Targeting the Distribution Gap using Augmentation

Sarah Danzi, Jennifer Mahle, John Boudreaux

Abstract

In 2019, researchers from UC Berkeley created new test datasets to assess how well current CIFAR and ImageNet classification models could generalize. Expecting to find diminishing accuracy drops on the new test sets due to adaptive overfitting, they instead observed an accuracy drop that fit almost linearly between the original test sets and the new test sets; an accuracy drop they attribute to a distribution shift in sampling during the test set creation process. In our work, we explore using data augmentation as a regularization mechanism to improve model generalization and lessen this distribution gap. Using the RandAugment and CutMix augmentation algorithms, we create five new datasets based on the CIFAR-10 training dataset, varying the intensity of the augmentation, and use these to train a subset of the models that the original research benchmarked. We then train these models at a reduced learning rate for 50 epochs with the original CIFAR-10 training dataset. The accuracy of each model is then assessed against both the CIFAR-10 and CIFAR-10.1 test sets. Running over 80 trials and expanding our experimental setup to test the impact of inserting augmentation at each phase of the pipeline (training, validation, test), our research shows a persistent distribution gap across experimental permutations. These results lead us to believe that augmentation is not a viable solution for minimizing the distribution gap and, in fact, often worsens overall accuracy.

1. Introduction

Image recognition and classification research and model training is often based on the performance of a few different academic standard datasets like CIFAR-10 or ImageNet. While this approach gives the advantage of being able to compare models to each other for performance metrics, training consistently on the same dataset introduces the question: "How much can we trust these classifiers in the real world?" Ultimately, the goal of image recognition and classification research is to produce models that are applicable in the real world. These models should perform well in real world applications, where misclassification may have real world consequences. As such, demonstrating that models are able to generalize to data similar to, but not exactly the same, as the training data becomes an important area of study.

There is reason to believe that models trained on CIFAR-10 might not generalize to the data used to generate their training and evaluation datasets. Recent work from UC Berkeley (Recht et. al) generated a new dataset using identical protocol to that of CIFAR-10 and observed that classifiers performed notably worse on the newly created CIFAR-10.1 test dataset. There are several reasons that a model may not generalize; however, they find that this difference is attributed to a distribution gap. A distribution gap occurs when there is a systemic difference between the datasets' distributions, causing gaps in model accuracy. Despite the researcher's efforts to replicate the original CIFAR-10 distribution, the distribution gap causes noticeable differences in accuracy ranging from 3% to 15%.

Additional UC Berkeley research created CIFAR-10.2, replicating the CIFAR-10 methodology to create a new train and test dataset, and it too was found to have gaps in accuracy attributed to the distribution gap (Lu et al, 2020). A question of interest is how to train models to lessen this distribution gap and hopefully have better performance in real world applications. Reducing the distribution gap would help models to generalize, maintaining similar accuracy when applied to data that is different from the training data. This is especially important as computer vision becomes an increasingly pervasive technology applied to ever changing real world data, such as autonomous vehicles or real-time facial recognition.

Research has repeatedly shown data augmentation to be a feasible regularization approach, which can help models to generalize. The goal of doing this is to help models only focus on the most important hallmark features of each class rather than idiosyncrasies of the distribution of data available for training. Additionally, augmented data provides a way to increase training data diversity, but without the need for costly data collection and labeling. A paper by He et al. (2019) showed that data augmentation in the initial training phases with reduction in augmentation in the final training phase improved results for overall model accuracy; however, there was no attention to the distribution gap. This work leverages that of He et al., and furthers it to analyze this approach and its applicability in lessening the distribution gap. Specifically, we created static augmented datasets and used them to train a selection of previously published models to assess the distribution gap.

2. Methods and Data

To explore the impact of data augmentation on the distribution gap, we select a dataset, augmentation methods, and models on which to conduct our experiments. Details on each with rationale for our selections follow.

Dataset Selection

Due to the derivative nature of our research, we had the opportunity to choose between focusing our experiments on either of the datasets for which Recht, et al. (2019) found the distribution gap, ImageNet or CIFAR. Because their research observed a roughly equivalent distribution gap for models on each, neither presented a technical advantage. We ultimately selected to use CIFAR due to the greater number of datasets available for it:

- The CIFAR-10 dataset includes a training and test set with 50,000 images and 10,000 images respectively (Krizhevsky et al., 2009).
- The CIFAR-10.1 dataset consists of a test set with 2,000 images (Recht et al., 2019)
- The CIFAR-10.2 dataset includes a training and test set with 10,000 images and 2,000 images respectively (Lu et al, 2020)

Each dataset is uniformly balanced between 10 classes and built from 32x32 images originating from the Tiny Images data source.

Augmentation Methods

We select two augmentation methods to focus our experiments on, RandAugment (Cubuk et al., 2019) and CutMix (Yun et al., 2019).

RandAugment is chosen for two key reasons. One, its published results show that it matches or outperforms all other automated augmentation algorithms on the CIFAR-10 dataset. It does so by applying a random series of transformations which we hypothesize may be valuable in increasing model robustness to distribution shifts in the datasets. Two, it allows the researcher to specify hyperparameters that vary the number and intensity of augmentation applied. This enables our experiments to more precisely assess if, and at what level, augmentation impacts the distribution gap.

RandAugment's set of possible augmentations include AutoContrast, Rotation, Solarize, Brightness, Sharpness, ShearX and ShearY, among others. The two tunable hyperparameters are:

- *N*, the number of random transformations to apply sequentially
- *M*, the magnitude of the individual transformations

RandAugment datasets have a shorthand notation of RA(N,M) throughout the paper.

We include CutMix as a second augmentation method in our experimental design to allow us to assess whether we observe similar results across augmentation methods. As with RandAugment, CutMix is chosen for both its reported performance and augmentation approach. At the time of its publication in 2019, CutMix outperformed all other augmentation methods on the CIFAR-10 dataset. Unlike RandAugment, however, its approach to achieving these results is quite different: rather than applying one or more transformations to a single image, it splices two images together. The algorithm samples images randomly from a dataset and cuts out a portion of one image to replace it with that same region of another image. The algorithm assigns each of the two images a specific weight aligned to the percentage of the image that is replaced. These weights also determine the proportion of each original label that is used as the resulting label for the newly transformed image. The resulting image has a combination of different labels, each which relate to a different proportion of the two original images. CutMix datasets have a shorthand notation of CM(alpha) throughout the paper.

Model Selection

We select a diverse subset of the image classification models for which Recht et al. (2019), demonstrated the distribution gap to use in our experiments:

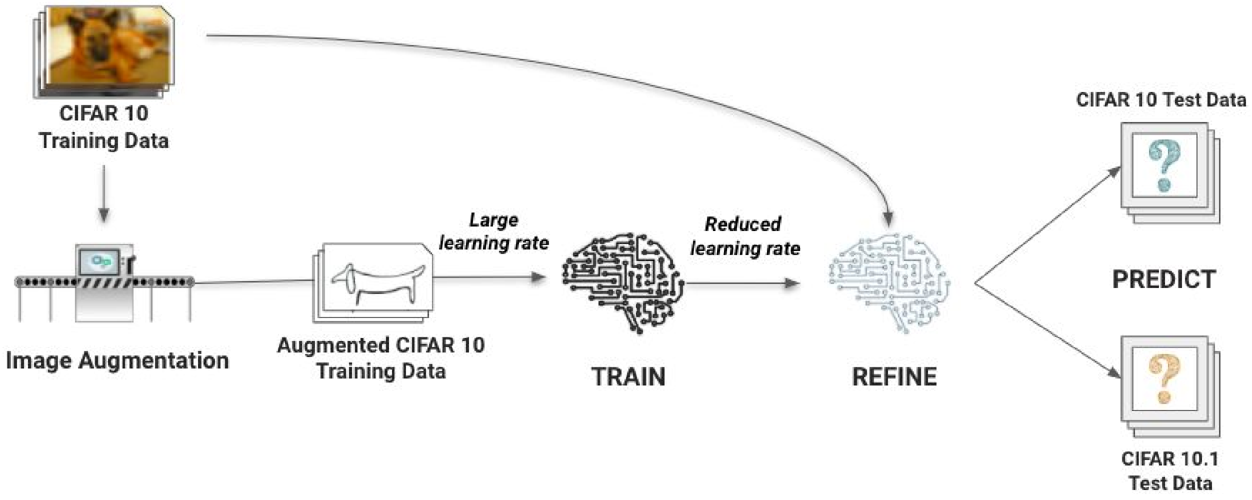
- resnet_basic_32 (Kaiming et al., 2015)
- wide_resnet_28_10 (Zagoruyko et al., 2016)
- resnext 29 4x64d (Saining et al., 2017)
- densenet_BC_100_12 (Huang et al., 2017)

Employing the same publicly available code that Recht et al. (2019) used[1], we confirm our ability to train each model on the CIFAR-10 training set and reproduce its respective baseline performance on both the CIFAR-10 and CIFAR-10.1 test sets prior to beginning any augmentation experiments. These results form the basis from which we will compare our experimental results for model accuracy, loss, and distribution gap.

3. Experimental Design

Our experimental design is based on research by He et al. (2019) who found that they were able to consistently use augmentation to improve model performance and generalization by dividing model training into two distinct phases. First, they would train a model on data that had undergone intensive data augmentation with elevated learning rates. They would then initiate a second training phase, or a refine phase, where they lower the learning rates and train the model with the original, unaugmented dataset to fine-tune model weights.

Figure 1 provides a graphic depiction of this approach as it applies to our research. Each step is described in detail below.



[1] https://github.com/hysts/pytorch_image_classification/

Figure 1. *Experimental Design*

Image Augmentation

The *Image Augmentation* step takes the CIFAR-10 training dataset as an input and applies one of our selected augmentation algorithms to generate a new training dataset. We deliberately select to perform augmentation outside of the model training phase to enable the creation of these datasets. In order to compare results across experiment trials, we felt that it was important that the same augmented training dataset be supplied to all models to ensure apples-to-apples comparisons. We generate a total of five augmented datasets using this approach.

RandAugment

Four of the augmented training datasets are generated using RandAugment. Hyperparameter settings for each are shown in Table 1.

Experiment Setup	***N*** **Number of Transformations**	***M*** **Magnitude of Transformations**
Original	*0*	*0*
A	1	20
B	2	5
C	2	20
D	3	20

Table 1. *Table of RandAugment dataset configurations*

Juxtaposed with the original images, samples from each of the four training datasets we generated with RandAugment are shown in Figure 2. The images become less and less discernible --- at least to the human eye --- as we increase the augmentation. This series of datasets gives us the opportunity to attempt to understand if and how different levels of augmentation improve generalization and shrink the distribution gap.

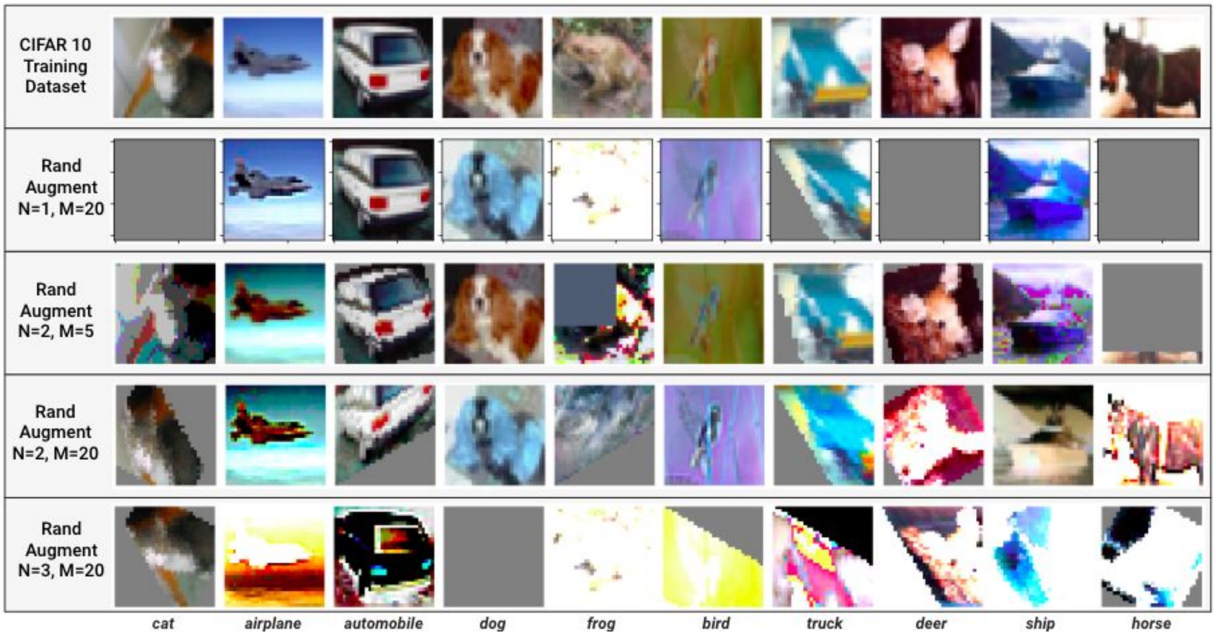


Figure 2. *An example for each CIFAR class in our RandAugment training datasets is shown. Images are significantly changed with higher N and M values*

CutMix

CutMix is used to generate one of the five augmented training datasets. As shown in Figure 3, each image in this dataset is a combination of two images from the original CIFAR-10 training dataset. Two labels exist for each image.

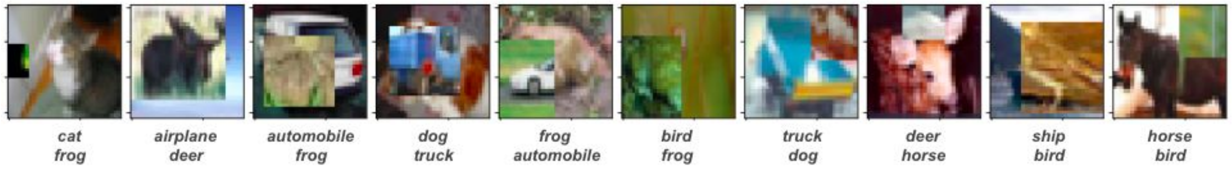


Figure 3. *An example for each CIFAR class in our CutMix-augmented training dataset is shown. CutMix combines two images from different classes into a single image*

Train

During the Train phase in Figure 1, each augmented dataset is used to train each of the four models we selected for our research. For our baseline experiments, we train each respective model for 400 epochs with an initial learning rate of 0.1, following the research approach of He et al. (2019). At the end of this step, we benchmark our accuracy, loss, and distribution gap for the CIFAR-10 and CIFAR-10.1 test datasets to assess performance at this phase.

Refine

For the Refine phase in Figure 1, we resume training the model with a reduced learning rate, using the unaugmented CIFAR-10 training dataset, for 50 epochs. The new, reduced learning rate is determined by the learning rate of the last epoch of the previous training cycle. For our models, this ranged from between 0.0008 to 0.001.

Predict

During the Predict phase in Figure 1, we evaluate model performance against both the CIFAR-10 and CIFAR-10.1 test datasets. We record our accuracy, loss, and distribution gap and compare it against the original, observed scores.

Supplementary Experiments

To fully assess the impact and usefulness of augmentation on closing the distribution gap, we conducted two additional experiments to supplement those described above. While our primary experimental design focused on understanding the impact of data augmentation during model training, we expanded our trials to also measure and characterize the impact of augmentation on validation data and testset data.

Additional details on these experiments and the results can be found in the appendix.

4. Results

Results for the experimentation with both RandAugment and CutMix suggest that augmentation is not an effective nor consistent way to lessen the distribution gap.

RandAugment Results

Figure 4 shows a graphical summary of all models trained on training sets augmented with RandAugment, with separate colors for each hyperparameter configuration as specified in Table 1. Closing the distribution gap would be indicated by these lines having a near-identical slope and small distance to the dashed reference line, which marks where accuracy would lie for performing equal on CIFAR-10.1 and CIFAR-10. As we can see, none of these augmentation experiments successfully achieve this task.

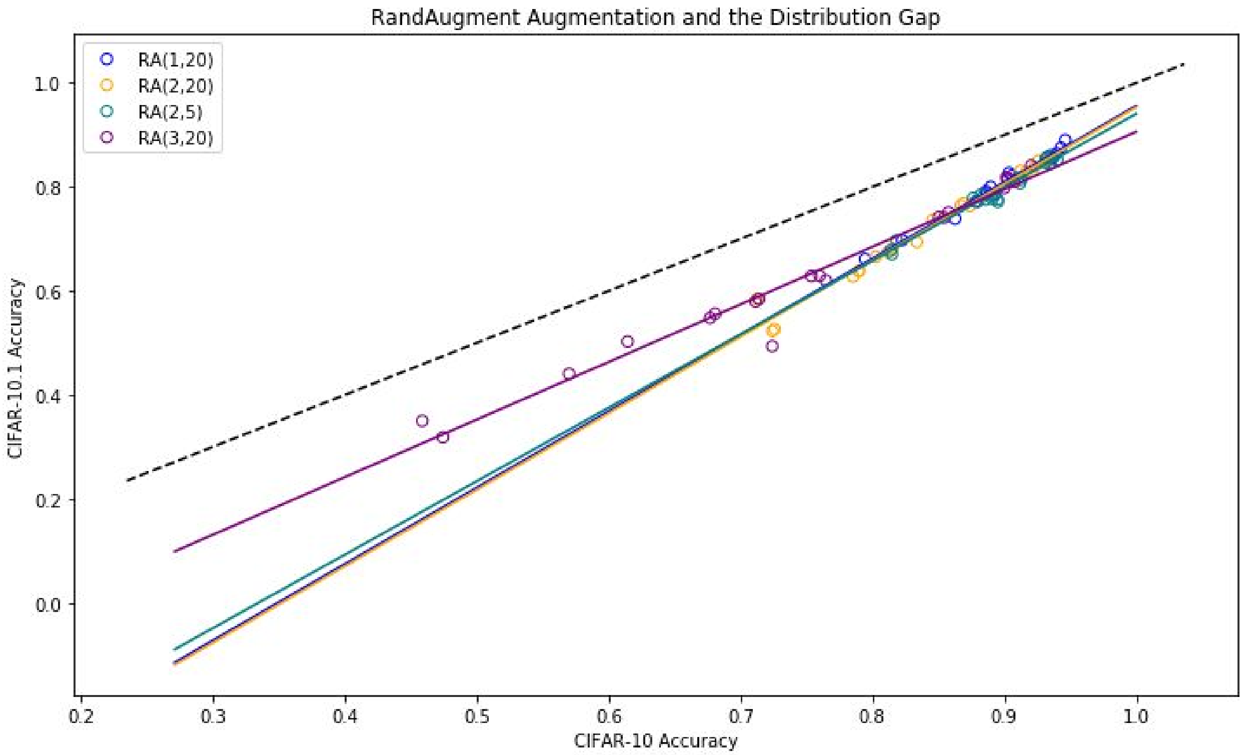


Figure 4. *Summary of RandAugment experiment results. No augmented training models appear to come close to equal performance on CIFAR-10 and CIFAR-10.1 test sets.*

Interestingly, adding the third transformation in the N=3, M=20 training set appears to have a gentler slope than the others. This slope value means comparatively, increases in performance for CIFAR-10 likely mean less generalizable results to other test sets.

The effects of each RandAugment training set were seen to various extents across the major models tested, as seen in Table 2 which reports the highest accuracy for each model.

RandAugment(N,M)		**Densenet**	**Resnet-32**	**Resnext-29**	**Wide Residual Net**
CIFAR-10	**Baseline**	**94.56%**	**92.32%**	**95.35%**	**95.78%**
	RA(1,20)	93.49%	88.97%	93.98%	94.63%
	RA(2,5)	93.13%	85.46%	93.54%	94.03%
	RA(2,20)	91.68%	86.70%	92.60%	93.15%
	RA(3,20)	90.76%	87.66%	90.18%	90.01%
CIFAR-10.1	**Baseline**	**88.30%**	**83.20%**	**89.05%**	**89.75%**
	RA(1,20)	86.05%	80.05%	86.45%	89.05%
	RA(2,5)	84.70%	77.90%	84.55%	85.20%
	RA(2,20)	82.90%	76.45%	85.05%	84.70%
	RA(3,20)	80.95%	74.10%	81.85%	79.80%
Distribution Gap	**Baseline**	**6.26%**	**9.12%**	**6.30%**	**6.03%**
	RA(1,20)	7.44%	8.92%	7.53%	5.58%
	RA(2,5)	8.43%	7.56%	8.99%	8.83%
	RA(2,20)	8.78%	10.25%	7.55%	8.45%
	RA(3,20)	9.81%	13.56%	8.33%	10.21%

Table 2. *Summary of RandAugment experiment results, by model type*

No models trained with RandAugment data outperformed their baseline counterparts for model accuracy on both CIFAR-10 or CIFAR-10.1. The distribution gap did improve by 0.45% for the wide residual net model for the RA(1,20) training set, but this trend was not consistent with the rest of the training sets and therefore we believe this is not a significant finding. Overall, more augmentation not only hurts model accuracy but also the distribution gap.

CutMix Results

CutMix augmentation was performed with fewer experiments than RandAugment and with a single hyperparameter value, alpha = 1. Figure 5 shows a graphical representation of these results compared to the RandAugment results.

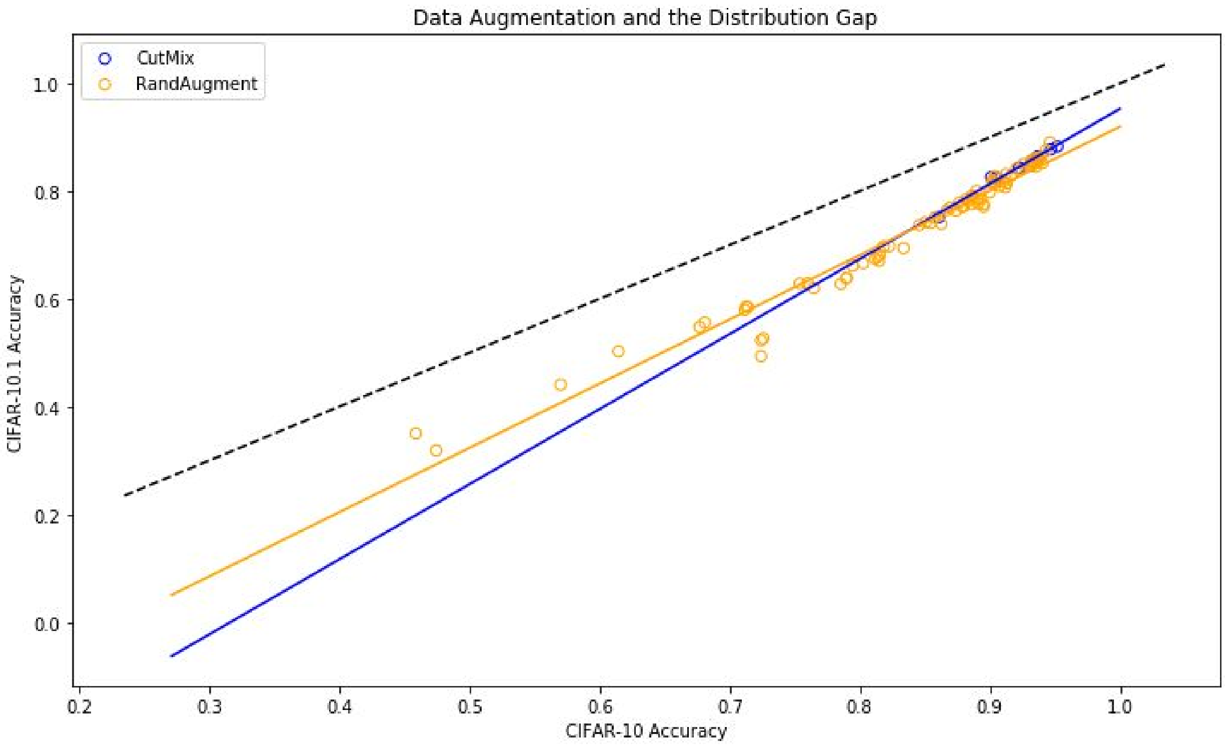


Figure 5. *CutMix experiment results display similar trends to RandAugment.*

CutMix, similar to RandAugment, does not appear to aid the distribution gap in a meaningful way. When comparing all experiments across RandAugment and CutMix, it appears neither truly helps a model achieve near-equal performance on CIFAR-10 and CIFAR-10.1. CutMix appears to approach true generalization marginally better, as indicated by the steeper slope of its fit line. While this is a result trending in the direction of lessening the distribution gap, it is important to note that this is typically associated with lower model performance compared to baseline so this is of limited utility. There is one exception to this rule with the CIFAR-10.1 test results and distribution gap for wide residual net models. The minor improvements are well within the range of the generalization gap for statistical error (Recht et. al, 2019) and are not considered significant.This can be more effectively seen in Table 3, which reports best performance by model. Given the initial experiment results, we did not pursue further experimentation with this augmentation procedure which is the reason for the reason for less data displayed.

CutMix(alpha)		Densenet	Resnet-32	Resnext-29	Wide Residual Net
CIFAR-10	**Baseline**	**94.56%**	**92.32%**	**95.35%**	**95.78%**
	CM(1)	93.69%	90.10%	94.68%	95.20%
CIFAR-10.1	**Baseline**	**88.30%**	**83.20%**	**89.05%**	**89.75%**
	CM(1)	86.45%	82.60%	87.75%	90.37%
Distribution Gap	**Baseline**	**6.26%**	**9.12%**	**6.30%**	**6.03%**
	CM(1)	7.24%	7.50%	6.93%	4.83%

Table 3. *Summary of CutMix experiment results by model type*

All experiment results can be found in the Github repository.[2]

5. Further Analysis

There were many further analyses done regarding the distribution gap, our experimental procedure, and results to better understand effects of augmentation on model predictions and misclassifications. For the sake of brevity, these are not included in the paper but can be found in the Appendix.

For analysis on comparing the test sets of CIFAR-10 and CIFAR-10.1, including bootstrap sampling to confirm the presence of the distribution gap and SSIM analysis between test sets, see Appendix A.

For further analysis of experimental results, see Appendix B. Analyses include experiments with alternative epochs for training and refinement, and in-depth looks into consistency of models misclassifying particular images.

For analysis of the effects of augmentation on alternative stages of the pipeline, see Appendix C. This appendix has the results of experiments changing validation data and test set data using data augmentation, which provided a different line of experimentation with ultimately the same conclusion as the main results previously reported.

6. Conclusion and Future Work

[2] https://github.com/danzisar/w210-capstone/blob/master/analysis/Results_tables.ipynb

Overall, it seems that data augmentation does not effectively bridge the distribution gap seen in computer vision research. Although we tested several different augmented datasets using four different previously published models, the distribution gap between CIFAR-10 and CIFAR-10.1 persists. Looking into different combinations of augmented training, validation, and test sets did not yield particularly promising results.

While data augmentation does not appear to directly help the distribution gap, we do think that data augmentation could provide other useful benefits in machine learning model generalization. We propose the following lines of experimentation and analysis to continue on our work and further the investigation into how augmentation might help computer vision.

Alternate Data Sources

While augmenting a particular dataset has not proven to be effective in the case of CIFAR-10, it is plausible that additional images coming from different sources could increase information entropy in the training set. Making new training datasets with sources like ShapeNet (Chang et al. 2015), which has 3D models for many CIFAR-10 classes, could aid algorithms to be more robust to changes in angle and perspective.

Active Learning

To take an alternative perspective from the experimentation in this work, one could use active learning to "learn" which images from a diverse pool would make the best new training set to reduce the distribution gap. When evaluating which new images to select, image augmentation techniques could be used to provide alternative metrics for consideration. These new metrics may result in a more robust distribution of images in the training set, making it a better candidate to reduce the distribution gap in future test sets.

Test Set Augmentation

Test set augmentation showed some favorable results (seen in Appendix C) for reducing the distribution gap but overall made accuracy suffer significantly. We hypothesize that different methods of augmentation may give better results, as some augmentations may block out too much of an image to be discernible. Other more mild augmentations may give better results for absolute accuracy.

Model Interpretability

While this study did a few initial analyses into exactly how models change as a result of augmented training data, we feel that there could be further insight in different directions of analyses. In particular, using techniques such as DeepExplainer (Lundberg and Lee, 2017) allows deeper analysis into which parts of the image are useful for classification. Performing further analysis on the visual results from such a technique ought to give further insight into how augmentation changed feature importance for particular parts of images. This line of questioning could be further bolstered by an ablation study.

References

Cubuk, E., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. (2019). AutoAugment: Learning Augmentation Strategies From Data. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Chang, Angel X. and Funkhouser, Thomas and Guibas, Leonidas and Hanrahan, Pat and Huang, Qixing and Li, Zimo and Savarese, Silvio and Savva, Manolis and Song, Shuran and Su, Hao and Xiao, Jianxiong and Yi, Li and Yu, Fisher (2015). ShapeNet: An Information-Rich 3D Model Repository. arXiv:1512.03012 [cs.GR]

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, & Quoc V. Le. (2019). RandAugment: Practical automated data augmentation with a reduced search space.

Gao Huang and Zhuang Liu and Kilian Q. Weinberger (2016). Densely Connected Convolutional Networks*CoRR, abs/1608.06993*.

He, Zhuoxun & Xie, Lingxi & Chen, Xin & Zhang, Ya & Wang, Yanfeng & Tian, Qi. (2019). Data Augmentation Revisited: Rethinking the Distribution Gap between Clean and Augmented Data.

Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun (2015). Deep Residual Learning for Image Recognition*CoRR, abs/1512.03385*.

Krizhevsky, A. Learning Multiple Layers of Features From Tiny Images, 2009. https://www.cs.toronto.e du/~kriz/learning-features-2009-TR.pdf.

Lu, Shangyun, Nott, Bradley Nott, Olson, Aaron, Todeschini, Alberto, Vahabi, Hossein, Carmon, Yair, & Schmidt, Ludwig. (2020) Harder or Different? A Closer Look at Distribution Shift in Dataset Reproduction

Lundberg, Scott M and Lee, Su-In. (2017) A Unified Approach to Interpreting Model Predictions. In Advances in Neural Information Processing Systems 30.

Recht, Benjamin, Roelofs, Rebecca, Schmidt, Ludwig, & Shankar, Vaishaal (2019). Do ImageNet Classifiers Generalize to ImageNet?CoRR, abs/1902.10811.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, & Wieland Brendel. (2018). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.

Saining Xie and Ross B. Girshick and Piotr Dollár and Zhuowen Tu and Kaiming He (2016). Aggregated Residual Transformations for Deep Neural Networks*CoRR, abs/1611.05431*.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In International Conference on Computer Vision, 2019.

Sergey Zagoruyko and Nikos Komodakis (2016). Wide Residual Networks*CoRR, abs/1605.07146*.

Appendix

The appendix contains a detailed examination of the various trials and experiments we conducted during our research. A table of contents is provided as an organizing construct for the reader.

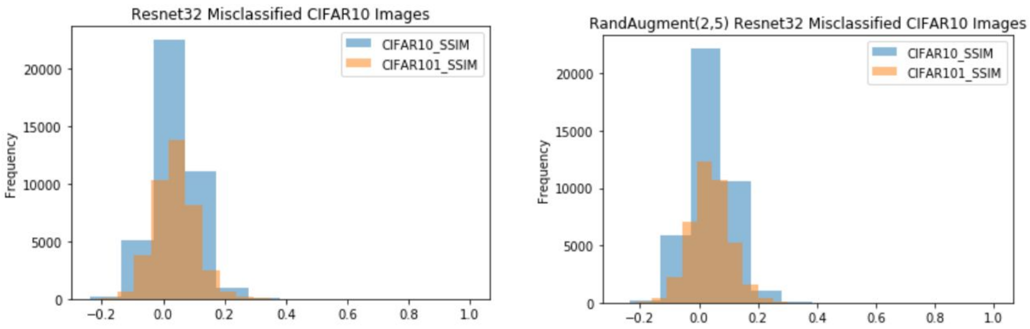
Appendix Table of Contents

A: Test Set Analysis

SSIM Analysis

To analyze misclassified images, we looked into whether there are disparities in image similarities when comparing misclassified testset images for CIFAR-10 and CIFAR-10.1. To conduct this analysis, we use the Structural Similarity Index Measure (SSIM), which measures the similarity between two images. Testset images with the greatest ratio of predicted to correct probability (r ratio) for each model were selected as the misclassified images for the analysis.

We then calculated SSIM comparison between misclassified images and CIFAR-10 testset and between CIFAR-10.1 testset. Figure 6 below shows the histograms of SSIM comparisons for Resnet32. We also calculated the mean and median SSIM to assess whether there are differences in similarities in aggregate. For misclassified testset images, CIFAR-10.1 has greater differences in SSIM across models and training sets. Results for other models and augmented training datasets produce similar results to those shown below, as did breaking the analysis out by the different classificationsAdditional results can be found on git[3].



[3] https://github.com/danzisar/w210-capstone/blob/master/analysis/Misclassified_Image-SSIM_MSE_Analysis.ipynb

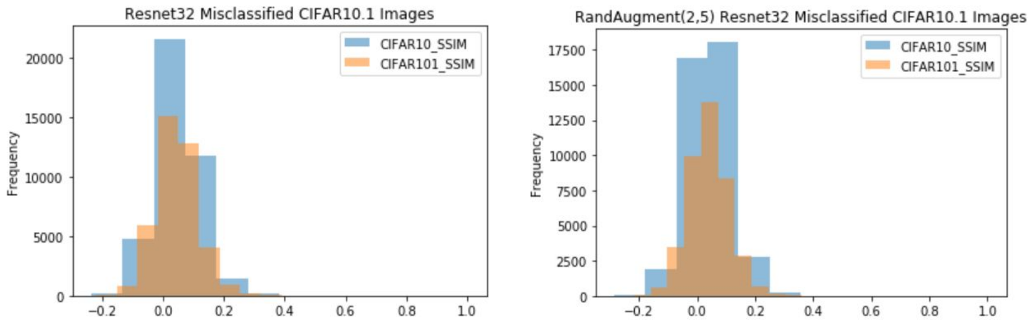


Figure 6. *Comparison of Misclassified Image Similarities via SSIM*

Additionally, we investigated whether there is a statistically significant difference between the CIFAR-10 and CIFAR-10.1 test sets[4]. To assess these differences, we calculated SSIM scores for each class by comparing every CIFAR-10 training set image in a class to every image in that same class in the test sets (CIFAR-10 and CIFAR-10.1). Once the SSIMs were calculated, we conducted t-tests between the set of SSIM values for each class (e.g., CIFAR-10 training automobiles vs. CIFAR-10 test automobiles and CIFAR-10 training automobiles vs. CIFAR-10.1 test automobiles). We found that the SSIM values were statistically different from each other for each of the 10 classes. Subsetting to images that all models predicted correctly also yielded statistically significant t-test results. Interestingly, when we subsetted to compare SSIM values between the images that all models predicted incorrectly, we found two classes, bird and ship, for which there was not a statistically significant difference. This suggests that, with regards to a distribution shift, these two classes may be the most similar between the CIFAR-10 and CIFAR-10.1 test sets. Table 4 shows the p-values, by class, for our t-tests that compare the set of SSIM values calculated by comparing CIFAR-10 training images to CIFAR-10 to the set of SSIM values calculated by comparing CIFAR-10 training images to CIFAR-10.1 test images.

[4] https://github.com/danzisar/w210-capstone/blob/master/analysis/TestsetEDA_SSIM.ipynb

Class	Comparing all Test Set Images	Using only Test Set Images that all RandAugment models predicted correctly	Using only Test Set Images that all RandAugment models predicted incorrectly
airplane	0.0	0.0	7.9234e-35
automobile	0.0	0.0	6.5861e-133
bird	0.0	0.0	0.3883
cat	2.4436e-98	1.6414e-120	8.5521e-29
deer	0.0	0.0	4.9795e-253
dog	2.9483e-70	6.0400e-117	6.6276e-106
frog	0.0	1.0514e-265	3.5814e-86
horse	1.5118e-233	4.5856e-210	2.5275e-25
ship	0.0	1.2191e-154	0.2153
truck	1.3751e-184	7.5700e-294	1.0447e-204

Table 4. *p-values for a t-test that compares the set of SSIM values calculated by comparing CIFAR-10 training images to CIFAR-10 to the set of SSIM values calculated by comparing CIFAR-10 training images to CIFAR-10.1 test images*

Figure 7 below shows the distribution of SSIM values for the CIFAR-10 and CIFAR-10.1 test sets respectively, broken out by category. Similarly, Figures 8 and 9 below shows the distribution of SSIM values, but for images that all models got correct and for images that all models got incorrect, respectively. For each of these figures, comparing the SSIM distribution for each of the categories shows that there are indeed differences between CIFAR-10 and CIFAR-10.1 test sets.

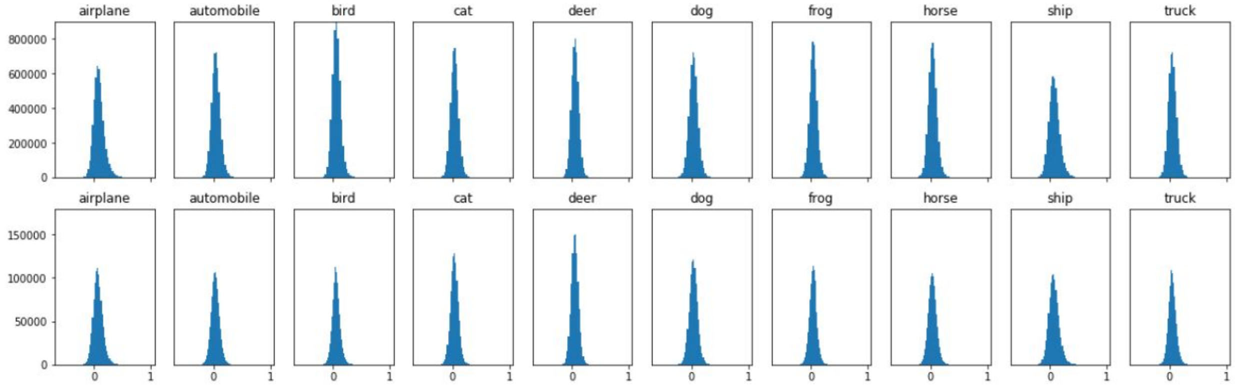


Figure 7. *Comparison of SSIM by Category for CIFAR-10 and CIFAR-10.1 Test Sets*

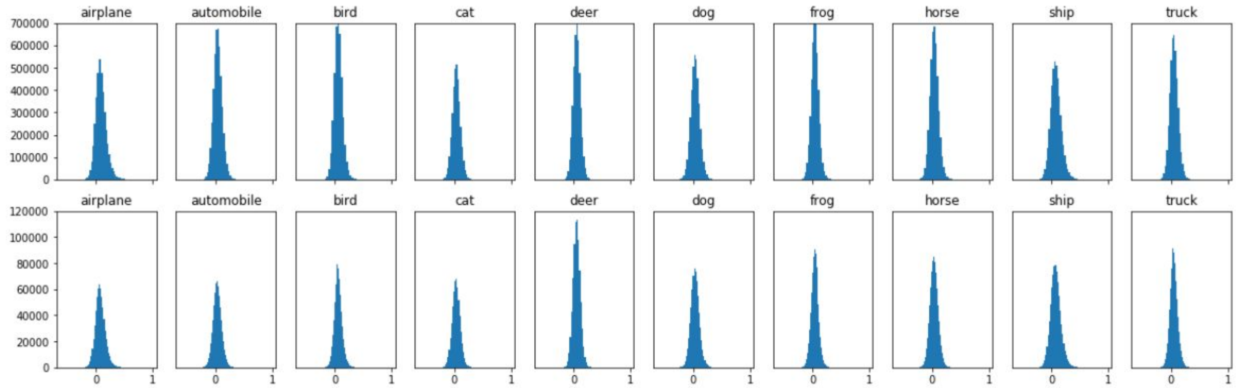


Figure 8. *Comparison of SSIM by Category for CIFAR-10 and CIFAR-10.1 Test Sets - Subsetting to Images all Models Correctly Classified*

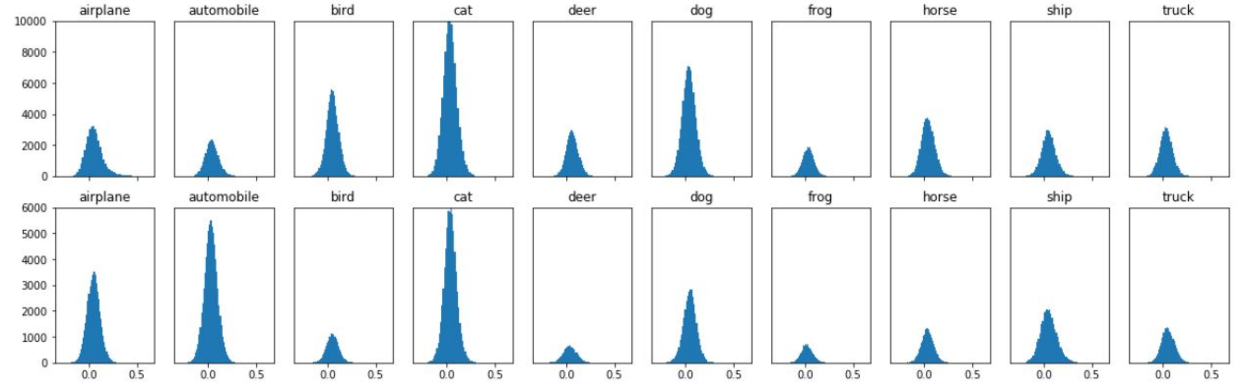


Figure 9. *Comparison of SSIM by Category for CIFAR-10 and CIFAR-10.1 Test Sets - Subsetting to Images all Models Incorrectly Classified*

Impact of Test Set Bootstrapping

Given that CIFAR-10 and CIFAR-10.1 are generated from the same corpus of data and follow the same protocol, it could be possible that the distribution gap between these two could be explained by random sampling error. We test this hypothesis by taking several bootstrapped samples of the CIFAR-10 test set (n=10,000) to the same size of the CIFAR-10.1 test set (n=2,000)[5]. If the performance gap observed between these two test sets was from sampling error, we would see that the performance on the bootstrapped samples should not be different with statistical significance. Performing this test with just the wide residual net model (wrn) was enough to provide evidence that this does not appear to be the case. Figure 10 shows the results from this analysis, with the red line marking the accuracy on CIFAR-10.1 test set, and the blue bars representing the distribution of accuracy on bootstrapped test sets. The accuracy on CIFAR-10.1 test set falls at 89.75%, whereas the accuracy on bootstrapped test sets range from 94-97% accuracy, substantially above that of the CFAR10.1 test set.

[5] https://github.com/danzisar/w210-capstone/blob/master/analysis/TestsetEDA_DistributionGap.ipynb

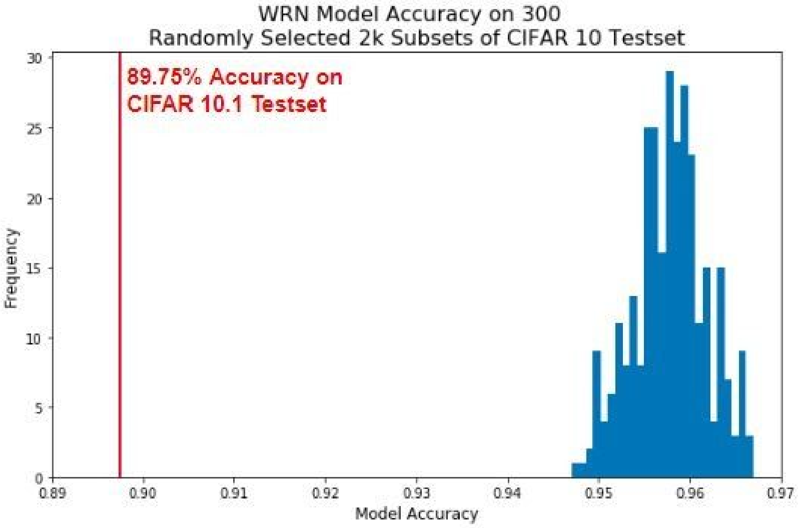


Figure 10. *Bootstrapped test sets from CIFAR-10 do not approach CIFAR-10.1, confirming the presence of a distribution gap as opposed to random sampling error*

B: Model Results Analysis

Alternative Allocations of Augmented Training and Refinement

While the main experiment results summarized in Figures 4 and 5 show that augmentation was not effective in eliminating the distribution gap, this was tested using 400 epochs of augmented training and 50 epochs of refinement for every experiment. We hypothesized different levels of augmentation and refinement, while keeping the same number of overall training epochs the same, could result in more favorable results for the distribution gap. Training on Resnet32 was split between a number of epochs with a high learning rate with RandAugment data (n=2, m=20) and training on the original CIFAR-10 data, such that the total number of epochs of training was equal to 450. The models did not show any notable performance increase over the original CIFAR-10 distribution gap ($p < 1e-18$). Figure 11 shows this result.

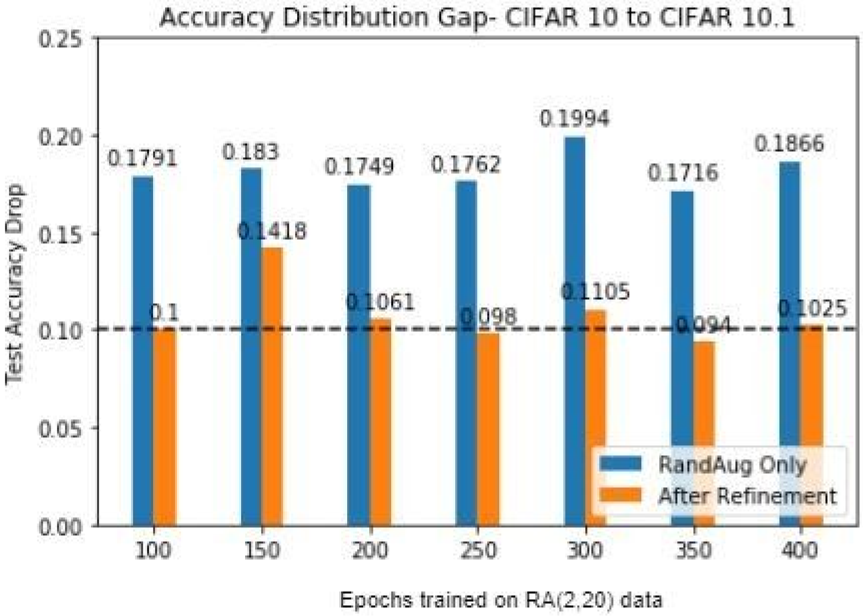


Figure 11. *Different levels of augmentation does not appear to significantly impact the distribution gap*

Further checks into model accuracy for each test set showed that the number of epochs allocated to training on augmented data did not make any model perform with higher accuracy for CIFAR-10 or CIFAR-10.1 test sets. These can be seen in Figure 12 and Figure 13.

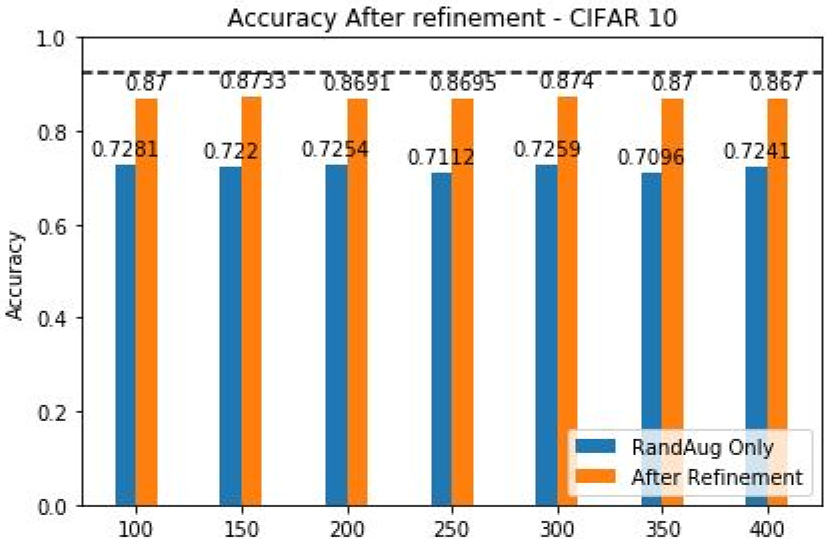


Figure 12. *Model accuracy grouped by number of epochs trained on augmented data for CIFAR-10. No configuration outperforms the baseline*

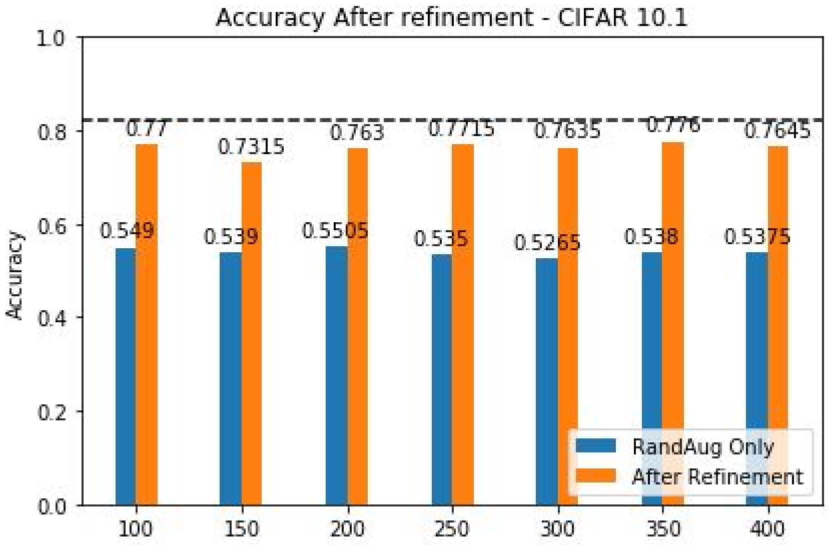


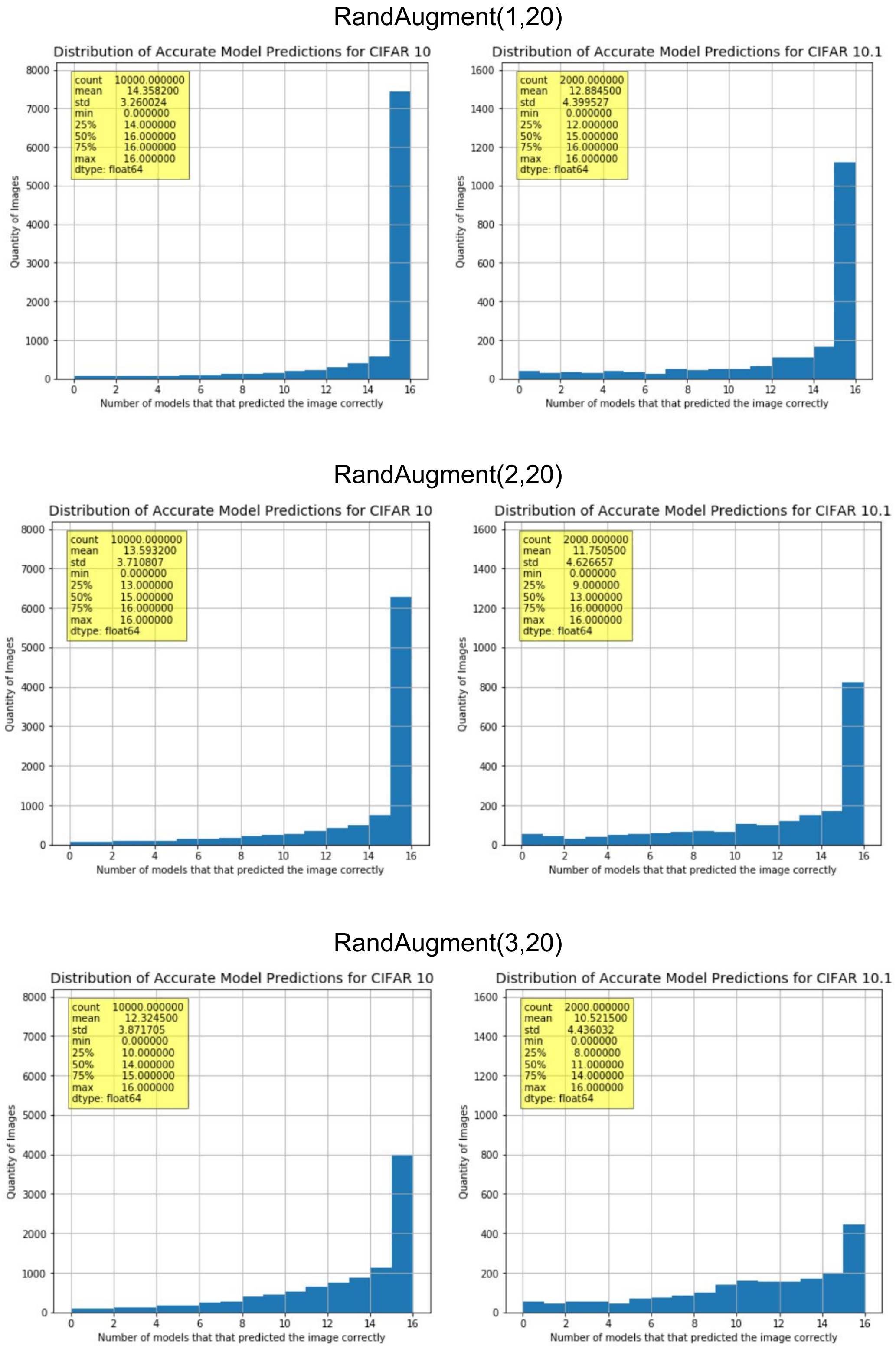
Figure 13. *Model accuracy grouped by number of epochs trained on augmented data for CIFAR-10.1. No configuration outperforms the baseline*

The code for this analysis can be found in the associated github repository.[6]

Aggregate misclassification across models

After training many different models with RandAugment protocol and not observing differences in the distribution gap, we analyzed how differently the models were assigning class probabilities as a result of the augmentation. On an aggregate level, this can be seen by looking at the number of models that misclassify each image trained on unaugmented data versus being trained on each of the augmented datasets we created. Figure 14 shows results of interest for this analysis, with each set of graphs showing the results for RandAugment datasets with increasing augmentation levels for M=20 and N ranging from 1 to 3. Additional graphs can be found in the borebook on git[7] Overall, increasing augmentation reduced the number of images every individual model misclassified, but it also reduced the number of images that every model is able to correctly classify.

[6] https://github.com/danzisar/w210-capstone/blob/master/analysis/Resnet_training_refining_cutoffs.ipynb
[7] https://github.com/danzisar/w210-capstone/blob/master/analysis/PredictionResultsAnalysis.ipynb



RandAugment(1,20)

Distribution of Accurate Model Predictions for CIFAR 10

```
count    10000.000000
mean        14.358200
std          3.260024
min          0.000000
25%         14.000000
50%         16.000000
75%         16.000000
max         16.000000
dtype: float64
```

Distribution of Accurate Model Predictions for CIFAR 10.1

```
count    2000.000000
mean       12.884500
std         4.399527
min         0.000000
25%        12.000000
50%        15.000000
75%        16.000000
max        16.000000
dtype: float64
```

RandAugment(2,20)

Distribution of Accurate Model Predictions for CIFAR 10

```
count    10000.000000
mean        13.593200
std          3.710807
min          0.000000
25%         13.000000
50%         15.000000
75%         16.000000
max         16.000000
dtype: float64
```

Distribution of Accurate Model Predictions for CIFAR 10.1

```
count    2000.000000
mean       11.750500
std         4.626657
min         0.000000
25%         9.000000
50%        13.000000
75%        16.000000
max        16.000000
dtype: float64
```

RandAugment(3,20)

Distribution of Accurate Model Predictions for CIFAR 10

```
count    10000.000000
mean        12.324500
std          3.871705
min          0.000000
25%         10.000000
50%         14.000000
75%         15.000000
max         16.000000
dtype: float64
```

Distribution of Accurate Model Predictions for CIFAR 10.1

```
count    2000.000000
mean       10.521500
std         4.436032
min         0.000000
25%         8.000000
50%        11.000000
75%        14.000000
max        16.000000
dtype: float64
```

Figure 14. *For RandAugment, increasing augmentation changes the number of models able to correctly classify images*

Test Set Misclassification Overlap

In addition to the aggregate-level look into how many models misclassify each image, we investigated each model's misclassified images. In particular, we wanted to see if there were in particular trends around augmentation helping or hurting certain images being classified correctly. In other words, if no augmentation or minor augmentation misclassified an image but the same image was correctly classified in more augmented models, this could give evidence that the models help generalize better to new data.

To analyze this phenomena, graphics for each model family (Resnet, Densenet, Resnext, Wide Residual Net) were created. The graphics use the 20 strongest misclassified (highest assigned probability) test set images for each model training across the RandAugment training. We then group each image and give a mark where the image has been misclassified in each column. Each row is a separate model, ordered with the least augmented models on top and the most augmented on the bottom. Example graphics for Resnet32 can be seen in Figure 15 for CIFAR-10 test set, and Figure 16 for CIFAR-10.1 test set.

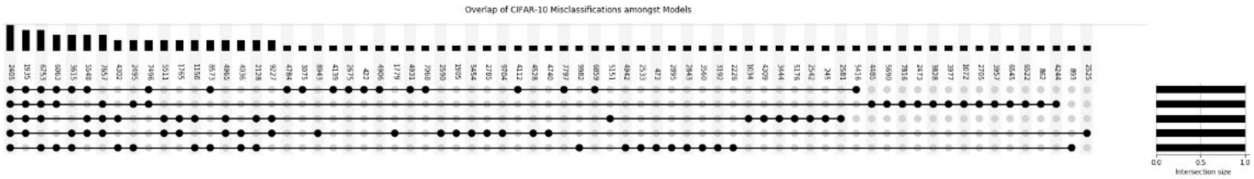


Figure 15. *CIFAR-10 misclassified images across different Resnet model trainings*

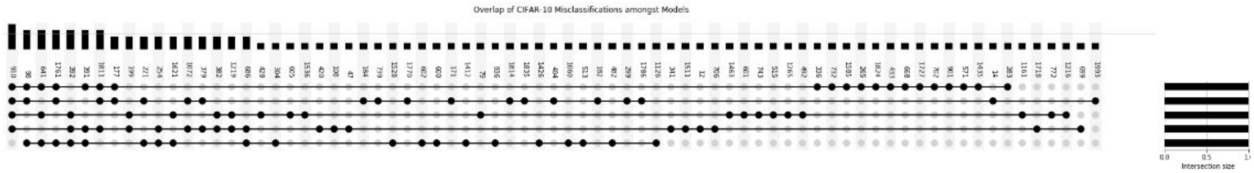


Figure 16. *CIFAR-10.1 misclassified images across different Resnet model trainings*

Overall, we do not see any sort of clear trend for Resnet or any other models. It is interesting that models appear to misclassify the same images more or less at random, indicating that the effects of augmentation are variable rather than showing uniform trends.

Graphics for all visualizations can be seen in the github repository.[8]

Maximum Assigned Model Probabilities

The same phenomena can be seen on an individual model level as well. When looking at assigned image probabilities, augmented models appear to be less certain in their assigned

[8] https://github.com/danzisar/w210-capstone/blob/master/analysis/Testset_misclassified_overlap.ipynb

probabilities compared to the non-augmented counterparts. This applies to both correctly and incorrectly assigned labels. Figure 17 displays selected results demonstrating this trend.

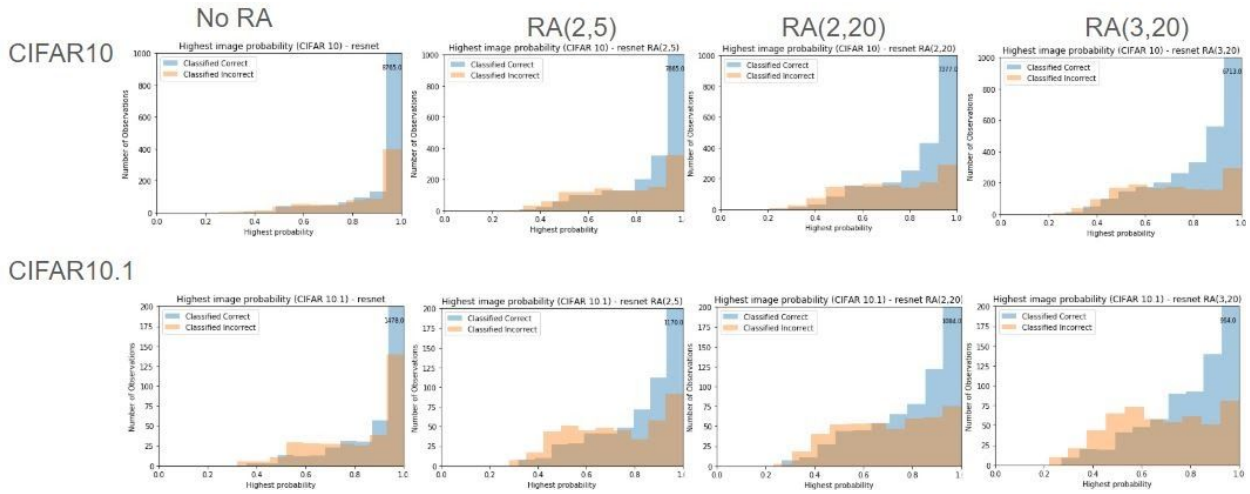


Figure 17. *Models become more uncertain as they are trained on more heavily augmented data*

It is unclear if these results demonstrate that these models are now better generalized to perform on new data or just simply perform differently. Lower accuracy scores compared to the original publications on both CIFAR-10 and CIFAR-10.1 suggest that this approach simply makes models worse overall.

C: Impact of Augmentation at Each Phase of the Pipeline

In pursuing augmentation as a means for closing the distribution gap, we pursued experiments that would allow us to understand the impact of applying augmentation at each phase of the pipeline: training, validation, and inference. The table below summarizes each of the combinations run as part of this research.

TRAINING		REFINE	INFERENCE
Training Data	Validation Data		
CIFAR-10	CIFAR-10	-	CIFAR-10
			CIFAR-10.1
RandAugment(CIFAR-10, N=x, M=y) $\forall\ (x,y) \in$ {(1,20),(2,20),(3,20),(2,5)}	RandAugment(CIFAR-10, N=x, M=y)	-	CIFAR-10
			CIFAR-10.1
			RandAugment(CIFAR-10, N=x, M=y)
			RandAugment(CIFAR-10.1, N=x, M=y)
		CIFAR-10	CIFAR-10
			CIFAR-10.1
			RandAugment(CIFAR-10, N=x, M=y)
			RandAugment(CIFAR-10.1, N=x, M=y)
	CIFAR-10	-	CIFAR-10
			CIFAR-10.1
			RandAugment(CIFAR-10, N=x, M=y)
			RandAugment(CIFAR-10.1, N=x, M=y)
		CIFAR-10	CIFAR-10
			CIFAR-10.1
			RandAugment(CIFAR-10, N=x, M=y)
			RandAugment(CIFAR-10.1, N=x, M=y)

Table 5. *Experiment Permutations*

Each of the following subsections discusses the results for each variation.

Validating Training Epochs with Augmented Data

Because validation results during training epochs are used to tune model parameters, we experimented with augmentation on the validation data set to determine if the distribution gap would be affected. We ran the experiment with all four models and each of our training/validation datasets generated using RandAugment.

The table below depicts the results for the densenet model on each RandAugment(N,M) dataset. Similar in nature to the results observed for the other three models, a consistent, meaningful pattern does not emerge.

N	M	Refined with Unaugmented Data	Testset	Accuracy: Augmented Validation	Accuracy: Unaugmented Validation	Delta in Accuracy	Unaugmented Higher Accuracy?
1	20	False	CIFAR-10	88.62	88.66	0.04	True
			CIFAR-10.1	79.15	78.75	0.40	False
		True	CIFAR-10	93.45	93.49	0.04	True
			CIFAR-10.1	85.30	86.05	0.75	True
2	5	False	CIFAR-10	89.52	89.07	0.45	False
			CIFAR-10.1	77.10	77.70	0.60	True
		True	CIFAR-10	93.13	93.21	0.08	True
			CIFAR-10.1	84.70	85.70	1.00	True
	20	False	CIFAR-10	79.00	81.54	2.54	True
			CIFAR-10.1	63.90	68.15	4.25	True
		True	CIFAR-10	91.68	91.23	0.45	False
			CIFAR-10.1	82.90	83.15	0.25	True
3	20	False	CIFAR-10	71.15	67.69	3.46	False
			CIFAR-10.1	57.95	54.80	3.15	False
		True	CIFAR-10	90.76	90.24	0.52	False
			CIFAR-10.1	80.95	81.35	0.40	True

Table 6. *Results of Experiment Trials Using Augmented Validation Data*

Gains (or losses) observed in accuracy between approaches for validation data seem to grow as the severity of the augmentation increases (e.g., N=3, M=20) and are most significant when the model does not undergo 50 epochs of refinement with unaugmented data. Unfortunately, the data points for which we observe the greatest impact of using augmented or unaugmented validation data are also associated with diminished accuracy scores. We thus fail to find any meaningful indication that augmenting the validation dataset aids in closing the distribution gap.

Complete details of our approach and results for all models can be found in our github repository.[9]

[9] https://github.com/danzisar/w210-capstone/blob/master/analysis/AugmentationValidationAnalysis.ipynb

Inferencing with Augmented Test Sets

As part of our experimentation, we ran trials to understand the impact that augmenting testsets, following the same procedure used to augment the dataset used to train the model, would have on accuracy. We ran the experiment, testing each of the four models against each of our four RandAugment configurations. Figures 18 and 19 display the observed results. The results have been broken into two scatter plots due to the significant impact the refine phase of our experimental approach has on the results. Figure 18 shows the results when we evaluate with models that have been refined with unaugmented CIFAR-10 data. Figure 19 shows the results when we evaluate with models that have been refined with unaugmented CIFAR-10 data.

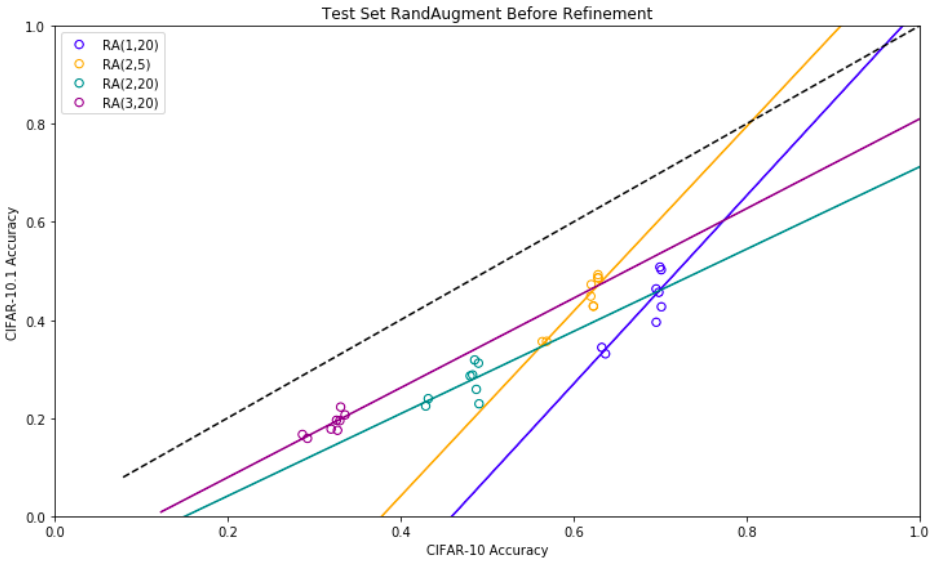


***Figure 18**. Accuracy Scores for Inferencing Performed on Augmented Testsets*

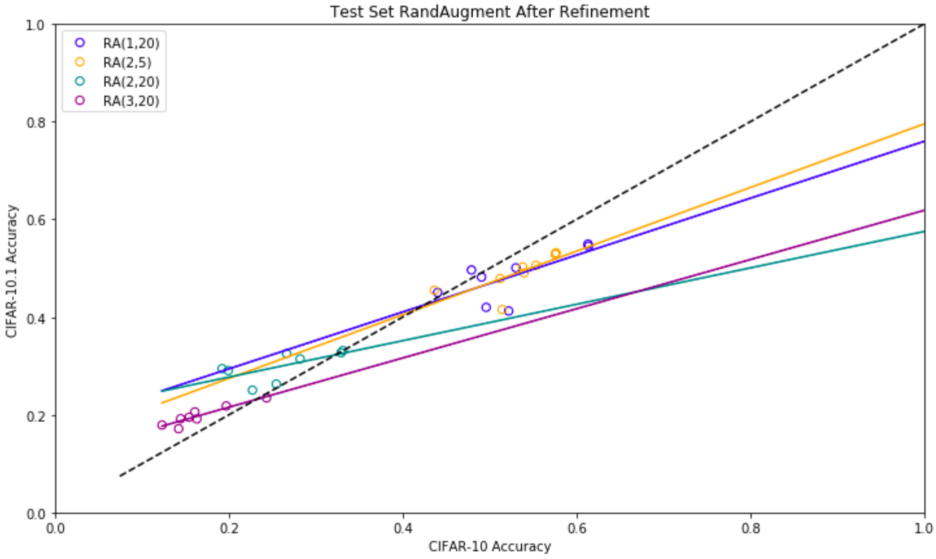


***Figure 19**. Accuracy Scores for Inferencing Performed on Augmented Testsets*

Several trends emerge from the observed results. First, the accuracy of all models drops dramatically. The four models for which we ran our experiments all had original accuracies ranging from 92.5 to 96.4 percent. The highest scoring model in our experiment is resnext_29_4x64 with an accuracy rate of 70.18 percent. This score is achieved when using the least augmented training and test set, where only one transformation was applied to the data using RandAugment. Across all models, we observe a consistent trend that the greater the augmentation applied, both in number of transformations and magnitude of the transformations, the greater the drop in accuracy.

Of interest in the results is the effect that the refine phase of our experimental approach has on a model's accuracy. Whether or not a model was refined for 50 epochs with unrefined data seems to significantly impact the model's inferencing abilities. The effect is exaggerated when discussing it in terms of the distribution gap because CIFAR-10 and CIFAR-10.1 seem to respond differently. Prediction scores for an augmented CIFAR-10 test set see significant drops when the model is refined. Prediction scores for an augmented CIFAR-10.1 test set improve when the model is refined. These divergent responses provide us with our only success in shrinking the distribution gap. In fact, when using the RandAugment(1,20) training and test sets, densenet's accuracy in predicting CIFAR-10.1 (49.65%) exceeds its accuracy in predicting CIFAR-10 (47.92%). Unfortunately, due to the severe drop in model performance, our ability to close the distribution gap here cannot be considered a success.

Based on these observed results, we ran an additional experiment with a single model (densenet) on a single augmented configuration (RandAugment(2,20)), to determine if we could improve model accuracy by extending the number of training epochs. However, we observed stable accuracy scores beginning at epoch 300.

Training Epochs	Training Dataset	Test Dataset	Loss	Accuracy
100	CIFAR-10 RA(2,20)	CIFAR-10 RA(2,20)	1.5644	0.4336
200	CIFAR-10 RA(2,20)	CIFAR-10 RA(2,20)	1.7676	0.4753
300	CIFAR-10 RA(2,20)	CIFAR-10 RA(2,20)	1.8928	0.4804
400	CIFAR-10 RA(2,20)	CIFAR-10 RA(2,20)	1.9671	0.4834
500	CIFAR-10 RA(2,20)	CIFAR-10 RA(2,20)	1.9929	0.4818
600	CIFAR-10 RA(2,20)	CIFAR-10 RA(2,20)	1.9850	0.4827
700	CIFAR-10 RA(2,20)	CIFAR-10 RA(2,20)	1.9831	0.4818
800	CIFAR-10 RA(2,20)	CIFAR-10 RA(2,20)	1.9949	0.4816
900	CIFAR-10 RA(2,20)	CIFAR-10 RA(2,20)	2.0125	0.4826
1000	CIFAR-10 RA(2,20)	CIFAR-10 RA(2,20)	1.9957	0.4831
100	CIFAR-10 RA(2,20)	CIFAR-10.1 RA(2,20)	2.4279	0.2295
200	CIFAR-10 RA(2,20)	CIFAR-10.1 RA(2,20)	3.0293	0.2685
300	CIFAR-10 RA(2,20)	CIFAR-10.1 RA(2,20)	3.9590	0.2775
400	CIFAR-10 RA(2,20)	CIFAR-10.1 RA(2,20)	4.1677	0.2880
500	CIFAR-10 RA(2,20)	CIFAR-10.1 RA(2,20)	4.2094	0.2870
600	CIFAR-10 RA(2,20)	CIFAR-10.1 RA(2,20)	4.2083	0.2900
700	CIFAR-10 RA(2,20)	CIFAR-10.1 RA(2,20)	4.4202	0.2855
800	CIFAR-10 RA(2,20)	CIFAR-10.1 RA(2,20)	4.3584	0.2910
900	CIFAR-10 RA(2,20)	CIFAR-10.1 RA(2,20)	4.3936	0.2910
1000	CIFAR-10 RA(2,20)	CIFAR-10.1 RA(2,20)	4.3737	0.2905

***Table 7**. Accuracy Scores for Inferencing Performed on Augmented Testsets*

Greater detail and analysis on this experiment can be found in our github repository[10].

[10] https://github.com/danzisar/w210-capstone/blob/master/analysis/AugmentedTestsetAnalysis.ipynb