Generalizing ResNet to Various Transformations

Albert Qi
Harvard College
Cambridge, Massachusetts, USA
albertqi@college.harvard.edu

ABSTRACT

As machine learning has grown, image classification has become more and more complex. Models have increased in size, and datasets have gotten larger. However, there are many situations in which images themselves may be distorted. Maybe an object is perceived from a different perspective, or perhaps different lightings throughout the day cause shifts in saturation and hue. Models such as ResNet-18 are not very robust to these transformations. Thus, I tackle two fundamental questions: To which transformations, if any, is ResNet-18 robust, and can ResNet-18 be improved to generalize to various transformations?

To answer these questions, I first test ResNet-18 on the CIFAR-10 [2] test set, resulting in an accuracy of 0.81. Afterward, I rerun the tests under a diverse set of transformations, showing that accuracy drops significantly. I then develop a more robust and generalizable model by retraining ResNet-18 on the CIFAR-10 dataset with random transformations applied at runtime. This results in an increase in accuracy for all transformations except the identity, indicating that training under a distorted dataset is highly beneficial. Finally, I test if robustness to a specific transformation is generalizable to others as well and show that, regardless of the transformation, robustness tends to generalize very well to photometric transformations.

All source code can be accessed at https://github.com/albertqi/robust-resnet18.

1 INTRODUCTION

Image classification is an incredibly powerful tool, enabling machines to interpret and categorize the contents within images. This capability has revolutionized many industries, with use cases ranging from disease identification and medical diagnostics to autonomous vehicles that rely on real-time image classification to navigate safely. Yet, as more and more processes become reliant on image classification, its robustness to errors and fluctuations becomes more and more crucial as well.

Imagine a factory that produces many different toys. As these toys are produced, they move along a single conveyor belt that has a camera somewhere to the side. This camera takes a photo as an object passes through its field of view, and the object in the photo is then classified as one of many different toys.

Now, suppose that the camera goes out of focus due to an accumulation of dust on the camera lens. In this case, can the classifier still recognize the toys that are being produced? If not, then such a problem could easily lead to mislabeled toys, inventory issues, and errors in production records and statistics.

Alternatively, suppose that the toys are not perfectly oriented on the conveyor belt but randomly rotated instead. Will this rotation cause significant issues in the classifier? If so, can the classifier be trained to recognize the rotated toys? These are some important questions to address and may help prevent major issues from arising in situations beyond just a factory's production line.

To address these issues, I begin by testing the loss and accuracy of ResNet-18 on the CIFAR-10 test set, giving me a baseline accuracy of 0.81. Then, I rerun the model on the test set under a variety of transformations, demonstrating that accuracy decreases drastically. The model performs the worst under GaussianBlur and the best under RandomPosterize, resulting in accuracies of 0.28 and 0.68, respectively. Afterward, I retrain ResNet-18 on the CIFAR-10 dataset with random transformations applied at runtime, giving me a more robust model. Accuracy only decreases by 1 percentage point on the untransformed CIFAR-10 test set, and accuracy increases significantly under the set of transformations. The new model performs the worst under ElasticTransform and the best under RandomPosterize, resulting in accuracies of 0.52 and 0.75, respectively. Finally, I retrain ResNet-18 under one specific transformation and test the resulting model's robustness to all transformations. I repeat this for every single transformation in order to see whether or not robustness to a specific transformation can be generalized to other transformations, too.

Overall, I make the following contributions within this paper:

(1) I highlight the need to improve the robustness and generalizability of ResNet-18.
(2) I test the current robustness of ResNet-18 on a diverse set of transformations applied to the CIFAR-10 dataset, demonstrating that performance under these distortions is very poor.

(3) I develop and evaluate a model that is more robust to such transformations without losing any significant amount of accuracy on the untransformed dataset.
(4) I test if robustness to one specific transformation is generalizable to other transformations as well, showing that robustness tends to generalize very well to photometric transformations.

2 BACKGROUND

Improving the robustness of machine learning models is not an entirely new topic. Rather, there are some existing literature that address similar issues.

2.1 Dataset Generation

Musat et al. [4] highlight the issue of adverse weather significantly impacting the sensors used in autonomous vehicles. As a result, they argue that it is paramount to have a dataset of images under a diverse set of weather conditions. They generate such a dataset by utilizing GAN and CycleGAN architectures to create images based on the Cityscapes dataset under various weather conditions.

Halder et al. [1] tackle the same issue but use a physical particle simulator instead of GAN and CycleGAN architectures to inject rain and fog into existing images. This results in a set of augmented images based on the Kitti and Cityscapes datasets.

While both of these are interesting and insightful methods of generating datasets, they are (1) only focused on weather conditions, and (2) overcomplicated for the purposes of my project. Instead, I use the `v2` module from the `torchvision.transforms` package, which contains a large set of transformations that can easily be applied to any dataset of images.

2.2 Improving Generalizability

Besides testing how loss and accuracy change under certain transformations, I also develop a model that is more generalizable and robust to these types of distortions.

Madan et al. [3] emphasize the importance of aligning human and machine vision by ensuring similarity in their visual diets. By incorporating scene context and a wide variety of real-world transformations into their datasets, they are able to develop a model that generalizes better to transformations such as lighting, viewpoint, and material changes.

Furthermore, Yang et al. [7] show that, in theory, there need not be any tradeoff between accuracy and robustness. Thus, it should be possible for a robust model to be perfectly accurate. However, they emphasize that developing such a model is challenging and may require methods that are used infrequently.

Lastly, [5, 6] demonstrate that while adversarially-trained ML models perform better than their standard counterparts for downstream image classification tasks, they do not actually retain any robustness when the downstream datasets are distorted.

With these results in mind, I train ResNet-18 on a version of CIFAR-10 that includes many real-world transformations. Additionally, I am careful about overfitting to CIFAR-10, as that may cause poor performance on images from new datasets.

3 DESIGN

I begin by testing ResNet-18 on the CIFAR-10 dataset under several different transformations. From these insights, I develop a model that is robust to these distortions. Lastly, I test the generalizability of robustness to specific transformations.

3.1 Testing ResNet-18

I use the CIFAR-10 test set in order to test the robustness of ResNet-18. Using the pretrained weights of ResNet-18, I pick a transformation to apply on the dataset. After applying the transformation, I also preprocessing the dataset using basic inference transforms. Then, I run one iteration of testing, repeating this process for each of the following transformations:

```
TRANSFORMS = [
    # No Transformation.
    v2.Identity(),

    # Geometric Transformations.
    v2.ElasticTransform(alpha=200.0, sigma=5.0),
    v2.RandomPerspective(distortion_scale=0.5,
                         p=1.0),
    v2.RandomRotation(degrees=180.0),

    # Photometric Transformations.
    v2.ColorJitter(brightness=0.5, contrast=0.5,
                   saturation=0.5, hue=0.5),
    v2.GaussianBlur(kernel_size=(9, 9),
                    sigma=8.0),
    v2.RandomInvert(p=1.0),
    v2.RandomPosterize(bits=2, p=1.0),
]
```

Figure 1 displays some examples of the images in the CIFAR-10 dataset under a variety of transformations.

3.2 Retraining ResNet-18

In order to develop a model that is robust to these transformations, I retrain ResNet-18 on the CIFAR-10 dataset with transformations randomly applied at runtime. Any transformation out of the aforementioned list of transformations is

Figure 1: Examples of Image Transformations

equally likely to be chosen for any given image. After the random transformation is applied, I also apply basic inference transforms. I then retrain ResNet-18 until convergence.

After the model converges, I retest its performance on the CIFAR-10 dataset under the previously mentioned transformations. This allows me to determine how much the model improved, if at all.

3.3 Generalizability of Robustness

To see whether or not robustness to one specific transformation can be generalized to others as well, I first pick a transformation to apply on the CIFAR-10 dataset. Note that unlike in Section 3.2, I do not apply a random transformation to each image but instead apply a specific transformation to the entire dataset. Afterward, I also apply basic inference transforms on the dataset. Then, I retrain ResNet-18 until convergence.

After reaching convergence, I retest the model's performance on the CIFAR-10 dataset under the aforementioned set of transformations. This entire process is repeated for every single transformation.

Table 1: Loss and Accuracy Under Various Transformations for Pretrained ResNet-18

Transformation	Loss	Accuracy
None	0.55	0.81
ElasticTransform	2.65	0.30
RandomPerspective	1.51	0.50
RandomRotation	2.26	0.38
ColorJitter	1.10	0.66
GaussianBlur	2.68	0.28
RandomInvert	1.95	0.44
RandomPosterize	1.01	0.68

4 EVALUATION

I begin by evaluating the pretrained ResNet-18 model on the CIFAR-10 dataset under several different transformations. Then, after retraining a robust model, I evaluate the new model on a distorted CIFAR-10 dataset. Lastly, I evaluate the generalizability of robustness to specific transformations.

4.1 Experimental Setup

I run all of my tests with excess memory on an 8-core GPU that is integrated into the Apple M1 chip. All of my training and testing is performed with a batch size of 128. Training and testing data are split at a 5:1 ratio.

The primary evaluation metrics that I use are loss and accuracy, but accuracy is a little more important since loss can be rather abstract. I know that my robust model is successful if the accuracy of my model on a transformed dataset is greater than that of the original ResNet-18 model under the same transformation.

4.2 Testing ResNet-18

I first test the pretrained ResNet-18 model on a diverse set of transformations. Table 1 shows that ResNet-18 has an accuracy of 0.81 on the untransformed CIFAR-10 dataset. Out of all of the other transformations, GaussianBlur results in the lowest accuracy of 0.28, and RandomPosterize results in the highest accuracy of 0.68.

This is not surprising, as gaussian blur distorts an image to the point where it is almost unrecognizable. Because the shapes and textures in the image are so heavily transformed, both humans and ML models are unable to rely on shape and

Table 2: Loss and Accuracy Under Various Transformations for Robust ResNet-18

Transformation	Loss	Accuracy
None	0.57	0.80
ElasticTransform	1.34	0.52
RandomPerspective	0.93	0.67
RandomRotation	1.26	0.55
ColorJitter	0.79	0.73
GaussianBlur	1.06	0.63
RandomInvert	0.86	0.71
RandomPosterize	0.75	0.75

Table 3: Changes in Loss and Accuracy Under Various Transformations from Pretrained to Robust ResNet-18

Transformation	Δ Loss	Δ Accuracy
None	0.02	−0.01
ElasticTransform	−1.31	0.22
RandomPerspective	−0.58	0.17
RandomRotation	−1.00	0.17
ColorJitter	−0.31	0.07
GaussianBlur	−1.62	0.35
RandomInvert	−1.09	0.27
RandomPosterize	−0.26	0.07

texture cues to classify objects within such an image. For example, the blurred image in Figure 1 is very unrecognizable, as there are no edges that allow a human or machine to easily identify an object. On the other hand, a simple random posterize does not change the shape or texture but only alters the color instead. The posterized image in Figure 1 is clearly an automobile. Even though the color is distorted, the shapes and textures still allow humans and machines to identity the wheels on the automobile. Thus, it is unsurprising that RandomPosterize performs the best out of the transformations.

Additionally, the pretrained ResNet-18 model seems to be fairly robust to photometric transformations (with the exception of GaussianBlur) but not so much to geometric transformations. Transformations like ColorJitter, RandomInvert, and RandomPosterize only alter the color properties of an image (e.g., brightness, contrast, saturation, hue). The performances under these transformations are reasonably okay, with accuracies of 0.66, 0.44, and 0.68, respectively. Compare this to ElasticTransform, which does not alter the color of an image but instead changes its geometric properties. The performance under this transformation is much worse, with an accuracy of 0.30. With such small images, it seems that any distortion in an image's geometry may drastically change the performance of the classifier.

4.3 Retraining ResNet-18

I then retrain ResNet-18 on the CIFAR-10 dataset with transformations randomly applied at runtime. Table 2 shows that performance drastically increases across all areas, except in the case where there is no transformation and instead, accuracy decreases by 1 percentage point from 0.81 to 0.80. With the robust model, ElasticTransform results in the lowest accuracy of 0.52, and RandomPosterize results in the highest accuracy of 0.75. Again, the model seems to perform better under photometric transformations than it does under geometric transformations.

The 1 percentage point decrease in accuracy from 0.81 to 0.80 on the untransformed dataset is negligible, meaning the robust model still has a lot of predictive power on the base CIFAR-10 dataset. Now, however, the model also has strong predictive power in scenarios where the dataset is distorted. Table 2 shows that every single transformation results in an accuracy above 0.50. This is quite an improvement from the pretrained model.

It makes a lot of sense that the retrained model performs better on all transformations except the identity, as in practice, there may be a slight tradeoff between accuracy and robustness. In order to improve the model's robustness to transformations, it needs to sacrifice a little bit of accuracy on the untransformed CIFAR-10 dataset. This is still very advantageous, though, as introducing distortions into the dataset significantly increases the model's robustness to these distortions.

Additionally, Table 3 outlines the changes in loss and accuracy from pretrained ResNet-18 to robust ResNet-18. Note that ColorJitter and RandomPosterize increase the least in accuracy at $\Delta = +0.07$. This is likely because pretrained ResNet-18 already performs okay under these transformations, meaning there is less opportunity for improvement. On the other hand, GaussianBlur results in the highest change in accuracy at $\Delta = +0.35$, likely because it starts at the lowest accuracy.

Ultimately, the robust model gives us a drastic increase in accuracy under a set of distortions without losing any significant amount of accuracy on the untransformed dataset. Thus, training a model under these transformations is incredibly beneficial.

4.4 Generalizability of Robustness

Finally, I evaluate the generalizability of robustness for every transformation. Tables 4 and 5 show the loss and accuracy of ResNet-18 trained under specific transformations, respectively.

Table 4: Loss of ResNet-18 Trained Under Specific Transformations

	None	ElasticTransform	RandomPerspective	RandomRotation	ColorJitter	GaussianBlur	RandomInvert	RandomPosterize
None	0.55	2.65	1.51	2.26	1.10	2.68	1.95	1.01
ElasticTransform	0.98	1.08	1.06	1.98	1.83	2.39	2.72	1.69
RandomPerspective	0.65	2.08	0.77	2.28	1.30	2.74	2.24	1.17
RandomRotation	0.76	2.17	1.38	0.95	1.51	2.89	2.33	1.17
ColorJitter	0.54	2.79	1.63	2.67	0.60	2.84	1.19	0.87
GaussianBlur	2.48	6.75	3.74	4.72	3.41	0.95	4.89	3.73
RandomInvert	1.38	3.76	2.72	3.26	1.47	4.03	0.54	1.86
RandomPosterize	0.69	3.39	1.75	2.59	1.29	3.35	2.44	0.61

Table 5: Accuracy of ResNet-18 Trained Under Specific Transformations

	None	ElasticTransform	RandomPerspective	RandomRotation	ColorJitter	GaussianBlur	RandomInvert	RandomPosterize
None	**0.81**	0.30	**0.50**	0.38	**0.66**	0.28	0.44	**0.68**
ElasticTransform	**0.66**	**0.62**	**0.63**	0.39	0.45	0.32	0.22	0.47
RandomPerspective	**0.78**	0.41	**0.73**	0.38	**0.61**	0.28	0.36	**0.63**
RandomRotation	**0.74**	0.33	**0.52**	**0.67**	**0.53**	0.24	0.31	**0.60**
ColorJitter	**0.82**	0.25	0.47	0.31	**0.80**	0.25	**0.62**	**0.71**
GaussianBlur	0.45	0.15	0.26	0.23	0.34	**0.68**	0.21	0.31
RandomInvert	**0.58**	0.14	0.26	0.21	**0.56**	0.12	**0.81**	0.45
RandomPosterize	**0.77**	0.25	0.46	0.34	**0.62**	0.20	0.37	**0.79**

Accuracies above 0.50 are shown in bold.

According to Table 5, it seems that ElasticTransform, RandomRotation, and GaussianBlur are very challenging transformations for ResNet-18. Only by training under each specific transformation can the model reach an accuracy above 0.50, suggesting that these transformations are ones that must be included in training a robust model. This is not too surprising, as these are also the exact transformations that result in the three lowest accuracies for the pretrained model (see Table 1).

On the other hand, ColorJitter and RandomPosterize seem to be the easiest transformations for ResNet-18. Again, these are both photometric transformations under which the pretrained model already seems to perform fairly well. Thus, there may not be the need to include ColorJitter and RandomPosterize when training a robust model.

Lastly, note that RandomRotation is the only transformation under which training ResNet-18 results in accuracies of at least 0.50 in five total areas (including no transformation). This implies that robustness to RandomRotation generalizes very well to other transformations, too.

Ultimately, the generalizability of robustness seems to depend on the transformation, but regardless of the transformation, robustness always seems to generalize easily to photometric transformations.

5 FUTURE WORK

In the future, I would like to perform more tests on the generalizability of robustness. Namely, I want to answer the question, what would be the performance of a model that is only trained on ElasticTransform, RandomRotation, and RandomPosterize? Given that these transformations are the most challenging for ResNet-18, would training under just these transformations be enough to guarantee robustness in all areas? I would like to perform more narrow tests in order to pinpoint exactly which transformations are actually most crucial for training a robust model.

Additionally, I would like to repeat my tests with more transformations, including more geometric and more photometric transformations. This may allow me to gain more insight from my tests and would allow me to compare performances solely within geometric transformations or solely within photometric transformations.

Lastly, I would like to see how switching to ResNet-34 or ResNet-50 would affect my results. Would adding more layers allow for better performance of a robust model? I would think so, but actually performing tests with ResNet-34 or ResNet-50 would be necessary to definitively answer this question.

6 CONCLUSION

Image classification has become increasingly popular, with applications ranging across many different industries. From facial recognition to disease identification, an increasing amount of our lives now relies on image classification. Yet, as the world continues to change in unpredictable ways, it is paramount that classifiers are able to handle distorted images as well.

Thus, I test the ResNet-18 model on the CIFAR-10 dataset in order to see (1) how robust the model is to a set of transformations, and (2) whether or not I can increase the robustness of the model.

I begin by running one iteration of testing on the CIFAR-10 test set using the pretrained ResNet-18 model, resulting in a baseline accuracy of 0.81. Next, I rerun the tests under a variety of transformations, demonstrating that accuracy decreases drastically. I then retrain ResNet-18 by applying random transformations at runtime, giving me a more robust model with significantly better accuracies across all transformations except the identity. This highlights the advantages of including distortions in the training set. Finally, I test if robustness to specific transformations is generalizable to other transformations as well, demonstrating that robustness tends to generalize fairly well to photometric transformations.

REFERENCES

[1] Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. 2019. Physics-Based Rendering for Improving Robustness to Rain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 10203–10212.

[2] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images. https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf

[3] Spandan Madan, You Li, Mengmi Zhang, Hanspeter Pfister, and Gabriel Kreiman. 2024. Improving Generalization by Mimicking the Human Visual Diet. (2024). arXiv:2206.07802 https://arxiv.org/abs/2206.07802

[4] Valentina Musat, Ivan Fursa, Paul Newman, Fabio Cuzzolin, and Andrew Bradley. 2021. Multi-Weather City: Adverse Weather Stacking for Autonomous Driving. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2906–2915. https://doi.org/10.1109/ICCVW54120.2021.00325

[5] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. 2020. Do Adversarially Robust ImageNet Models Transfer Better? *Advances in Neural Information Processing Systems* 33 (2020), 3533–3545.

[6] Yutaro Yamada and Mayu Otani. 2022. Does Robustness on ImageNet Transfer to Downstream Tasks?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9215–9224.

[7] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. 2020. A Closer Look at Accuracy vs. Robustness. *Advances in Neural Information Processing Systems* 33 (2020), 8588–8601.