
An Investigation of Image Classification with Learning by Self-Explanation: Class Activation Mapping and Channel Attention Mechanism

Chi-Jui Ho

Department of Electrical and Computer Engineering, UC San Diego
9500 Gilman Dr, La Jolla, CA 92093
chh009@ucsd.edu

Abstract

Image classification is a core problem of computer vision. Although convolutional neural network (CNN) model are superior to conventional frameworks in classification accuracy, they are less transparent than conventional ones. This issue has been alleviated by self-explanation frameworks that leverage the interaction between explainer and audience networks. In this report, we would like to further improve the transparency and accuracy by integrating the channel activation mapping and channel attention mechanism to self-explanation frameworks. Hopefully, the proposed mechanism can deepen our understanding to the operations of deep CNN-based classifiers. Preliminary results of the proposed approach are also provided. The source code is available on <https://drive.google.com/drive/folders/168xp9PxIJ-vTxwMsT4oPiFU5RB3eSc2?usp=sharing>

1 Introduction

Convolutional neural network (CNN) has been commonly applied to image classification. While deep CNN model has dominated recent developments of classification, the low transparency issue makes researchers difficult to figure out how the decisions are made through the CNN model. Unlike hand-craft classifiers, it is difficult for human to understand how feature extraction is completed in CNN-based classifiers. In addition, the selection of network architecture is usually exhausted because of the low transparency of classification policy.

To address these issues, self-explanation mechanism was proposed by Hosseini *et al.* [1]. It consists of two networks, explainer and audience. The explainer network reveals its decision policy of classification, and the audience network guides the selection of network architecture. Through the explanation-driven learning, an accurate and explainable classification process is constructed.

In this project, we plan to integrate two mechanisms, class activation mapping (CAM) and channel attention, to the self-explanation framework. The former has been developed to analyze the features are extracted by CNN-based classifiers, while the latter enhances the classification accuracy by reweighting the significant feature maps. Hopefully, CAM can better interpret explanation network and introducing channel attention mechanism can broaden the selection of network architecture. We believe the integration can deepen our understanding of deep CNN-based classifiers.

2 Review

In this section, we studied the principle of learning by self-explanation and how class activation mapping reflects the decision process of CNN-based classifiers. We also reviewed previous work on channel attention mechanisms.





	ResNet-50	Mixup [47]	Cutout [3]	CutMix
Image				
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4

Figure 1: An example of different regularization approaches.

2.1 Self-Explanation Mechanism

Learning by self-explanation (LeaSE) was inspired by human learning, which is explanation driven [1]. LeaSE consists of four mutually dependent steps and two classifiers, explainer and audience. The explainer is first trained on a dataset to minimize prediction loss and then asked to explain the prediction policy through adversarial attack [2]. Specifically, the pixels are reweighted according to the decision of explainer network. The audience network is then trained on reweighted images and the network architecture is selected by optimizing audience network.

2.2 Class Activation Mapping

To realize features learned by deep CNN model, class specific features are localized by class activation mapping (CAM) [3]. However, CAM is only applicable to the specific network architecture that performs global average pooling right before the softmax layer. To provide a general model interpretation, Selvaraju *et al.* proposed a method called gradient-weighted CAM (Grad-CAM) [4]. GRAD-CAM is applicable to a wide-variety of architectures. In addition, it is able to interpret the network at every convolutional layer, allowing us to observe feature activations at all levels.

2.3 Channel Attention Mechanism

The representational power of CNN-based classifiers can be improved by adopting the channel attention mechanism. The squeeze-and-excitation (SE) block [5] is a pioneering channel attention module that rescales the feature map according to the channel-wise dependency. Wang *et al.* enhanced the learnability and saved the parameters by handling cross-channel interaction without dimensional reduction [6].

3 Proposed Approach

We plan to first improve the regularization method in LeaSE. Moreover, we propose to incorporate Grad-CAM and channel attention mechanism to self-explanation learning. Specifically, Grad-CAM and channel attention integrated to the second and the fourth step of LeaSE. We note that the explanation is achieved through adversarial attack [9], which reweights each pixel according to their significance. Inspired by the superior qualitative result of Grad-CAM interpretation, we plan to replace adversarial attack with Grad-CAM. We believe the integration can make it easier to understand how the decision is made by explainer network and hence make the results more interpretable.

Furthermore, the architecture searching of LeaSE adopts the framework proposed by Liu *et al.* as backbone approach [7]. Nonetheless, channel attention mechanism is not included in its searching space. We plan to first replace the architecture of audience network, ResNet 18 [8], with SE-ResNet18, to see if a higher accuracy can be achieved. We also plan to perform architecture searching while considering channel attention as a component of architecture. Hopefully, the proposed approaches will enable us to have a deeper understanding of image classification by deep CNN model.

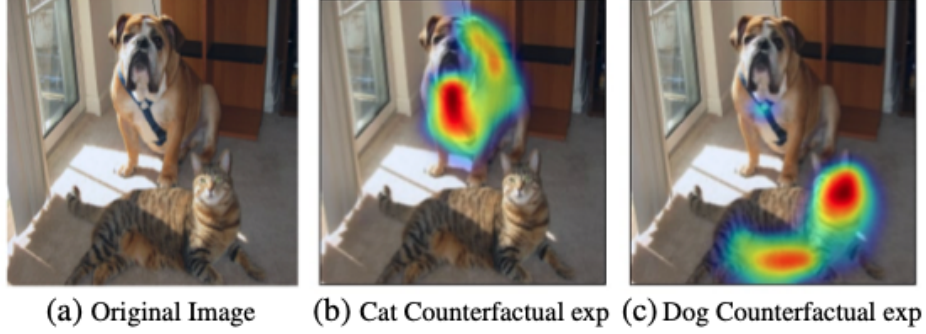


Figure 2: An example of grad-cam visualization.

3.1 Regularization

We first noted that Cutout algorithm was applied to regularize the first training phase of LeaSE. Cutout uses a regional dropout approach to alleviate the over-fitting issue of CNN-based classifiers. It is notable that Cutout ignores the dependency among different classes, and hence other regularization approaches that leverage the relations among classes may yield a higher classification accuracy. We implement two regularization approaches in this project: Mixup and Cutmix.

Both Mixup and Cutmix use regular and mixed images as training data with a random selection. Mixup algorithm generates mixed images through linear interpolation. In other words, images $I_m = \lambda I_x + (1 - \lambda)I_y$ are generated, where I_x and I_y are regular images from the dataset. To determine the ratio of I_x and I_y , a random variable λ is generated through Beta sampling $\lambda \sim \mathcal{B}[\beta, \beta]$. The class label of I_m is also changed from a one-hot vector to a soft label $l_x + (1 - \lambda)l_y$, where l_x and l_y are the class label of I_x and I_y , respectively. This algorithm encourages the classifier to continuously map the images to the latent space, instead of simply clustering them according to the class labels, and hence alleviate over-fitting.

Two random variables are needed in Cutmix: r is uniformly sampled from 0 to 1: $r \sim \mathcal{U}[0, 1]$, and λ is generated through Beta sampling $\lambda \sim \mathcal{B}[\beta, \beta]$. Empirically, the variable β is set as 1. Once $r < 0.5$, the following mechanism is activate to generate a mixed image: a region of I_x is cropped and then pasted to the corresponding positions on I_y . The ratio $r_c = \sqrt{1 - \lambda}$ determines the size of cropped region $(W_c, H_c) = (r_c \times W, r_c \times H)$, where W and H are the width and height of regular images I_x , while W_c and H_c are the width and height of the cropped region. The class label of mixed image l_m is also a linear combination of l_x and l_y : $l_m = \lambda_m l_x + (1 - \lambda_m)l_y$, where $\lambda_m = \frac{H_c \times W_c}{H \times W}$. Compared with Cutout and Mixup, Cutmix leverages regional dropout, interactions among classes, and the usage of whole image regions. An illustration of these algorithms is shown in Fig. 1, which is from the paper for Cutmix.

3.2 Class Activation Mapping

It is an exhausted process to train the four-phases of LeaSE altogether. Compared with plain model training, the loss of LeaSE training converges much slower. Therefore, we introduce CAM to replace the second phase, which applies adversarial attack to find out the significance pixels of image classification.

The target of gradient-based CAM is to generate a heat map that depicts the significant features for image classification. Because Grad-CAM does not require additional model training, it is easier to implement than adversarial attack. To understand how the k^{th} feature map at level m , $A^{k,m}$, contributes to the decision of class label c , we can compute the average gradient

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y_c}{\partial A_{ij}^{k,m}}, \quad (1)$$

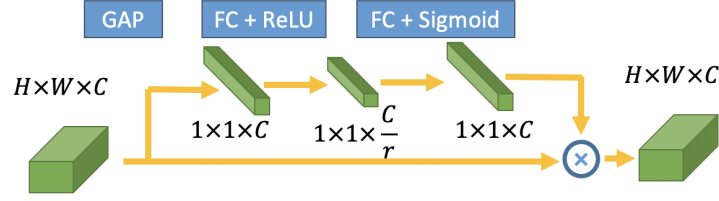


Figure 3: An illustration of SE-block. GAP: global average pooling, FC: fully-connected layer, and ReLU: rectified linear units.

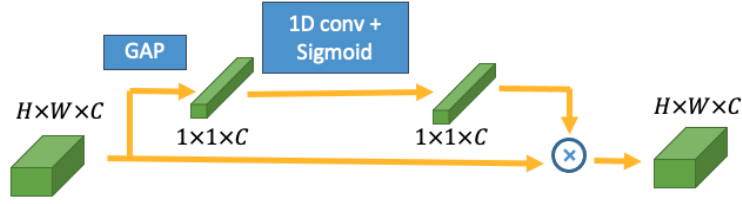


Figure 4: An illustration of ECA layer. GAP: global average pooling and 1D-conv: 1 dimensional convolution.

100 where α_k^c is the contribution of feature map A_k to class c and Z is a normalization constant of global
 101 average pooling. Therefore, the heat map H of class c is obtained by computing a weighted sum of
 102 all feature maps:

$$H_m^c = ReLU\left(\sum_k \alpha_k^c A^{k,m}\right), \quad (2)$$

103 where $ReLU$ is rectified linear unit, a non-linear activation function. Compared with CAM, Grad-
 104 CAM has a wider range of applications because is able to visualize all the layers instead of just the
 105 last layer. Moreover, CAM can only visualize the network that adopts average pooling and linear
 106 classifier, while Grad-CAM is free from this restriction. An example of Grad-CAM visualization is
 107 shown in Fig. 2.

108 We plan to use Grad-CAM to visualize the heat map activated by the the classifier. After the first
 109 training phase, a network with high precision on regular images is obtained. Then, we use Grad-CAM
 110 to generate the heat map of each images for all classes. For instance, in Cifar 10, a total of 10
 111 heat maps are generated for each image. According to these heat maps, the regular image can be
 112 reweighed. Therefore, a total of 10 reweighed images are obtained. We plan to concatenate them
 113 to generate a 30 channel images, called grouped reweighed images, and use the grouped reweighed
 114 images to train a classifier, which is the third phase of LeaSE. Compared with the second phase of
 115 LeaSE, Grad-CAM based reweighing does not need any retraining and hence saves computational
 116 burden in model training.

117 3.3 Channel Attention Mechanism

118 Searching architecture is the fourth step of LeaSE. It leverages different network structures, including
 119 dilated convolution, plain network, and skip connections, to estimate the class label. Specifically, the
 120 network has a flexible block that composed of several parallel operations, and the result is a weighted
 121 sum of the predictions from different operations. We note that the input and output channel of the
 122 operation are the same, which means that channel attention mechanism is able to be added to the
 123 parallel operations.

Channel attention mechanism reweights the significance of feature maps extracted by different channels. It can be achieved by squeeze-and-excitation (SE) or effective channel attention (ECA). An illustration of SE-block is shown in Fig. 3. The branch of SE-block first use a global average pooling to reduce the size of each feature map to 1×1 . Then, two fully-connected layers are used to squeeze and excite the dimension. The final operation of the branch is a sigmoid function, which limits the range of each element to $[0, 1]$. The result of SE-branch is a tensor sized of $1 \times 1 \times C$ that represents the significance of each channel. It rescales the input feature maps, which is sized of $H \times W \times C$, and therefore magnify the significant of key feature maps while downplay the significance of other maps.

The concept of ECA is similar to that of SE-block. An illustratino of it is shown in Fig. 4. It also performs global average pooling to resize feature maps to a size of 1×1 . To avoid losing information in squeezing and leverage the correlations in neighboring feature maps, it performs an 1 dimensional convolution followed by Sigmoid to generate a tensor sized of $1 \times 1 \times C$ that represents the significance of each channel. Similar to SE-block, it rescales the input feature maps according to their significance and correlations to each other.

It is known that both SE-block and ECA layer improves the classification accuracy of plain network and ResNet in image classification. Therefore, we introduce them to the candidate architectures. Hopefully, by leveraging the different choices of architectures in the fourth phase of training, the classification accuracy of LeaSE can be further improved.

4 Preliminary Results on Midterm Report

This section illustrated the results presented in midterm reports, including the classification accuracy under different regularization approaches and how does CAM work in this project.

4.1 Regularization

We tested all the three regularization approaches discussed in Sec. 3.1 on Cifar 10. The preliminary results are shown in Table 1. As we can see, Both Cutmix [10] and Mixup [11] outperforms Cutout in terms of top 1 classification accuracy on validation set. This result indicates that using mixed images in model training alleviates the over-fitting issue and enhance the robustness of CNN-based classifiers to different features. It is also notable that Mixup only outperforms Cutmix by a slight margin. Therefore, more experiments are needed to decide which regularization method is the best choice to be integrated to LeaSE.

4.2 Class Activation Map

According to the trained network discussed in Sec. 4.1, we generate a few heat maps, as shown in Fig. 5. Fig. 5 (a) and (b) shows an example image of dog and a stack of example image and its heat map, respectively. We can see that all the hot spots of heat map concentrates on the body of the dog, which shows that decision policy of the classifier is interpretable. Hopefully, by amplifying the significant features for classification, the proposed algorithm can be an alternate of the second phase of LeaSE training and hence reduces the training burden. However, one concern is that the low resolution of heat map may make some details unavailable to the classifier. Therefore, we believe the regular image should still be a part of input in future implementation.

4.3 Implementation

We found that different networks are used in the files train.py and train_search.py. Therefore, we suppose the model weights trained by the former file is not applicable to the latter. To address this issue, we use the network defined in train_search.py for all the experiments. After obtaining sufficiently high accuracy on train.py, the first phase of LeaSE training, we plan to proceed the LeaSE training based on the trained network.

Table 1: Top 1 classification accuracy under different regularization algorithms

Regularization method	Top 1 classification accuracy (%)
Cutout	80.97
Mixup	83.12
Cutmix	82.86

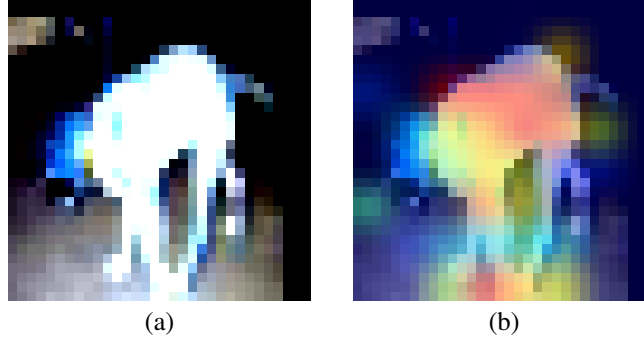


Figure 5: An example of our grad-cam implementation.

169 4.4 Summary of Midterm Report

170 Due to the limited time, this midterm report only presents the results on Cifar 10. We plan to conduct
 171 the experiments on Cifar 100 as well and see if the superiority of Mixup and Cutmix discussed in
 172 Sec. 4.1 still holds. Then, we plan to incorporate Grad-CAM to LeaSE training and see if the second
 173 phase is replaceable. We also plan to test the performance after introducing SE-block as one of the
 174 candidate in architecture searching.

175 5 Experimental Results on Final Report

176 In this section, we discussed the experimental results. We first select a baseline model for classification
 177 and then describe the results of leveraging channel attention modules in LeaSE. We also discussed
 178 the difficulties we faced in implementation in the last subsection.

179 5.1 Baseline Model

180 We adopt ResNet-18 as our baseline classifier. Because the limited time and computational resource,
 181 we only have the results after training ResNet-18 after 20 epochs, which is the same condition to
 182 train other networks in this project. Therefore, although it was reported that ResNet-18 can reach
 183 over 95% classification accuracy on Cifar-10, we can only have a classification accuracy of 72.15%
 184 in this project. In addition to limited epochs, we believe the accuracy can be further improved by
 185 testing different optimizers such as Adam or using a larger batch size. However, due to the limited
 186 memory, the batch size can only be 32, which is four times lower than the suggested value. The
 187 limitation of batch size also results in a time-consuming model training, making it difficult to observe
 188 the classification accuracy in long epochs.

189 5.2 Channel Attention Module

190 We add two moduels, SE-block and ECA module, to the searching field s_3 , which also includes skip
 191 connection and a four-layer convolutional network in each cell. ECA module is another channel
 192 attention approach that was exploited after midterm report. We note that the input and output size
 193 of SE-block should be the same, so it is only added to normal cells but not reduction cells. We
 194 also tested different combinations of regularization methods and channel attentions. The results of
 195 incorporating SE-blocks and ECA are shown in Table 2. We can see that under SE-block, cutmix and
 196 mixup algorithms yield similar performance. However, when ECA is added to the searching field,
 197 cutimx outperforms mixup by over 1% classification accuracy. In other words, using multiple channel

Table 2: Top 1 classification accuracy under different combinations of regularization algorithms and channel attention mechanisms

Regularization method + Network	Top 1 classification accuracy (%)
ResNet-18 (baseline)	80.97
Cutmix	80.97
Mixup	83.12
Cutmix + SE	83.97
Mixup + SE	83.96
Cutmix + SE + ECA	85.61
Cutmix + SE + ECA (10 more epochs)	86.68
Mixup + SE + ECA	83.86

attention modules along with cutmix results in performance gain although the cost of computation should be considered. Therefore, a preferable choice is to use cutmix for registration and both SE-block and ECA in channel attention.

5.3 Training with Class Activation Maps

Instead of leveraging adversarial attack, we use CAM to reweight the images in both training and validation. Specifically, the CAM results are generated by a pre-trained ResNet-18 on cifar 10. For each image, we first generate the heat map for each class label and then stacked all the 10 heat maps. However, the results were inferior to that of using original images for training. A possible reason is that although heat maps show the key features of images, too much information was eliminated. For instance, even the hot spots in Fig. 2 correspond to the main body of the cat, its contour and structure becomes unclear if we notice the heat map only. Nonetheless, due to the ability of CAM to interpret a CNN-based classifier, we believe it is a potential approach to LeaSE, which relies on the interaction between audience and explainer networks. We may not make a full use of CAM-based approaches, but we believe this direction has potential and worth future experiments.

5.4 Searching Space Learning

While searching a desirable architecture in model training, we encountered a serious over-fitting and slow convergence process. Specifically, although the classification accuracy on training data gradually increases, the accuracy on validation set remains around 10%, which is close to random guessing. We believe it is due to the noise added by the attacker overly influences the image, making it nearly uncorrelated to the class label. Hence, the network can only fit the noise but not the image content and a serious overfitting is resulted. We believe that it is a better choice to individually train the attacker instead of training all the modules altogether.

6 Discussions

In this section, we described possible extensions for LeaSE and some issues encountered when developing the proposed approaches to LeaSE.

6.1 Possible Extensions

The experimental results in Sec. 4.1 shows the significance of regularization. A proper selection of regularization method alleviates overfitting and hence improve the robustness of classifier. It was reported that integrating Cutout and shake-shake network design yields even higher accuracy than using Cutout only [10], [12]. An illustration of shake-shake module is shown in Fig. 6. Although architecture of shake-shake design is similar to the one used in this project, it adopts dynamic weights to different operations in a cell in model training. Specifically, the weights among different branches are resampled in both forward and backward propagation. In other words, it leverages both network and data dropout to further enhance the robustness of CNN-based classifiers. Although we did not try incorporating shake-shake in LeaSE, we believe that this would be a potential improvement of it due to their similarities in network architecture.

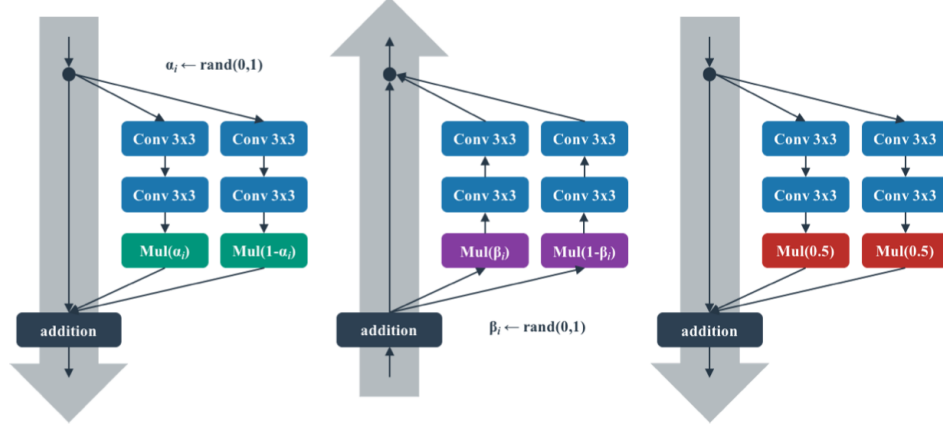


Figure 6: An illustration of shake-shake module. The mechanism for forward, backward, and testing are shown from left to right.

6.2 Implementation Issues

When deploying the proposed ideas in LeaSE, we have encountered the following issues of source code that makes it challenging to seek for further improvements. A possible core dumped may happen if the line for memory summary was not commented out. Moreover, different default hyperparameters are used in the code for architecture train and search. Specifically, they adopt different layers and initialized channels. It is required to figure out the superior choice of the two and hence increases the difficulties in implementation. Another notable thing is that in the code for architecture searching, the number of channels of audience network, ResNet-18, should be set as 10 or 100 instead of the default value, 1000. Moreover, it seems that the optimizer of audience network is in absence, and the trained audience network is not used in validation in every epoch. In addition, it is a little bit confusing that how to perform different steps of LeaSE individually, which we believe is essential to leverage the two network. We believe that more inspiring ideas could be incorporated to LeaSE and yield further improvements if the above issues are considered.

7 Conclusion

Learning by explanation exploits the idea of human learning in training neural networks. In this project, we leverage the channel attention mechanism in learning by self-explanation. The experimental results show that adding channel attention layers to the searching space enhances the learnability of the network and hence improves the classification accuracy. Moreover, the choice of regularization also affects the performance. We found that deploying Cutmix in model training increases the robustness of network to over-fitting. Although the implementation of class activation map does not provide further performance gain, we believe it is a potential approach to LeaSE. LeaSE is a potential approach to address the black-box issue of deep learning and makes the decision policy of CNN-based classifiers more transparent. In future application, we believe that leveraging shake-shake design and Cutmix can further alleviate over-fitting and yields improvements.

References

- [1] R. Hosseini and P. Xie, "Learning by Self-Explanation, with Application to Neural Architecture Search," in *arxiv 2012.12899*, 2020.
- [2] I. J. Goodfellow, J. Shlens, J., and C. Szegedy, "Explaining and harnessing adversarial examples," in *arXiv:1412.6572*, 2014.
- [3] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Learning Deep Features for Discriminative Localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921-2929, doi: 10.1109/CVPR.2016.319.

- 266 [4] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations
267 from Deep Networks via Gradient-Based Localization," in *IEEE International Conference on Computer Vision*,
268 2017, pp. 618-626, doi: 10.1109/ICCV.2017.74.
- 269 [5] J. Hu, L. Shen and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern*
270 *Recog.*, 2018, pp. 7132–7141.
- 271 [6] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep
272 convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 11534–11542.
- 273 [7] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable Architecture Search," in *International Conference*
274 *on Learning Representations*, 2019
- 275 [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf.*
276 *Comput. Vis. and Pattern Recognit.*, 2016, pp. 770–778.
- 277 [9] K. Ren, T. Zheng, Z. Qin, and X. Liu. "Adversarial attacks and defenses in deep learning." *Engineering* 6, no.
278 3 (2020): 346-360.
- 279 [10] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, Y, "Cutmix: Regularization strategy to train strong
280 classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer*
281 *Vision*, 2019, pp. 6023-6032.
- 282 [11] H. Zhang, C. Moustapha, D. N. Yann, and L.-P. David, "mixup: Beyond empirical risk minimization," in
283 *arXiv preprint arXiv:1710.09412*, 2017.
- 284 [12] X. Gastaldi, "Shake-shake regularization," in *arXiv preprint arXiv:1705.07485*, 2017.