EECE 5564 Project Update

Group 01

November 23, 2022

---

1 Project Updates

We begin with feature extraction for each image in CIFAR10 as well as CIFAR10c. For simplicity, we only consider corruption severity levels 1 (low), 3 (medium), and 5 (high). We preprocess the images by first upsampling them to 224x224 and normalizing them using the mean/std deviation of the CIFAR10 training dataset. We use a ImageNet-pretrained ResNet18 model provided by the torchvision library as our feature extractor, which outputs a 512-dimensional feature for each input image. Our objective in the first part of this project was to visualize this feature space and understand what kind of analysis could be done.

1.1 Exploratory Data Analysis

We begin our analysis by visualizing the feature representations using PCA. In figure 1, we see that while there is some clear separation between classes (i.e. classes 0 and 1), the blobs are not very well separated. For future visualization, we may investigate more robust dimensionality reduction measures such as TSNE in order to better understand the latent structure.

1.1.1 K-Means

We investigate the performance of a baseline clustering algorithm (K-means) on the extracted features of the data and visualize it using dimensionality reduction (with the help of PCA) and we repeat the process for some of the corrupted data to try and answer the question of how these corruptions affects the data. As can be seen in Figure 3, the extracted features are not well separated, and from this analysis alone the effect of the corruptions is difficult to measure.

1.1.2 Pairwise Comparison

To simplify the analysis, we took each corruption one at a time and visualized the features with and without the corruption, fitting PCA only on the clean data. We see that some corruptions do not move the features around much at all (Gaussian Blur), but others result in a large translation of the feature space (Shot Noise). We highlight this result in figure **??**. In other words, as different corruptions lead to different types of preturbations in the feature space. Future work may look to predict the type of corruption such that we know how to adjust our classifier.

2 Classification Baselines

As a baseline metric, we evaluate logistic regression with 4-fold cross validation to compute the accuracy on CIFAR10 and CIFAR10-C. Our baseline accuracy is 82.25 on CIFAR10 and (82.54, 73.71, 62.15) on CIFAR10-C. Note that in this baseline we are fitting the models on the corrupted samples, which will achieve a higher accuracy than a model trained only on clean data forced to generalize to corrupted samples. Both metrics may be valid, but this gives us an initial baseline of performance.

3 Downstream Classification Performance

We also explored the effect of each corruption at each severity level on downstream classification. For simplicity, we use logistic regression as our model and report the mean accuracy using 4-fold cross validation using only the clean dataset and the data from one other corruption. We use only one severity level at a time, meaning the number of clean and corrupted samples are equal. These results are somewhat correlated with section 1.1.2, as corruptions which had the biggest translation in PCA-space typically had lower accuracy (again, compare Gaussian Blur and

Corruption	(Mild, Medium, High)
brightness	(0.82, 0.82, 0.79)
contrast	(0.81, 0.77, 0.69)
defocus_blur	(0.81, 0.76, 0.69)
elastic_transform	(0.75, 0.72, 0.64)
fog	(0.81, 0.77, 0.66)
frost	(0.79, 0.73, 0.7)
gaussian_blur	(0.81, 0.73, 0.67)
gaussian_noise	(0.73, 0.61, 0.57)
glass_blur	(0.68, 0.66, 0.61)
impulse_noise	(0.75, 0.64, 0.55)
jpeg_compression	(0.73, 0.67, 0.64)
motion_blur	(0.77, 0.69, 0.66)
pixelate	(0.8, 0.76, 0.68)
saturate	(0.78, 0.82, 0.77)
shot_noise	(0.75, 0.65, 0.58)
snow	(0.78, 0.72, 0.68)
spatter	(0.8, 0.73, 0.73)
speckle_noise	(0.76, 0.68, 0.6)
zoom_blur	(0.75, 0.74, 0.71)

Table 1: Per-Corruption/Severity Accuracy Using Logistic Regression.

Shot Noise). This makes sense as the decision boundary for the standard and corrupted datasets will be shifted from one another.

4 Planned Work

In future work, we will first explore alternative clustering and dimensionality reduction algorithms to gain a better intuition. We will also work off our findings in sections 1.1.2 and 3. to explore classifier robustness to specific shifts in feature space. It would also be benefitial to explore whether or not is is possible to classify the corruption type or severity for a given sample. Even if we are not able to create a better model for corrupted samples, knowledge of the type and severity of corruption may improve the uncertainty estimation of our model.

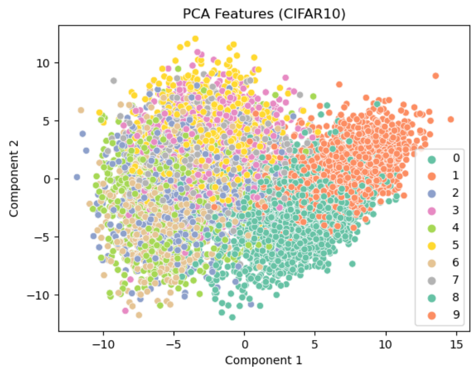
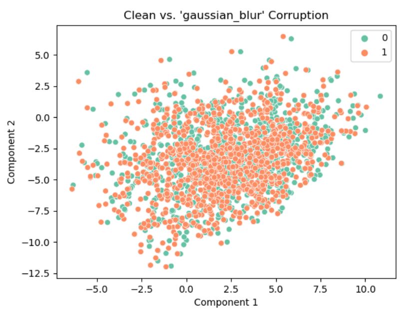
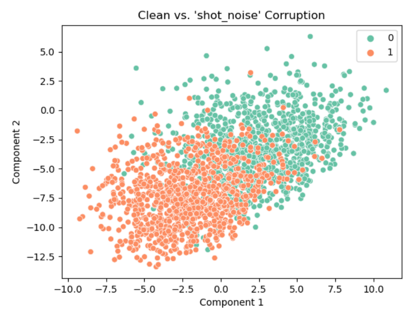


Figure 1: PCA Visualization of Features on CIFAR10 Clean Dataset

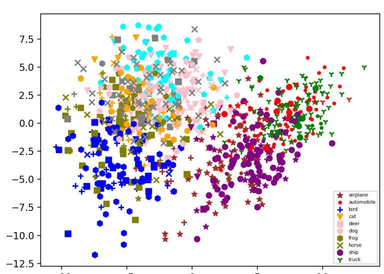


(a) Gaussian Blur PCA Features

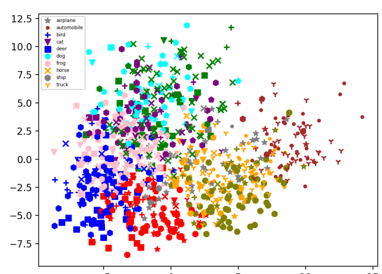


(b) Shot Noise PCA Features

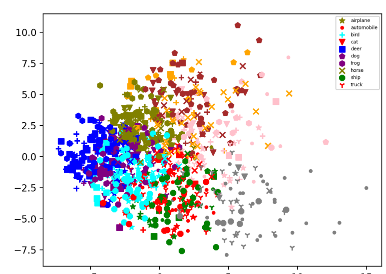
Figure 2: Comparison of Feature Shift Between Corruptions. 0-clean 1-corrupted.



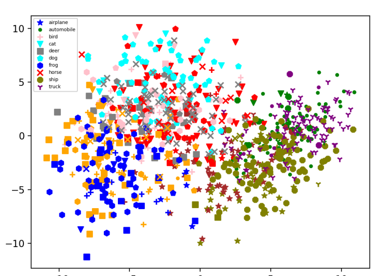
(a) Uncorrupted images



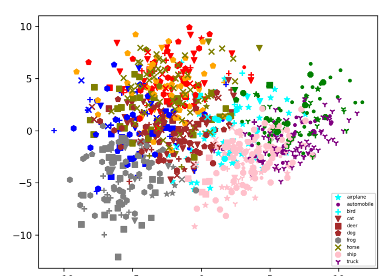
(b) Gaussian Noise Sev1



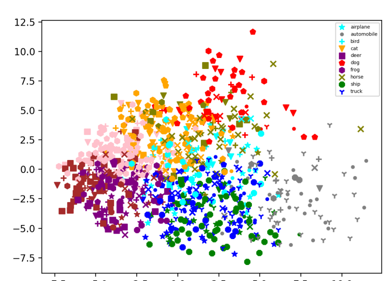
(c) Gaussian Noise Sev5



(d) Brightness Sev3



(e) Fog Sev3



(f) Gaussian Noise Sev3

Figure 3: Feature visualization of different image corruptions at different levels. In each image, the color of each point is the result of KMeans and the marker used indicated its true class, according to the legend