CS 182 Project Milestone: Computer Vision Project

Chethan Bhateja, Chirag Sharma, Varun Jadia
chetbhateja@berkeley.edu, shchirag@berkeley.edu, jvarun@berkeley.edu

In order to develop an image classification model for Tiny Imagenet that is robust to noisy data and adversarial perturbations and patches, we decided to adopt a transfer learning approach with a pre-trained Inception v3 model as the base. For our preliminary analysis, we wanted to first understand how the pre-trained Inception v3 model responds to explicitly augmented input data from Tiny Imagenet and compare the prediction quality with results for the original Tiny Imagenet data.

1 Data augmentation methods

The explicit data augmentation methods that we explored for the Tiny Imagenet images include up/down translations, right/left translations, vertical flips, horizontal flips, rotations by arbitrary angles, and the addition of Gaussian-distributed noise to each pixel. These lean more towards the 'common' type of image corruptions, i.e., what we would expect to see in noisy real-world data. For small enough perturbations, we should expect a robust model to perform well even on this augmented data since we are generally only shifting the main subject of the image, changing its orientation, or adding some noise to it – our final model should be capable of learning representations of objects in images that are independent of these characteristics.

We wrote code that generates an augmented dataset by taking each image in the original dataset and outputting the following 25 augmentations of it:

- Original image
- Image flipped horizontally
- Image flipped vertically
- 5 random horizontal translations of the image by x pixels, where x is uniformly sampled from $[-15, 15]$
- 5 random vertical translations of the image by y pixels, where y is uniformly sampled from $[-15, 15]$
- 5 copies of the image with Gaussian-distributed noise added to each pixel, where the noise has mean μ, uniformly sampled from $[-100, 100]$, and standard deviation σ, uniformly sampled from $[0, 10]$
- 7 rotations of the image by θ degrees, where θ is uniformly sampled from $[0, 180]$

Observe that for the $64 \times 64 \times 3$ images from Tiny Imagenet, the random horizontal and vertical translations are always relatively small perturbations. Similarly, since each pixel takes on a value from $0 - 255$, we expect the noise additions to be relatively small perturbations as well. On the other hand, the image is practically unchanged following any rotation, so we explore a wide range of possible rotations.

2 Pretrained Inception v3 set-up

We chose the Inception v3 model for it's unique architecture and good baseline performance (around 80% on the original Imagenet dataset). Moreover, PyTorch has a pretrained Inception v3 model ready to import and use directly for prediction. First, we decided to test the PyTorch model performance on Tiny Imagenet. Since the original Imagenet dataset has 1000 classes but Tiny Imagenet has only 200 classes, we changed the shapes of the output layers to match Tiny Imagenet – the final fully connected layer in the auxiliary net was changed to have dimensions $(768, 200)$ and the main fully connected layer was changed to have dimensions $(2048, 200)$. Moreover, since Inception v3 requires a specific normalization for the input data, we re-normalized our train and validation data to the required mean and standard deviation.

3 Fine tuning the model

Next, we fine-tuned the out-of-the-box model by re-training the weights for the final classification layers (the final fully connected layers that we replaced in the auxiliary net and the main net). Since the previous layers extract representations of the image in feature-space, we expect them to not be affected by the change from 1000 classes to 200 classes. Thus, we used the pretrained weights from earlier layers to extract features for the fully connected layers and then trained the fully connected layers for 3 epochs to obtain new weights for classification.

4 Results

After finetuning the model by training for 3 epochs, we ran prediction tasks on the training data, original validation data, and augmented validation data (with 25 augmentations of each original image, as described in Section 1). We obtained the following prediction accuracies:

- Training set accuracy: 60.41%
- Original validation set accuracy: 59.10%
- Augmented validation set accuracy: 39.31%

Clearly, the validation accuracy is much lower on the augmented set despite the augmentations being fairly small perturbations as discussed earlier. The training accuracy is also lower than we expected, but this is likely due to only training the classification layers for 3 epochs. Nevertheless, these results are encouraging as they illustrate a clear pitfall of state-of-the-art image classification models like Inception.

5 Next steps

So far, we have explored different data augmentation methods, fine-tuned a pretrained Inception v3 model, and tested its performance on augmented and non-augmented validation data. The natural next step is to explore the addition of augmented data to the training process to see if we get a significant boost in augmented validation set accuracy. In particular, if we train on randomly perturbed data, we want to see how much this affects the model's performance on randomly perturbed validation data. Moreover, it will be interesting to see if this behavior changes as we train our model for more epochs, try more augmentations, and increase the degree of perturbation. Other next steps include trying more data augmentation and noise addition techniques, testing against specific adversarial corruptions such as patch attacks, and trying other approaches such as improved loss-functions to make our model more robust.

We will follow the outline described in our project proposal, including the construction of an adversarial network trained to output images that our model is expected to perform poorly on and filter visualization in the convolutional layers to understand features of interest in adversarial images. We expect filter visualization to provide unique insight into what features are the most important in representing the content of an image – in particular, by maximally perturbing images such that the model is still able to fairly accurately classify them, we can understand what features are preserved in the image even after a large number of augmentations and adversarial attacks.