



university of  
groningen

faculty of mathematics  
and natural sciences

artificial intelligence

# Proxy Attention : Comparing and Combining Augmentation with Attention

Graduation Project  
(Computational Intelligence and Robotics)

Subhaditya Mukherjee (s4747925)

March 31, 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

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Context and Novelty . . . . .	6
1.2	Motivation . . . . .	6
1.3	Challenges . . . . .	6
1.4	Problem Statement . . . . .	6
1.5	Research Questions . . . . .	6
1.6	Thesis Outline . . . . .	6
<b>2</b>	<b>Background</b>	<b>7</b>
2.1	Interpretability . . . . .	7
2.2	Gradient Based Explanations . . . . .	7
2.3	Augmentation . . . . .	7
2.4	Datasets . . . . .	7
2.4.1	CIFAR 100 . . . . .	7
2.4.2	Stanford dogs . . . . .	7
2.4.3	Imagenette . . . . .	7
2.4.4	ASL . . . . .	7
2.4.5	Food-101 . . . . .	7
<b>3</b>	<b>State of the Art</b>	<b>10</b>
3.1	Gradient Based Explanations . . . . .	10
3.2	Augmentation . . . . .	11
3.3	Architectures . . . . .	12
3.4	Summary and Limitations . . . . .	12
<b>4</b>	<b>Proposed Approach</b>	<b>13</b>
4.1	Design Decisions . . . . .	13
4.2	Hyper Parameters . . . . .	13
4.2.1	Clear Every Step . . . . .	13
4.2.2	Gradient Threshold Considered . . . . .	13
4.2.3	Multiply Weight . . . . .	13
4.2.4	Proxy Steps . . . . .	13
4.2.5	Subset Of Wrongly Classified . . . . .	13
4.2.6	Gradient Method . . . . .	13
4.2.7	Architectures . . . . .	13



<b>5 Implementation</b>	<b>14</b>
5.1 Overview . . . . .	14
5.2 Hyper parameters . . . . .	14
5.2.1 Clear Every Step . . . . .	14
5.2.2 Gradient Method . . . . .	14
5.2.3 Gradient Threshold Considered . . . . .	14
5.2.4 Multiply Weight . . . . .	14
5.2.5 Proxy Steps . . . . .	14
5.2.6 Subset Of Wrongly Classified . . . . .	14
5.3 Data Loading and Pre Processing . . . . .	14
5.3.1 Directory structure . . . . .	14
5.3.2 Label function . . . . .	14
5.3.3 Clearing proxy images . . . . .	14
5.3.4 Encode, Stratify, Kfold . . . . .	14
5.3.5 train and test, val separate . . . . .	14
5.3.6 Augmentations . . . . .	14
5.4 Training Details . . . . .	15
5.5 Grid Search . . . . .	15
5.6 Optimizations . . . . .	15
5.6.1 Mixed Precision . . . . .	15
5.6.2 Gradient Scaling . . . . .	15
5.6.3 No grad . . . . .	15
5.6.4 Batched Proxy step . . . . .	15
5.6.5 Trial Resumption . . . . .	15
5.6.6 Models . . . . .	15
5.7 Gradient Based Methods . . . . .	15
5.8 Proxy Attention . . . . .	15
5.8.1 Callback Mechanism . . . . .	15
5.9 Tensorboard . . . . .	15
5.10 Transfer learning . . . . .	15
5.11 Optimizer . . . . .	15
5.12 LR scheduler . . . . .	15
5.13 Loss function . . . . .	15
5.14 Batch sizer finder . . . . .	15
5.15 Result Aggregation . . . . .	15
5.16 Inference . . . . .	15
<b>6 Evaluation</b>	<b>16</b>
6.1 Metric Based Analysis . . . . .	16
6.2 Visual Based Analysis . . . . .	16
6.3 Summary . . . . .	16
<b>7 Conclusion</b>	<b>17</b>
7.1 Contributions . . . . .	17
7.2 Lessons Learned . . . . .	17
7.3 Future Work . . . . .	17
<b>8 Appendix</b>	<b>18</b>



# LIST OF FIGURES

2.1	CIFAR100 Sample . . . . .	8
2.2	Stanford Dogs Sample . . . . .	8
2.3	Imagenette Sample . . . . .	8
2.4	ASL Sample . . . . .	9



# LIST OF TABLES



## 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

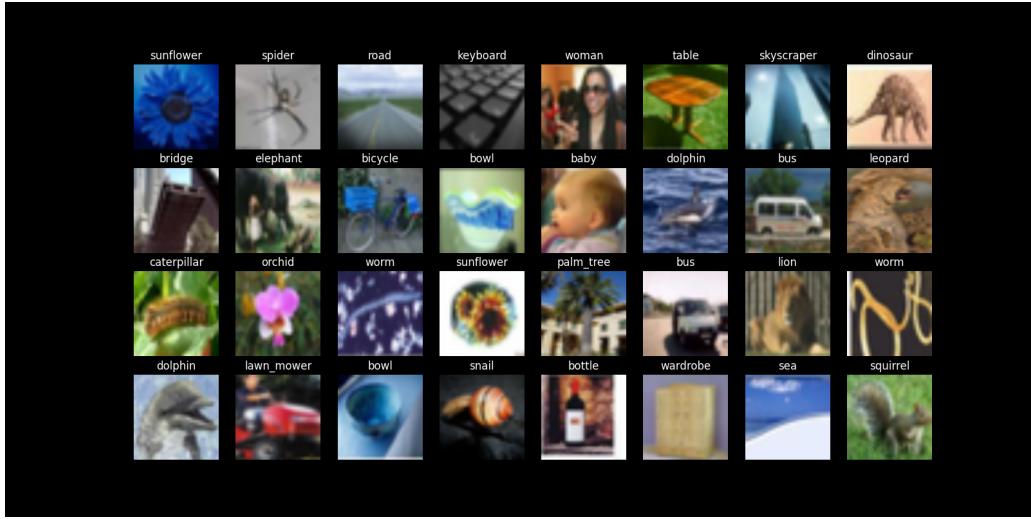


Figure 2.1: CIFAR100 Sample



Figure 2.2: Stanford Dogs Sample

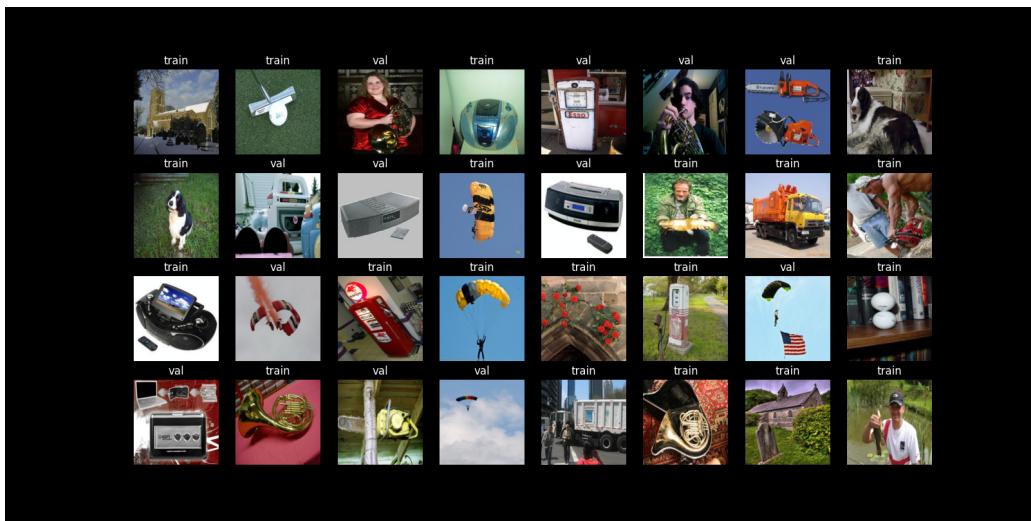


Figure 2.3: Imagenette Sample

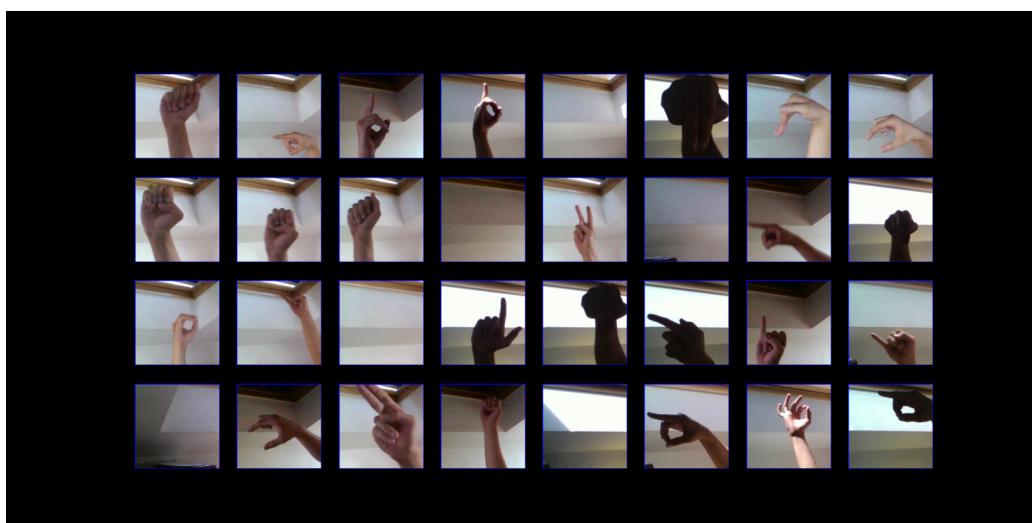


Figure 2.4: ASL Sample

# 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 mathmatically represented as  $L_{ScoreCAM}^c = ReLU(\sum_k w_k^c A^k)$ , 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

Salience Map

Noise Tunnel

Integrated Gradients

Sam Resnet

Conductance

Deep Fool

Deep Lift

Generalizing Adversarial Exp With Gradcam

Shap

Smooth Grad

Smooth Grad Square

Lime

Sp Lime

Summit  
 Rise  
 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

Attributemix

Augmentaiton with curriculum leanring

Another augmentation strategy proposed by [3] 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 [4] 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 [5] 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.

Co mixup Unlike Cutout [5], where the chosen patch is replaced with zero pixels, in CutMix [6] 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  $\tilde{x} = M \odot x_A + (1 - M) \odot x_B ; \tilde{y} = \lambda y_A + (1 - \lambda) y_B$ . 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  $\tilde{x}, \tilde{y}$  is created by combining two other training samples  $x_A, y_A$  and  $x_B, y_B$ . To control the combination ratio  $\lambda$ , a sample from the  $\beta(1, 1)$  distribution is chosen. This combination is quite similar to [7] but differs in the sense that CutMix focuses on generating locally natural images. Building up on [6], Walawalkar et al. propose an alternative method of replacing patches in an image they call Attentive CutMix [8]. 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 background regions that are important for the task while also updating the label information.

In their paper Singh et al. [9] 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.

GridMask

Image Mixing and deletion



Zhang et al. propose another method of data augmentation that uses a CAM [10] 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.

Keep augment

Latent space interpo

Puzzle mix

Randaugment

While Cutout augmentation [5] 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 [11]. 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.

Random distortion

Remix

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 [12] tackles this problem by replacing the patch with a proportional resized version of the selected image. This method is similar to CutMix [6] 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 [13]. 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 multiple 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.

Saliencymix

Sample pairing

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 [14], which builds up on both CutMix [6] and Cutout [5] 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  $\lambda = \frac{\sum_{i=1}^W \sum_{j=1}^H G_{ij}}{WH}$ . 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  $(x_{new}, y_{new}) = (G.xa + (1 - G).xb, \lambda.ya + (1 - \lambda).yb)$

Smote

Snap mix

Spec augment

Visual context Augmentation

### 3.3 Architectures

Resnet 18, 50

VGG

Vision Transformer

### 3.4 Summary and Limitations



## 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

### 5.15 Result Aggregation

### 5.16 Inference



## 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] 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.
- [4] Jianhua Lin. “Divergence Measures Based on the Shannon Entropy”. In: () .
- [5] 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.
- [6] 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).
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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).
- [11] 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).

- [12] 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.
- [13] 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).
- [14] 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](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).