by cut and paste but by averaging their pixel intensites. Sample Pairing follows an interval based augmentation policy, where the network is first trained for a 100 epochs normally before being introduced to the mixed images. This process is also repeated cyclicaly with eight epochs of training with mixed images followed by 2 epochs of training with normal images only.

With the success of mask based approaches for data augmentation, there have been many papers that attempt to fix the flaws of previous research. One such method is SmoothMix [26], which builds up on both CutMix [14] and Cutout [13] but modifies the mask to have softer edges. The intensity of the masked edges gradually decreases and depends on the strength of the mask. The updated pixel values are thus obtained by mixing the mask with the original image according to the formula $\lambda = \frac{\Sigma_{i=1}^{W}\Sigma_{j=1}^{H}G_{ij}}{WH}$. Where G_{ij} is the pixel value of mask G and H,W are the height and width of the image respectively. The new pixel values are then $(x_{new}, y_{new}) = (G.xa + (1-G).xb, \lambda.ya + (1-\lambda).yb)$

One of the older methods of data augmentation is SMOTE [27]. This algorithm is not domain specific but in the context of computer vision, it can be used to balance datasets that suffer from imbalanced labels. SMOTE generates new samples by combining the K-nearest neighbors of the minority class images to form new instances. Although many of the other methods discussed in this paper are more effective, SMOTE is still a useful tool to have.

Huang et al. propose SnapMix [28], where choosing the size of the patch to be cut is determined from the beta distributions of both the original and target images. The extracted patches are then merged with random image regions, each of which are of different sizes. Labels are also updated by taking the composition of the images into account. Cao et al. address the problem of class imbalance by performing data augmentation on images that are part of a minority class. From the labels of the images that were mixed, the final label is chosen as the label of the image with the least representation in the dataset. The authors call this method ReMix [29]. Dvornik et al. propose Visual Context Augmentation [30] that uses a NN to understand the context of objects in the image before pasting them in the target image. The authors generate training data by first generating pairs of context images with the objects masked out. These images are then fed into the NN to learn the difference between objects and bacgrounds given the masked pixels. Once the model has learnt this information, instances of the objects are placed into the masked regions of the target image.

While there are many techniques based on Mixup [15], they are mostly focused on generating new samples of images from the existing data. Doing so is useful, but sometimes leads to the generation of examples that confuse the network and are not representative of the actual data. To tackle this issue, Kim et al. [31] propose Pizzle Mix, an algorithm that learns to copy patches of images between each other while taking saliency into account. Puzzle Mix learns to minimize the equation $h(x_0,x_1) = (1-z)\odot\Pi_0^T x_0 + z\odot\Pi_1^T x_1$ where x_0, x_1 are the two images, z_i is a binary mask, $\lambda = \frac{1}{n}\Sigma_i z_i$ is the mixing ratio and Π_0, Π_1 represent $n \times n$ grids that denote the amount of mass that is transported during transport of the image patch to another location.

Attributemix Augmentaiton with curriculum leanring Co mixup Image Mixing and deletion

Keep augment Latent space interpo Randaugment Random distortion Saliencymix

Spec augment

3.2.1 Summary

3.2.2 Limitations

- Context
- Does not make use of what the network knows
- Does not help the network learn from its mistakes

3.3 Architectures

Resnet 18, 50
VGG
Vision Transformer

CHAPTER 4

PROPOSED APPROACH

4.1 Design Decisions

Efficient Computation Updating Dataloaders Batched Implementation Callbacks Training Resumption Logging

4.2 Hyper Parameters

4.2.1 Clear Every Step

4.2.2 Gradient Threshold Considered

4.2.3 Multiply Weight

4.2.4 Proxy Steps

4.2.5 Subset Of Wrongly Classified

4.2.6 Gradient Method

4.2.7 Architectures

CHAPTER 5

IMPLEMENTATION

5.1 Overview

5.2 Hyper parameters

5.2.1 Clear Every Step

5.2.2 Gradient Method

5.2.3 Gradient Threshold Considered

5.2.4 Multiply Weight

5.2.5 Proxy Steps

5.2.6 Subset Of Wrongly Classified

5.3 Data Loading and Pre Processing

5.3.1 Directory structure

5.3.2 Label function

5.3.3 Clearing proxy images

5.3.4 Encode, Stratify, Kfold

5.3.5 train and test, val separate

5.3.6 Augmentations

Imagenet Normalize Tensor Num workers

5.4 Training Details

5.5 Grid Search

5.6 Optimizations

5.6.1 Mixed Precision

5.6.2 Gradient Scaling

5.6.3 No grad

5.6.4 Batched Proxy step

5.6.5 Trial Resumption

5.6.6 Models

TIMM

5.7 Gradient Based Methods

5.8 Proxy Attention

5.8.1 Callback Mechanism

5.9 Tensorboard

5.10 Transfer learning

5.11 Optimizer

5.12 LR scheduler

5.13 Loss function

5.14 Batch sizer finder

To maximize training performance, a batch size finder is used to find the optimal batch size for each of the models.

5.15 Result Aggregation

5.16 Inference

Algorithm 1 Batch Size Finder Algorithm

```
Require: dataset_size, max_batch_size, failed
  batch_size = 2
  while TRUE do
    if max_batch_size is not None & batch_size ≥ max_batch_size then
      batch_size ← max_batch_size
    end if
    if batch_size ≥ dataset_size then
      batch_size ← batch_size//2
    end if
    if failed is False then
      loop
        inputs ← random((batch_size, input_shape))
        targets ← random((batch_size, output_shape))
        outputs ← model(inputs)
        loss ← MSE(outputs, targets)
        loss.backward()
        optimizer.step()
        optimizer.zero_grad()
        failed ← True
        batch_size ← batch_size * 2
      end loop
    else if failed is True then
      failed ← False
      batch_size ← batch_size//2
    end if
  end while
```

CHAPTER 6

EVALUATION

6.1 Metric Based Analysis

6.2 Visual Based Analysis

6.3 Summary

CHAPTER 7

CONCLUSION

7.1 Contributions

7.2 Lessons Learned

7.3 Future Work

CHAPTER 8

APPENDIX

BIBLIOGRAPHY

[1] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: ().

[2] Haofan Wang et al. *Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks*. Version 2. Apr. 13, 2020. arXiv: arXiv:1910.01279. URL: http://arxiv.org/abs/1910.01279 (visited on 02/16/2023). preprint.

[3] Ramprasaath R Selvaraju et al. "Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization". In: (), p. 9.

[4] Ramprasaath R. Selvaraju et al. *Grad-CAM: Why Did You Say That?* Jan. 25, 2017. arXiv: arXiv:1611.07450. URL: http://arxiv.org/abs/1611.07450 (visited on 02/20/2023). preprint.

[5] Jost Tobias Springenberg et al. *Striving for Simplicity: The All Convolutional Net*. Apr. 13, 2015. DOI: 10.48550/arXiv.1412.6806. arXiv: arXiv:1412.6806. URL: http://arxiv.org/abs/1412.6806 (visited on 11/18/2022). preprint.

[6] Narine Kokhlikyan et al. *Captum: A Unified and Generic Model Interpretability Library for PyTorch*. Sept. 16, 2020. DOI: 10.48550/arXiv.2009.07896. arXiv: arXiv:2009.07896. URL: http://arxiv.org/abs/2009.07896 (visited on 04/04/2023). preprint.

[7] Daniel Smilkov et al. *SmoothGrad: Removing Noise by Adding Noise*. June 12, 2017. DOI: 10.48550/arXiv.1706.03825. arXiv: arXiv:1706.03825. URL: http://arxiv.org/abs/1706.03825 (visited on 11/28/2022). preprint.

[8] Lorenz Richter et al. "VarGrad: A Low-Variance Gradient Estimator for Variational Inference". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 13481–13492. URL: https://proceedings.neurips.cc/paper/2020/hash/9c22c0b51b3202246463e986c7e205df-Abstract.html (visited on 02/20/2023).

[9] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. *Axiomatic Attribution for Deep Networks*. June 12, 2017. DOI: 10.48550/arXiv.1703.01365. arXiv: arXiv:1703.01365. URL: http://arxiv.org/abs/1703.01365 (visited on 03/24/2023). preprint.

[10] Vitali Petsiuk, Abir Das, and Kate Saenko. *RISE: Randomized Input Sampling for Explanation of Black-box Models*. Sept. 25, 2018. arXiv: arXiv:1806.07421. URL: http://arxiv.org/abs/1806.07421 (visited on 02/20/2023). preprint.

[11] Dan Hendrycks et al. *AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty*. Feb. 17, 2020. arXiv: arXiv:1912.02781. URL: http://arxiv.org/abs/1912.02781 (visited on 01/16/2023). preprint.

[12] Jianhua Lin. "Divergence Measures Based on the Shannon Entropy". In: ().

[13] Terrance DeVries and Graham W. Taylor. *Improved Regularization of Convolutional Neural Networks with Cutout*. Nov. 29, 2017. DOI: 10.48550/arXiv.1708.04552. arXiv: arXiv:1708.04552. URL: http://arxiv.org/abs/1708.04552 (visited on 03/27/2023). preprint.

[14] Sangdoo Yun et al. "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, Korea (South): IEEE, Oct. 2019, pp. 6022–6031. ISBN: 978-1-72814-803-8. DOI: 10.1109/ICCV.2019.00612. URL: https://ieeexplore.ieee.org/document/9008296/ (visited on 02/20/2023).

[15] Hongyi Zhang et al. *Mixup: Beyond Empirical Risk Minimization*. Apr. 27, 2018. DOI: 10.48550/arXiv.1710.09412. arXiv: arXiv:1710.09412. URL: http://arxiv.org/abs/1710.09412 (visited on 03/27/2023). preprint.

[16] Devesh Walawalkar et al. *Attentive CutMix: An Enhanced Data Augmentation Approach for Deep Learning Based Image Classification*. Apr. 5, 2020. DOI: 10.48550/arXiv.2003.13048. arXiv: arXiv:2003.13048. URL: http://arxiv.org/abs/2003.13048 (visited on 03/29/2023). preprint.

[17] Geoff French, Avital Oliver, and Tim Salimans. *Milking CowMask for Semi-Supervised Image Classification*. June 5, 2020. DOI: 10.48550/arXiv.2003.12022. arXiv: arXiv:2003.12022. URL: http://arxiv.org/abs/2003.12022 (visited on 03/31/2023). preprint.

[18] Zhun Zhong et al. "Random Erasing Data Augmentation". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.07 (Apr. 3, 2020), pp. 13001–13008. ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v34i07.7000. URL: https://aaai.org/ojs/index.php/AAAI/article/view/7000 (visited on 10/21/2022).

[19] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. "Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection". In: Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 1301–1310. URL: https://openaccess.thecvf.com/content_iccv_2017/html/Dwibedi_Cut_Paste_and_ICCV_2017_paper.html (visited on 03/31/2023).

[20] Krishna Kumar Singh et al. *Hide-and-Seek: A Data Augmentation Technique for Weakly-Supervised Localization and Beyond*. Nov. 6, 2018. DOI: 10.48550/arXiv.1811.02545. arXiv: arXiv:1811.02545. URL: http://arxiv.org/abs/1811.02545 (visited on 03/27/2023). preprint.

[21] Pengguang Chen et al. *GridMask Data Augmentation*. Jan. 13, 2020. DOI: 10.48550/arXiv.2001.04086. arXiv: arXiv:2001.04086. URL: http://arxiv.org/abs/2001.04086 (visited on 03/31/2023). preprint.

[22] Bolei Zhou et al. "Learning Deep Features for Discriminative Localization". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, June 2016, pp. 2921–2929. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.319. URL: http://ieeexplore.ieee.org/document/7780688/ (visited on 02/20/2023).

[23] Jie Qin et al. *ResizeMix: Mixing Data with Preserved Object Information and True Labels*. Dec. 20, 2020. arXiv: arXiv:2012.11101. URL: http://arxiv.org/abs/2012.11101 (visited on 03/29/2023). preprint.

[24] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. "Data Augmentation Using Random Image Cropping and Patching for Deep CNNs". In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.9 (Sept. 2020), pp. 2917–2931. ISSN: 1051-8215, 1558-2205. DOI: 10.1109/TCSVT.2019.2935128. arXiv: 1811.09030 [cs]. URL: http://arxiv.org/abs/1811.09030 (visited on 03/30/2023).

[25] Hiroshi Inoue. *Data Augmentation by Pairing Samples for Images Classification*. Apr. 11, 2018. arXiv: arXiv:1801.02929. URL: http://arxiv.org/abs/1801.02929 (visited on 03/30/2023). preprint.

[26] Jin-Ha Lee et al. "SmoothMix: A Simple Yet Effective Data Augmentation to Train Robust Classifiers". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020, pp. 756–757. URL: https://openaccess.thecvf.com/content_CVPRW_2020/html/w45/Lee_SmoothMix_A_Simple_Yet_Effective_Data_Augmentation_to_Train_Robust_CVPRW_2020_paper.html (visited on 03/29/2023).

[27] *SMOTE: Synthetic Minority Over-sampling Technique | Journal of Artificial Intelligence Research*. URL: https://www.jair.org/index.php/jair/article/view/10302 (visited on 03/31/2023).

[28] Shaoli Huang, Xinchao Wang, and Dacheng Tao. "SnapMix: Semantically Proportional Mixing for Augmenting Fine-grained Data". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.2 (2 May 18, 2021), pp. 1628–1636. ISSN: 2374-3468. DOI: 10.1609/aaai.v35i2.16255. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16255 (visited on 03/31/2023).

[29] Jie Cao et al. "ReMix: Towards Image-to-Image Translation With Limited Data". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 15018–15027. URL: https://openaccess.thecvf.com/content/CVPR2021/html/Cao_ReMix_Towards_Image-to-Image_Translation_With_Limited_Data_CVPR_2021_paper.html (visited on 03/31/2023).

[30] 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). 2018, pp. 364–380. URL: https://openaccess.thecvf.com/content_ECCV_2018/html/NIKITA_DVORNIK_Modeling_Visual_Context_ECCV_2018_paper.html (visited on 10/21/2022).

[31] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. "Puzzle Mix: Exploiting Saliency and Local Statistics for Optimal Mixup". In: *Proceedings of the 37th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, Nov. 21, 2020, pp. 5275–5285. URL: https://proceedings.mlr.press/v119/kim20b.html (visited on 04/04/2023).