Analysis of Image Augmentation Methods on Different Types of Learning Problems

Alekhya Dronavalli
20204017
Université de Montréal
alekhya.dronavalli@umontreal.ca

Krishna Maneesha Dendukuri
20213807
Université de Montréal
krishna.maneesha.dendukuri@umontreal.ca

Prishruit Punia
20178903
Université de Montréal
prishruit.punia@umontreal.ca

Rakshit Shetty
20213127
Université de Montréal
rakshit.shetty@umontreal.ca

Timothy Lee
20213311
Université de Montréal
timothy.lee@umontreal.ca

Yassir Mamouni
20179687
Université de Montréal
yassir.mamouni@umontreal.ca

Abstract

Image classification is a classic machine learning task and remains an on-going research topic. Many different techniques have been developed over the years to tackle different challenges associated with this task. The primary motivation of this project is to explore and compare several image augmentation techniques and their combinations in different learning settings. The goal of this study is to establish a simple library/repository that contains some of the more recent image augmentation methods and evaluate their individual and combined effect on model performance and explore their interpretability.

The effect of both simple and advanced augmentation methods are investigated and compared. The main experiments were performed on the CIFAR-10 dataset and used a ResNet-20 and ALL-CNN architecture. The results suggest that for low resolution dataset such as CIFAR-10, simple augmentation methods can outperform the advanced augmentation methods presented in the literature. More interestingly, combining different augmentation methods can improve the model performance. The git repository can be found here and the WandB logs here.

1 Introduction

Convolutional neural network (CNN) models are one of the most popular and effective method in solving various computer vision problems such as image classification, object detection, and image segmentation. However, ensuring these CNN architectures are capable of achieving a high generalization performance remains to be a challenging task, especially when the amount of available labelled data is low. Data augmentation is a one of the most popular techniques employed to improve model generalization performance and artificially increase the available labelled data.

Data Augmentation is a technique used to increase the amount of data by adding modified copies of existing data or newly created synthetic data based on existing data. The current literature has several data augmentation methods that are proposed to improve the model performance on

Preprint. Under review.

various settings, models and types of datasets. Combining different advanced data augmentation techniques and applying them on low resolution datasets is a relatively overlooked field. In this paper, the observations on the effect of various data augmentation techniques applied independently and combined are presented. The scenarios in which the techniques improved the model generalization performance were analyzed and discussed. The primary objective is to explore and compare individual data augmentation techniques and the combinations of different techniques in supervised and semi-supervised learning settings. A variation of performing data augmentation was explored where the original data is concatenated along with the augmented data.

2 Literature Review

Many different augmentation methods have been proposed to tackle challenges associated with supervised and semi-supervised learning. Among the proposed methods, a vast majority of them are dedicated towards overcoming one of the biggest obstacles in this type of learning task, utilizing the available labeled and unlabeled data efficiently to achieve better generalization. The studies performed in [1] explore the ensembles of deep learners on the data level by adding images generated by different simple data augmentation techniques. The results show that the robustness of the CNNs have increased and that data augmentation methods can improve model generalization. Other work has shown that the increase in robustness is a result of image augmentation methods acting as an effective noise injection technique [10] or acting as a regularization technique for the learning model [2].

The methods investigated include basic augmentation methods such as translation, scaling, and etc. outlined in [1] and [2], advanced augmentation methods such as Cutmix [4], Mixup [4], and Cutout [6], automatic methods such as Randaugment [8] and TrivialAugment [9]. The following paragraphs provide a brief description of the methods investigated.

Cutout is an augmentation method that simulates occlusion where a patch or patches of the original image is removed. This technique is equivalent to applying dropout at the input space and forces the learning model to take the entire image into consideration rather than focusing solely on key features. Similar to Cutout, Cutmix augments the image by removing a patch or patches from the original image. However, in this method, the patch is then replaced by a patch or patches from another randomly selected image in the dataset. Instead of erasing or replacing image patches, Mixup mixes the pixel values of two images and forces the model to learn the linear interpolations between the two images. It is worth noting that models trained with this method have shown robustness against adversarial attacks. The basic augmentation methods investigated in [13] and [14] involve learning representations that are discriminative between images and at the same time, invariant on geometric transformations.

RandAugment and TrivalAugment are both automatic augmentation methods that randomly selects a set of augmentations from their reservoir and applies the augmentation to the original image following a strength value. The reservoirs are primarily composed of basic augmentation techniques. The main difference between the two methods is that RandAugment applies a tuned number of augmentations from a search space with a fixed strength. On the other hand, Trivial Augment samples a single augmentation and a strength value from a search space, hence making it free from any hyperparameters.

Previous literature has investigated the effectiveness of utilizing various techniques of ensembling different augmentation methods and have shown success in increasing model accuracy and stability [1] [10]. Nevertheless, to the knowledge of the authors, there was no previous work performed on applying advanced augmentation methods concurrently to different portions of the same dataset and explicitly evaluating the effect of different combination ratios. In addition, there is a gap in the literature on exploring the performance of ensembling advance image augmentation techniques and class based accuracy. Therefore in this study we have explored questions like (1) which augmentation technique individually performs better in the Supervised and Semi-Supervised settings, (2) does the combination of these techniques outperform the best individual techniques, further (3) does the performance of a method change when we switch between the settings and finally (4) how do the combinations perform compared to their individual components and do they complement each other.

3 Methods

This section describes the model architecture, the setup for various augmentation configurations, and the learning problem investigated. CIFAR-10 is chosen as the dataset for this study because the likelihood of preserving the label post-augmentation is higher for most methods, when compared to any digits dataset.

3.1 Augmentation Methods

Table 6 in the appendix presents the details of how the augmentation methods are applied. Note that PyTorch's built-in implementation of various augmentation methods are used when applicable. All augmentation methods would generate a new image for each labeled data. The corresponding hyperparameters are uniformly sampled between the specified interval. For methods that require the use of mask or padding, the masked or padded pixels would have an RGB value of [0, 0, 0]. Table 7 contains the list of augmentation configurations explored. The values in the cells indicate the percentage of the original data that is augmented and the corresponding methods.

Config0 is the baseline configuration where none of the data is augmented. Config1 to Config10 are configurations where only a single augmentation method is applied at a time. These configurations comprise of both; the basic and the advanced augmentation methods. The basic methods include Flip (Horizontal and Vertical), Translation, Rotation, Contrast Brightness and Gaussian Blur. Whereas, the advanced methods include methods such as Cutout, Cutmix, Mixup and the Automatic Augmentation methods such as RandAugment and TrivialAugment.

A mixture of augmentation methods are investigated through Config11 to Config26. Config27 and Config28 are created to investigate the model performance when combining the top scoring basic augmentation methods and the combination of the top scoring basic and advanced augmentation methods respectively.

The procedure of applying data augmentation in these experiments is different from the procedure presented in previous literature on advanced augmentations. Instead of training the model with augmented data only, the model is trained with both the original data and augmented data. The primary motivation behind this procedure is the view that data augmentation is a suitable strategy to artificially increase the number of training samples. Apart from this, it is also common to apply a combination of augmentation methods sequentially on every image in the dataset. This if done with generators (on the fly) can be less time consuming. However, our experiments explore different ratios of datasets being transformed by different augmentations. This can ensure better parallelism.

3.2 Network Architectures

Two different network architectures are used, ResNet-20 and ALL-CNN [16]. SGD with a learning rate of 0.1, momentum of 0.9, and a weight decay of 0.0005 is used as the optimizer. A learning rate scheduler reduces the learning rate by a factor of 10 at half of the total number of epochs and at three quaters of the total number of epochs. The input data for the ResNet-20 model has a batch size of 120 (increased to 240 after data augmentation) and the ALL-CNN model has a batch size of 256 (increased to 512 after data augmentation). The ResNet-20 models were trained for 160 epochs and the ALL-CNN models were trained for 175 epochs.

- ResNet is one of the most pivotal architecture for image recognition. It is principally composed of Convolutional Layers and "skip connections" that allows it to bypass non linearity. ResNet-20 is an extended Resnet-18 proposed by the authors specifically for CIFAR-10. It is noted that the Pytorch implementation of the ResNet-18 model is designed for ImageNet and is overly powerful for the CIFAR-10 datatset.
- All convolutional network, ALL-CNN is an architecture that achieved good results by using only convolutional layers and no fully connected layers on various object recognition datasets including CIFAR10. Fundamentally, the network replaces max-pooling layers by convolutional layers with increased stride. Implementing a simple model such as ALL-CNN enables comparison between different models.

Table 1: Summary of the 2 architectures wrt CIFAR-10

	Resnet-20	ALL-CNN-C
Main Feature	Skip Connections	All conv. layers
Number of Layers	20	12
Number of Parameters	0.27 M	1.3 M

3.3 Learning Tasks

The performance of different configurations are explored under two different types of setting, supervised learning and semi-supervised learning. In the supervised learning setting, all of the labeled training data are used to train the model. As described above, each augmentation would generate one new image and thus increase the total number of labeled training data from 50,000 to 100,000. To maintain the same dataset size, the benchmark configuration that uses no augmented data would be trained using 2 fold of the original training data. Figure 2 below shows the general procedure used for supervised learning.

The training data is divided into labeled and unlabeled data in the semi-supervised learning setting. 20 percent (10,000) of the samples are flagged as labeled and the remaining flagged as unlabeled. The training procedure is completed in three different steps. Augmentation is performed to the labeled samples to train a labeler model. The labeler model is then used to produce pseudo labels for the unlabeled samples. Lastly, augmentation is performed on the labeled samples and with the pseudo labeled samples to train the final classification model. Figure 1 below shows the general procedure used for semi-supervised learning.

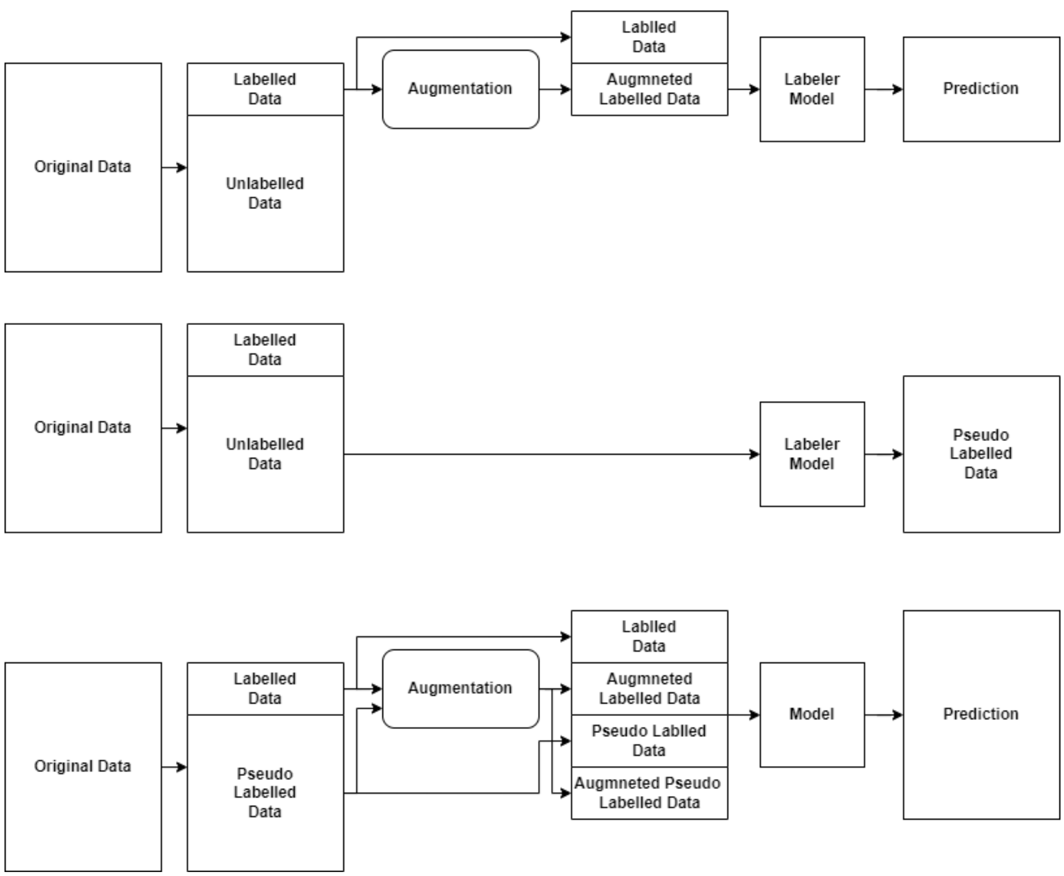

Figure 1: Semi-supervised Learning Training Procedure

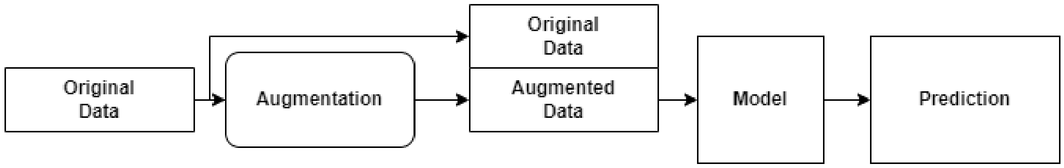

Figure 2: Supervised Learning Training Procedure

4 Results

Table 8 and Table 11 in the appendix presents all the obtained results using the ResNet-20 and the ALL-CNN models. For the ResNet-20 model, the individual augmentation methods that led to the best performing model are shown in Table 2. The top performing configurations for supervised and semi-supervised learning are shown in Table 3 and Table 5 respectively.

Config2, horizontal and vertical translation outperformed all of the investigated methods in both the learning settings; supervised and semi-supervised. This includes the case where basic methods were complemented by advanced methods. In general, it is observed that augmentation methods that score higher in supervised learning would also score higher in semi-supervised learning. Other than that, the results show that combining different augmentation methods usually lead to an improved performance compare to applying the methods independently.

Table 2: Top Performing Individual Augmentation Configurations

ID	Augmentation Description	Supervised Acc			Semi-Supervised Acc		
		Train	Valid	Test	Train	Valid	Test
Config2	100% horizontal and vertical translation.	100	90.84	91.58	100	79.04	80.49
Config3	100% rotations.	100	89.42	89.77	99.7	75.54	75.81
Config7	100% Cutmix.	99.99	88.72	89.34	99.49	71.78	71.87
Config8	100% Mixup.	100	88.8	89.18	100	69.9	70.47
Config1	100% horizontal and vertical flips.	100	89.2	88.95	100	71.62	71.52
Config0	**Baseline, without augmentation.**	**100**	**86.56**	**87.39**	**100**	**67.46**	**68.5**

Table 3: Top Performing Augmentation Configurations - Supervised

ID	Augmentation Description	Supervised Acc		
		Train	Valid	Test
Config2	100% horizontal and vertical translation.	100	90.84	91.58
Config11	20% horizontal and vertical flips, 20% horizontal and vertical translation, 20% rotations, 20% contrast and brightness adjustment, and 20% gaussian blur.	100	90.22	90.71
Config20	80% Cutmix and 20% Mixup.	100	89	90.25
Config23	20% Cutout, 60% Cutmix,	100	89.84	90.17
Config3	100% rotations.	100	89.42	89.77
Config0	**Baseline, without augmentation.**	**100**	**86.56**	**87.39**

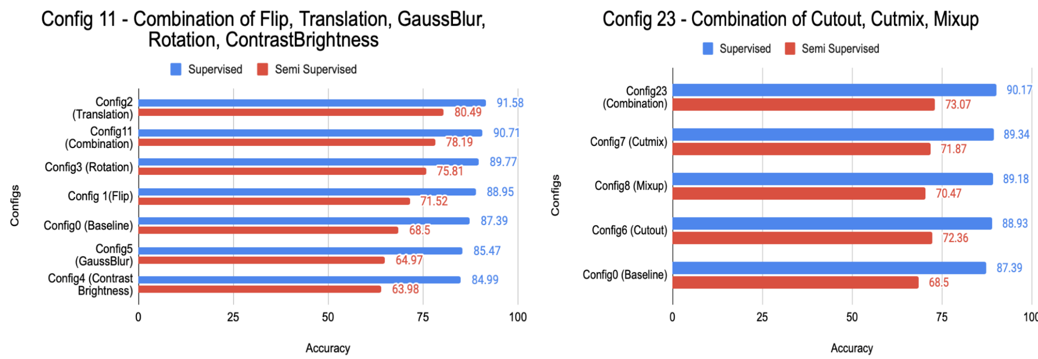

Figure 3: Best Combinations vs Its Individual Components

From Figure 3, we can see that a few configurations like config11, outperforms most of the simple individual techniques like rotation and flip, furthermore we can also infer that each individual component of a combination is either contributing positively(Rotation) or negatively(Gaussian Blur) with respect to the baseline, these analysis help in understanding which augmentations could be beneficial for our task. Moreover, we have also observed that all the advanced image augmentation techniques except RandAugment and TrivialAugment, have positively affected the model performance whereas simple techniques like gaussian blur and contrast and brightness had negative impact for both Supervised and Semi-Supervised settings Table 8.

There has not been a clear consensus on the ratio of original dataset to the transformed dataset, that is effective for a model. Considering a dataset of size N, our experiments use a transformed dataset of size 2N. While this raises an issue of more space and computational complexity, it was observed that the performance boost in test accuracy was very limited when compared to the case with transformed dataset size of size N (using augmented data only). Due to a reduced batch size, the learning rate was lowered as well to make a fair comparison. However, this holds for a limited number of combinations explored in Supervised Learning with Resnet-20. We can obtain a more general view by running similar experiments for the rest of the combinations and learning settings.

Explainability : To observe the regions of the image the model focuses on while making the decision, we have explored methods for producing activation maps from the last convolution layer of the ResNet-20 network. From our experiments we observed that the method LayerCAM[18] produced better quality activation maps compared to methods like GradCAM[17]. From Figure 4, we can observe multiple things, firstly we see that in the supervised setting for many classes like Airplane and Automobile, config 11 focuses on the main object whereas config 2, sometimes focuses on the neighbouring environment, this can also be verified by looking at the per class accuracies of the respective config's from Table 4 where in the supervised setting for classes Airplane and Automobile config 11 performs slightly better than config 2 . Moreover, we can also observe that as we go from the supervised to semi-supervised setting our class activations get further dispersed from the main object this could be because of the lack of labelled data in this setting. We can also observe the difference in how both the methods behave when they switch from the supervised to the semi supervised setting, config 11 which was performing better on the 2 classes (i.e Airplane and Automobile), is now performing worse. by taking a look at these class activations we can also conclude that different augmentation methods lead to different regions of the image being focused at while making the decision.

Table 4: Per Class Accuracy (first 5) of the Top 2 Augmentation Methods

Classes	Airplane	Automobile	Bird	Cat	Deer
Supervised Config_2	92.4	95	88.3	80.8	93.4
Supervised Config_11	92.8	95.6	86.6	79.4	91.5
Semi-Supervised Config_2	84.4	91.6	68.1	65.4	75.6
Semi-Supervised Config_11	79.8	88.9	67.3	56.9	72.4

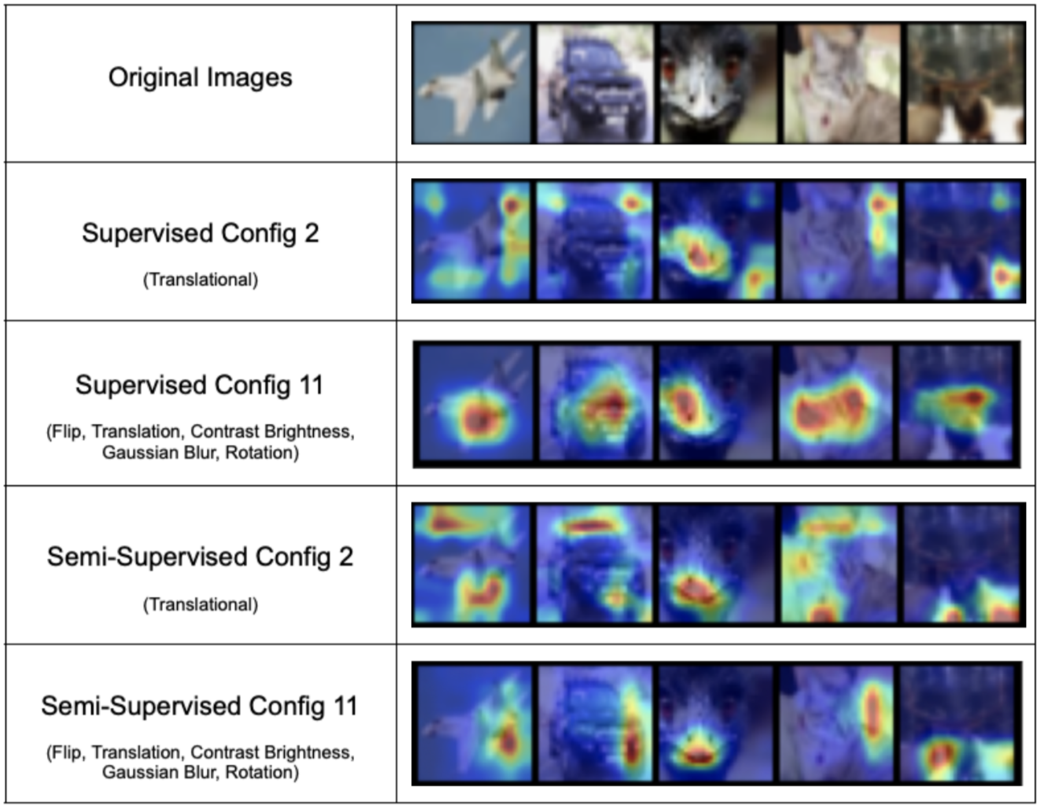

Figure 4: Activation Maps Generated By using LayerCAM[18] for Top 2 Augmentation Methods

Table 5: Top Performing Augmentation Configurations - Semi-supervised

ID	Augmentation Description	Semi-Supervised Acc		
		Train	Valid	Test
Config2	100% horizontal and vertical translation.	100	79.04	80.49
Config11	20% horizontal and vertical flips, 20% horizontal and vertical translation, 20% rotations, 20% contrast and brightness adjustment, and 20% gaussian blur.	99.94	77.86	78.19
Config3	100% rotations.	99.7	75.54	75.81
Config12	20% Cutout and 80% Cutmix.	99.6	72.66	73.4
Config23	20% Cutout, 60% Cutmix,	99.9	71.96	73.07
Config0	**Baseline, without augmentation.**	**100**	**67.46**	**68.5**

5 Conclusions

The results show that individual advanced image augmentation methods do not outperform the top simple augmentation methods. While applying a combination of different augmentation methods typically lead to superior model performance than applying the augmentation methods individually, none of the combined techniques were able to outperform the top scoring simple augmentation method given the experiment conditions. This suggests that different information is gained through different methods and some combinations may allow the methods to complement each other, this can also be seen from the generate class activation maps (Figure 4) that different methods lead to learning of different parts and regions to focus on.

It is noted that the reported performance gain of these advanced augmentation methods are always applied in addition to the top performing simple augmentation methods identified through this experiment. This enhances the initial question of how effective advanced augmentation methods truly are and suggests that the advanced augmentation methods may not be as effective as claimed. To maintain alignment with previous literature, it is possible that the advanced augmentation methods are more effective when applied on higher resolution data and that the model is only trained with augmented data.

6 Future Work

Training the model with augmented data only is a set of experiments that can be performed to further investigate the effectiveness of advanced augmentation methods. To maintain the same batch size of the training data, each labelled data would generate two augmented data and only the augmented data would be used. The current experiments were conducted using the ResNet-20 and the ALL-CNN models. While the general trend appears to align, some performance differences were observed. It would be prudent to repeat the experiments with different model architectures to determine if the same patterns appear. Similarly, utilizing different dataset would provide additional insights into the effect of different augmentation methods. The effect of the amount of augmented data is another topic worthwhile for future investigation. To determine if utilizing augmented data would reach a point of diminishing return and whether that point is consistent across all methods. Additional effort will be spent on more in-depth analysis of the results obtained through the visual explanation techniques. The effect of different augmentation methods on alternative learning settings such as few-shot learning or self-supervised learning also remain as topics to be investigated.

7 Contributions

Name	Tasks
Alekhya Dronavalli	- Prepared Presentation. - Supervised and semi-supervised learning setting implementation. - Investigated the implementation of ALL-CNN model. - Basic ALL-CNN model implementation. - Performed comparision, analysis and derived inferences from the results of two models. - Final report.
Krishna Maneesha Dendukuri	- Advanced augmentation method implementations - ResNet-20 implementation and hyper-parameter tuning. - Analysis of results and Explored additional tests & experiments - Milestone reports - Final report.
Prishruit Punia	- Prepared Presentation - Explainability : GradCAM and LayerCAM visualization methods - Final Report. - Formulated the Research Questions Explored in the study. - Performed analysis and derived inferences from the results.
Rakshit Shetty	- ALL-CNN models implementation. - Evaluation Metrics. - Experiment logging in WANDB. - Investigated potential PyTorch Lightning codebase.
Timothy Lee	- Basic Augmentation Methods. - Supervised and semi-supervised learning implementation. - ResNet-20 hyperparameter tuning. - Milestone reports. - Presentation delivery - Final report.
Yassir Mamouni	- ALL-CNN models implementation and hyperparameter tuning. - ResNet20 implementation. - Investigated implementation of few-shot learning. - Presentation delivery

8 References

[1] Nanni, Loris, et al. "Feature transforms for image data augmentation." arXiv preprint arXiv:2201.09700 (2022)

[2] Hernández-García, Alex, and Peter König. "Data augmentation instead of explicit regularization." arXiv preprint arXiv:1806.03852 (2018)..

[3] Gidaris, Spyros, Praveer Singh, and Nikos Komodakis. "Unsupervised representation learning by predicting image rotations." arXiv preprint arXiv:1803.07728 (2018).

[4] Naveed, Humza. "Survey: Image mixing and deleting for data augmentation." arXiv preprint arXiv:2106.07085 (2021).

[5] Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." Advances in Neural Information Processing Systems 32 (2019).

[6] DeVries, Terrance, and Graham W. Taylor. "Improved regularization of convolutional neural networks with cutout." (2017).

[7] Sohn, Kihyuk, et al. "Fixmatch: Simplifying semi-supervised learning with consistency and confidence." Advances in Neural Information Processing Systems 33 (2020): 596-608.

[8] Cubuk, Ekin D., et al. "Randaugment: Practical automated data augmentation with a reduced search space." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020.

[9] Müller, Samuel G., and Frank Hutter. "Trivialaugment: Tuning-free yet state-of-the-art data augmentation." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[10] Xie, Qizhe, et al. "Unsupervised data augmentation for consistency training." Advances in Neural Information Processing Systems 33 (2020): 6256-6268.

[11] Wang, Haofan, et al. "Score-CAM: Score-weighted visual explanations for convolutional neural networks." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020.

[12] Wen, Yeming, et al. "Combining ensembles and data augmentation can harm your calibration." (2020).

[13] Alexey Dosovitskiy, et al. "Discriminative unsupervised feature learning with convolutional neural networks". In Advances in Neural Information Processing Systems, 2014.

[14] Pulkit Agrawal, et al. "Learning to see by moving". In Proceedings of the IEEE International Conference on Computer Vision, 2015.

[15] Zhang, Hongyi, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. "mixup: Beyond empirical risk minimization." arXiv preprint arXiv:1710.09412 (2017).

[16] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller : Striving for Simplicity: The All Convolutional Net

[17] Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." Proceedings of the IEEE international conference on computer vision. 2017.

[18] Jiang, Peng-Tao, et al. "Layercam: Exploring hierarchical class activation maps for localization." IEEE Transactions on Image Processing 30 (2021): 5875-5888.

9 Appendix

Table 6: List of augmentations and implementation details

ID	Method	Implementation Details
A1	Horizontal and vertical flips	Apply horizontal flip, a vertical flip, or both at the same time with equal probability.
A2	Horizontal and vertical translation	Apply horizontal translation, a vertical translation, or both at the same time with equal probability.The image is shifted by a pixel value between [-16, 16].
A3	Rotations	Apply rotation in the range of [-180, 180].
A4	Contrast and brightness adjustment	Apply contrast adjustment, brightness adjustment, or both at the same time with equal probability. Contrast adjustment is performed using the adjust_contrast function in PyTorch, the 'contrast_factor' is selected to be in the range of [0.5, 1.5]. Brightness adjustment is performed using the 'adjust_brightness' function in PyTorch, the 'brightness_factor' is selected to be in the range of [0.5, 1.5].
A5	Gaussian blur	Operation is performed using the 'gaussian_blur' function in PyTorch, the 'kernel_size' is selected in the range of [1,16].
A6	Cutout	Apply a single rectangular mask to the image. The portion of the mask that lies outside the image is ignored. Randomly select a pixel within the image, this pixel represents the center of the mask. Select a value in the range of [1, 8], this value times 2 represents the width of the mask.Select a value in the range of [1, 8], this value times 2 represents the height of the mask.
A7	Cutmix	Apply a single rectangular mask to the image. The portion of the mask that lies outside the image is ignored. Fill the masked area with pixels located at the same area from another randomly selected image. The final label is adjusted based on the patch size. Randomly select a pixel within the image, this pixel represents the center of the mask. Select a value in the range of [1, 8], this value times 2 represents the width of the mask. Select a value in the range of [1, 8], this value times 2 represents the height of the mask.
A8	Mixup	Select a random image and overlay the two images. A strength parameter is used to adjust the pixel values of the two images (x_new = S*x_i + (1-S)*x_j). The final label is adjusted based on the patch size. The strength parameter is selected in the range of [0, 1].
A9	RandAugment	Utilizing the default parameters implemented in PyTorch.
A10	TrivalAugment	Utilizing the default parameters implemented in PyTorch.

Table 7: List of augmentation configurations

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Config0	-	-	-	-	-	-	-	-	-	-
Config1	100	-	-	-	-	-	-	-	-	-
Config2	-	100	-	-	-	-	-	-	-	-
Config3	-	-	100	-	-	-	-	-	-	-
Config4	-	-	-	100	-	-	-	-	-	-
Config5	-	-	-	-	100	-	-	-	-	-
Config6	-	-	-	-	-	100	-	-	-	-
Config7	-	-	-	-	-	-	100	-	-	-
Config8	-	-	-	-	-	-	-	100	-	-
Config9	-	-	-	-	-	-	-	-	100	-
Config10	-	-	-	-	-	-	-	-	-	100
Config11	20	20	20	20	20	-	-	-	-	-
Config12	-	-	-	-	-	20	80	-	-	-
Config13	-	-	-	-	-	50	50	-	-	-
Config14	-	-	-	-	-	80	20	-	-	-
Config15	-	-	-	-	-	20	-	80	-	-
Config16	-	-	-	-	-	50	-	50	-	-
Config17	-	-	-	-	-	80	-	20	-	-
Config18	-	-	-	-	-	-	20	80	-	-
Config19	-	-	-	-	-	-	50	50	-	-
Config20	-	-	-	-	-	-	80	20	-	-
Config21	-	-	-	-	-	20	20	60	-	-
Config22	-	-	-	-	-	20	40	40	-	-
Config23	-	-	-	-	-	20	60	20	-	-
Config24	-	-	-	-	-	40	20	40	-	-
Config25	-	-	-	-	-	40	40	20	-	-
Config26	-	-	-	-	-	60	20	20	-	-
Config27	30	40	30	-	-	-	-	-	-	-
Config28	10	10	10	-	-	-	56	14	-	-

Table 8: Augmentation configurations results for Resnet20

ID	Augmentation Description	Supervised Acc			Semi-Supervised Acc		
		Train	Valid	Test	Train	Valid	Test
Config0	Baseline, without augmentation.	100	86.56	87.39	100	67.46	68.5
Config1	100% horizontal and vertical flips.	100	89.2	88.95	100	71.62	71.52
Config2	100% horizontal and vertical translation.	100	90.84	91.58	100	79.04	80.49
Config3	100% rotations.	100	89.42	89.77	99.7	75.54	75.81
Config4	100% contrast and brightness adjustment.	100	83.84	84.99	100	63.44	63.98
Config5	100% gaussian blur.	100	86.2	85.47	100	64.34	64.97
Config6	100% Cutout.	100	88.84	88.93	100	71.58	72.36
Config7	100% Cutmix.	99.99	88.72	89.34	99.49	71.78	71.87
Config8	100% Mixup.	100	88.8	89.18	100	69.9	70.47
Config9	100% RandAugment.	73	65.78	65.83	68.53	52.34	52.64
Config10	100% TrivialAugment.	41.37	38.9	37.6	59.99	36.02	37.13
Config11	20% horizontal and vertical flips, 20% horizontal and vertical translation, 20% rotations, 20% contrast and brightness adjustment, and 20% gaussian blur.	100	90.22	90.71	99.94	77.86	78.19
Config12	20% Cutout and 80% Cutmix.	99.99	89.02	89.44	99.6	72.66	73.4
Config13	50% Cutout and 50% Cutmix.	100	89.28	90.16	99.91	71.54	72.08
Config14	80% Cutout and 20% Cutmix.	87.58	81.08	89.75	100	70.84	71.14
Config15	20% Cutout and 80% Mixup.	100	88.44	88.76	100	69.94	71.3
Config16	50% Cutout and 50% Mixup.	100	88.54	88.55	100	69.44	70.27
Config17	80% Cutout and 20% Mixup.	100	88.44	89.07	100	69.1	70.88
Config18	20% Cutmix and 80% Mixup.	100	89.52	89.72	100	70.92	71.12
Config19	50% Cutmix and 50% Mixup.	100	89.28	89.74	99.97	71.06	71.93
Config20	80% Cutmix and 20% Mixup.	100	89	90.25	99.86	71.64	72.31
Config21	20% Cutout, 20% Cutmix, and 60% Mixup.	100	89.06	89.69	99.99	71.32	72.28
Config22	20% Cutout, 40% Cutmix, and 40% Mixup.	100	89.34	89.65	100	72.46	72.47
Config23	20% Cutout, 60% Cutmix, and 20% Mixup.	100	89.84	90.17	99.9	71.96	73.07
Config24	40% Cutout, 20% Cutmix, and 40% Mixup.	100	89.12	89.61	100	70.38	70.46
Config25	40% Cutout, 40% Cutmix, and 20% Mixup.	100	89.46	89.67	99.97	71.52	71.98
Config26	60% Cutout, 20% Cutmix, and 20% Mixup.	100	89.6	89.29	100	70.86	71.54
Config27	30% horizontal and vertical flips, 40 % horizontal and vertical translation, and 30% rotations.	100	90.86	90.9			
Config28	10% horizontal and vertical flips, 10 % horizontal and vertical translation, 10% rotations, 56% Cutmix, and 14% Mixup.	100	90.4	90.81			

Table 9: Augmentation configurations results for ALL-CNN

ID	Augmentation Description	Supervised Acc			Semi-Supervised Acc		
		Train	Valid	Test	Train	Valid	Test
Config0	Baseline, without augmentation.	89.49	75.28	73.38	88.54	61.94	10
Config1	100% horizontal and vertical flips.	70.81	68.98	67.69	58.71	55.52	10
Config2	100% horizontal and vertical translation.	83.40	77.86	77.6	60.66	55.7	10
Config3	100% rotations.	87.36	79.36	77.89	55.48	53.16	10
Config4	100% contrast and brightness. adjustment.	98.83	68.46	68.76	72.43	56.44	10
Config5	100% gaussian blur.	99.351	69.38	68.13	77.17	53.38	10
Config6	100% Cutout.	98.164	79.28	79.65	70.01	61.62	10
Config7	100% Cutmix.	78.01	74.6	73.88	49.47	47.48	10
Config8	100% Mixup.	98.74	75.42	74.65	76.92	62.04	10
Config9	100% RandAugment.	99.40	80.98	80.14	75.92	60.34	15.51
Config10	100% TrivialAugment.	98.84	80.12	78.65	70.1	61.4	13.84
Config11	20% horizontal and vertical flips, 20% horizontal and vertical translation, 20% rotations, 20% contrast and brightness adjustment, and 20% gaussian blur.	89.68	78.84	77.6	58.38	55.16	10
Config12	20% Cutout and 80% Cutmix.	82.24	77.12	76.42	52.39	50.5	10
Config13	50% Cutout and 50% Cutmix.	86.28	77.88	77.52	58.42	55.68	10
Config14	80% Cutout and 20% Cutmix.	93.30	80.8	80.52	63.6	58.74	10
Config15	20% Cutout and 80% Mixup.	98.2	75.26	75.82	72.24	62.66	10
Config16	50% Cutout and 50% Mixup.	98.13	77.28	76.85	68.99	61.62	10
Config17	80% Cutout and 20% Mixup.	97.63	79.14	79.23	66.63	60.14	10
Config18	20% Cutmix and 80% Mixup.	95.87	77.06	76.84	75.62	58.42	58.59
Config19	50% Cutmix and 50% Mixup.	91.37	77.66	77.4	57.72	54.6	10
Config20	80% Cutmix and 20% Mixup.	82.43	76.66	76.14	52.56	50.46	10
Config21	20% Cutout, 20% Cutmix, and 60% Mixup.	93.19	80.52	79.28	67.2	61.18	10
Config22	20% Cutout, 40% Cutmix, and 40% Mixup.	90.97	78.48	79.08	60.22	56.3	10
Config23	20% Cutout, 60% Cutmix, and 20% Mixup.	86.48	79.06	77.6	55.53	53.64	10
Config24	40% Cutout, 20% Cutmix, and 40% Mixup.	94.18	79.42	79.69	63.7	59.38	10
Config25	40% Cutout, 40% Cutmix, and 20% Mixup.	89.26	80.32	79.24	56.74	52.96	51.9
Config26	60% Cutout, 20% Cutmix, and 20% Mixup.	93.65	80.72	79.68	62.96	58.08	10

Table 10: Comparision of augmentation configurations results for ALL-CNN and Resnet

ID	Augmentation Description	All CNN Supervised Acc			Resnet Supervised Acc		
		Train	Valid	Test	Train	Valid	Test
Config0	Baseline, without augmentation.	89.49	75.28	73.38	100	86.56	87.39
Config1	100% horizontal and vertical flips.	70.81	68.98	67.69	100	89.2	88.95
Config2	100% horizontal and vertical translation.	83.40	77.86	77.6	100	90.84	91.58
Config3	100% rotations.	87.36	79.36	77.89	100	89.42	89.77
Config4	100% contrast and brightness. adjustment.	98.83	68.46	68.76	100	83.84	84.99
Config5	100% gaussian blur.	99.351	69.38	68.13	100	86.2	85.47
Config6	100% Cutout.	98.164	79.28	79.65	100	88.84	88.93
Config7	100% Cutmix.	78.01	74.6	73.88	99.99	88.72	89.34
Config8	100% Mixup.	98.74	75.42	74.65	100	88.8	89.18
Config9	100% RandAugment.	99.40	80.98	80.14	73	65.78	65.83
Config10	100% TrivialAugment.	98.84	80.12	78.65	41.37	38.9	37.6
Config11	20% horizontal and vertical flips, 20% horizontal and vertical translation, 20% rotations, 20% contrast and brightness adjustment, and 20% gaussian blur.	89.68	78.84	77.6	100	90.22	90.71
Config12	20% Cutout and 80% Cutmix.	82.24	77.12	76.42	99.99	89.02	89.44
Config13	50% Cutout and 50% Cutmix.	86.28	77.88	77.52	100	89.28	90.16
Config14	80% Cutout and 20% Cutmix.	93.30	80.8	80.52	87.58	81.08	89.75
Config15	20% Cutout and 80% Mixup.	98.2	75.26	75.82	100	88.44	88.76
Config16	50% Cutout and 50% Mixup.	98.13	77.28	76.85	100	88.54	88.55
Config17	80% Cutout and 20% Mixup.	97.63	79.14	79.23	100	88.44	89.07
Config18	20% Cutmix and 80% Mixup.	95.87	77.06	76.84	100	89.52	89.72
Config19	50% Cutmix and 50% Mixup.	91.37	77.66	77.4	100	89.28	89.74
Config20	80% Cutmix and 20% Mixup.	82.43	76.66	76.14	100	89	90.25
Config21	20% Cutout, 20% Cutmix, and 60% Mixup.	93.19	80.52	79.28	100	89.06	89.69
Config22	20% Cutout, 40% Cutmix, and 40% Mixup.	90.97	78.48	79.08	100	89.34	89.65
Config23	20% Cutout, 60% Cutmix, and 20% Mixup.	86.48	79.06	77.6	100	89.84	90.17
Config24	40% Cutout, 20% Cutmix, and 40% Mixup.	94.18	79.42	79.69	100	89.12	89.61
Config25	40% Cutout, 40% Cutmix, and 20% Mixup.	89.26	80.32	79.24	100	89.46	89.67
Config26	60% Cutout, 20% Cutmix, and 20% Mixup.	93.65	80.72	79.68	100	89.6	89.29

Table 11: Comparision of augmentation configurations results for ALL-CNN and Resnet

ID	Augmentation Description	Resnet Semi Supervised Acc			ALL CNN Semi Supervised Acc		
		Train	Valid	Test	Train	Valid	Test
Config0	Baseline, without augmentation.	100	67.46	68.5	88.54	61.94	10
Config1	100% horizontal and vertical flips.	100	71.62	71.52	58.71	55.52	10
Config2	100% horizontal and vertical translation.	100	79.04	80.49	60.66	55.7	10
Config3	100% rotations.	99.7	75.54	75.81	55.48	53.16	10
Config4	100% contrast and brightness. adjustment.	100	63.44	63.98	72.43	56.44	10
Config5	100% gaussian blur.	100	64.34	64.97	77.17	53.38	10
Config6	100% Cutout.	100	71.58	72.36	70.01	61.62	10
Config7	100% Cutmix.	99.49	71.78	71.87	49.47	47.48	10
Config8	100% Mixup.	100	69.9	70.47	76.92	62.04	10
Config9	100% RandAugment.	68.53	52.34	52.64	75.92	60.34	15.51
Config10	100% TrivialAugment.	59.99	36.02	37.13	70.1	61.4	13.84
Config11	20% horizontal and vertical flips, 20% horizontal and vertical translation, 20% rotations, 20% contrast and brightness adjustment, and 20% gaussian blur.	99.94	77.86	78.19	58.38	55.16	10
Config12	20% Cutout and 80% Cutmix.	99.6	72.66	73.4	52.39	50.5	10
Config13	50% Cutout and 50% Cutmix.	99.91	71.54	72.08	58.42	55.68	10
Config14	80% Cutout and 20% Cutmix.	100	70.84	71.14	63.6	58.74	10
Config15	20% Cutout and 80% Mixup.	100	69.94	71.3	72.24	62.66	10
Config16	50% Cutout and 50% Mixup.	100	69.44	70.27	68.99	61.62	10
Config17	80% Cutout and 20% Mixup.	100	69.1	70.88	66.63	60.14	10
Config18	20% Cutmix and 80% Mixup.	100	70.92	71.12	75.62	58.42	58.59
Config19	50% Cutmix and 50% Mixup.	99.97	71.06	71.93	57.72	54.6	10
Config20	80% Cutmix and 20% Mixup.	99.86	71.64	72.31	52.56	50.46	10
Config21	20% Cutout, 20% Cutmix, and 60% Mixup.	99.99	71.32	72.28	67.2	61.18	10
Config22	20% Cutout, 40% Cutmix, and 40% Mixup.	100	72.46	72.47	60.22	56.3	10
Config23	20% Cutout, 60% Cutmix, and 20% Mixup.	99.9	71.96	73.07	55.53	53.64	10
Config24	40% Cutout, 20% Cutmix, and 40% Mixup.	100	70.38	70.46	63.7	59.38	10
Config25	40% Cutout, 40% Cutmix, and 20% Mixup.	99.97	71.52	79.24	71.98	52.96	51.9
Config26	60% Cutout, 20% Cutmix, and 20% Mixup.	100	70.86	71.54	62.96	58.08	10