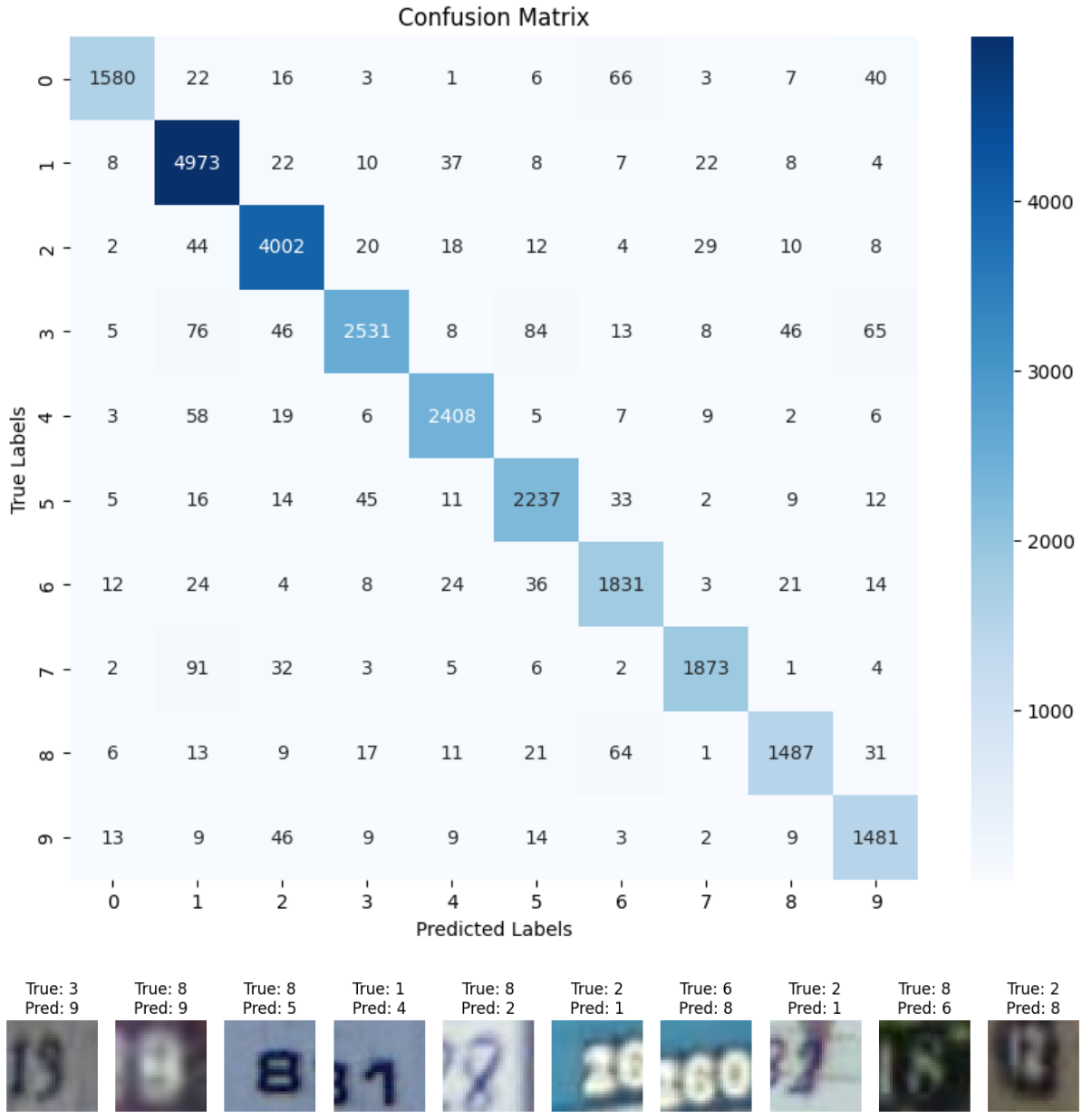
Machine Learning 2

Assignment 3

Part 1 - Training a CNN on SVHN

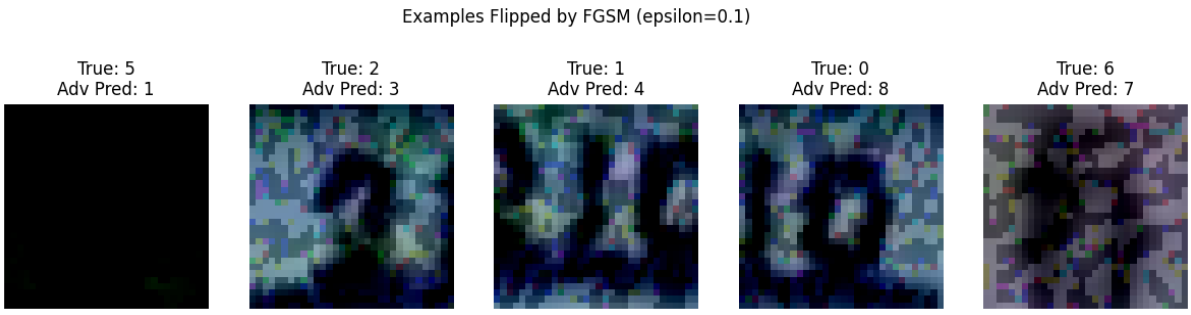
1.1) Analyze the performance of the model on the test set (e.g. through a confusion matrix). Display images that the model predicts incorrectly and their predicted classes. Discuss possible weaknesses of the model and their causes.

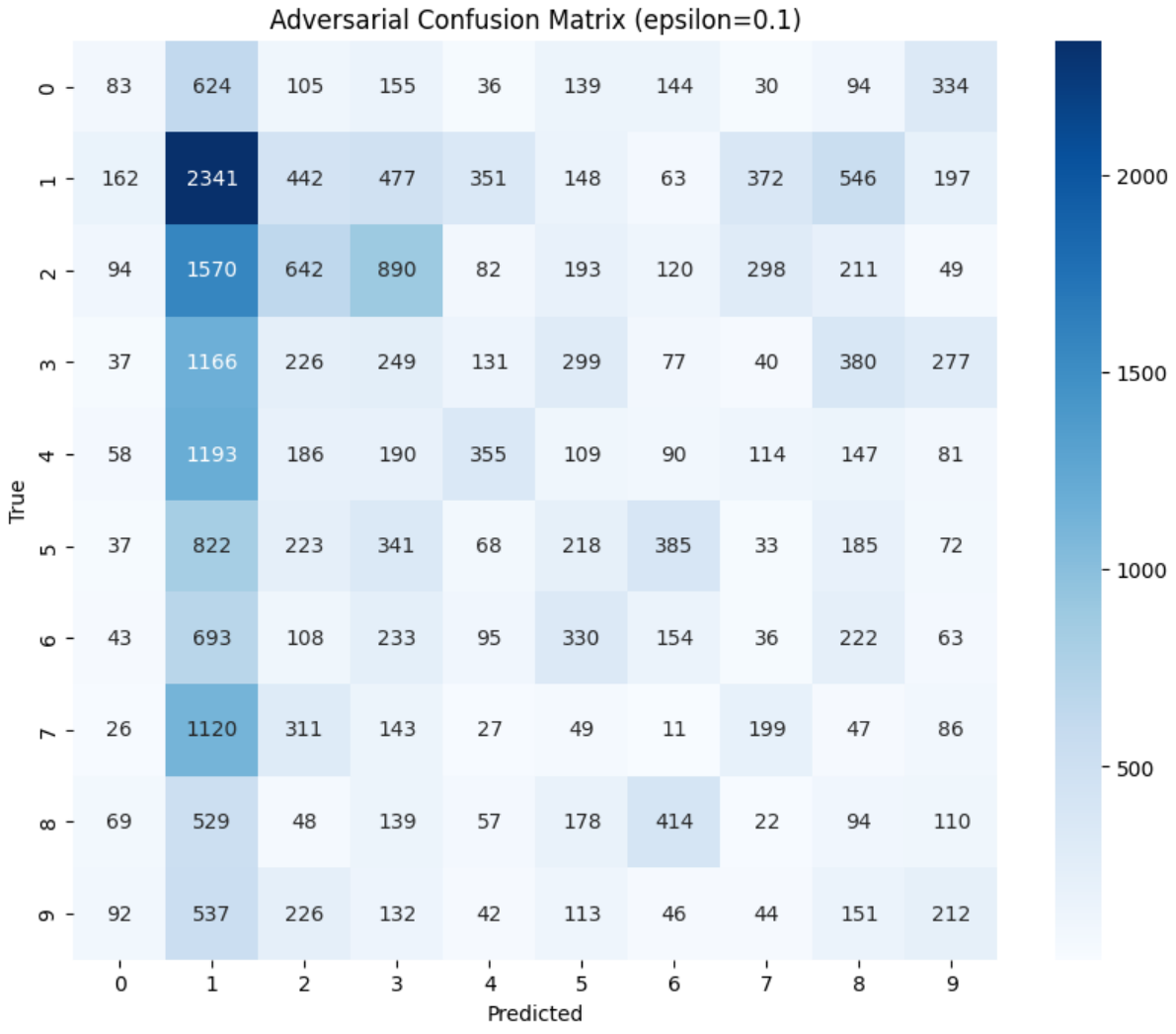


The model achieved 93.74% of final test accuracy. Overall the model achieves a very accuracy on the SVHN Dataset. Nevertheless, it seems that the model struggles more to predict the correct class between digits like 0 and 6 or 9 and 3. My explication is that they contain some shapes in common like rounded shapes. The model may rely too much on some specific shapes and is not robust enough to discriminate on thinner details.

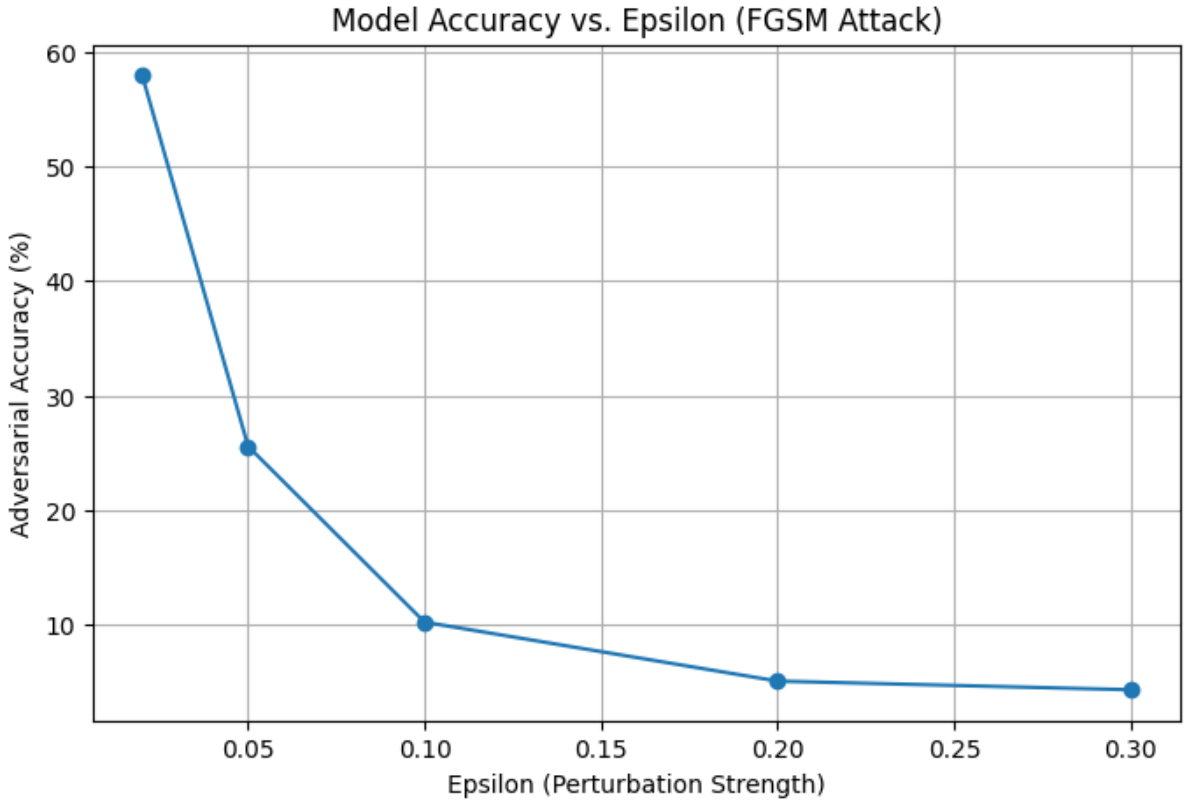
Part 2: Adversarial Attacks on our Model

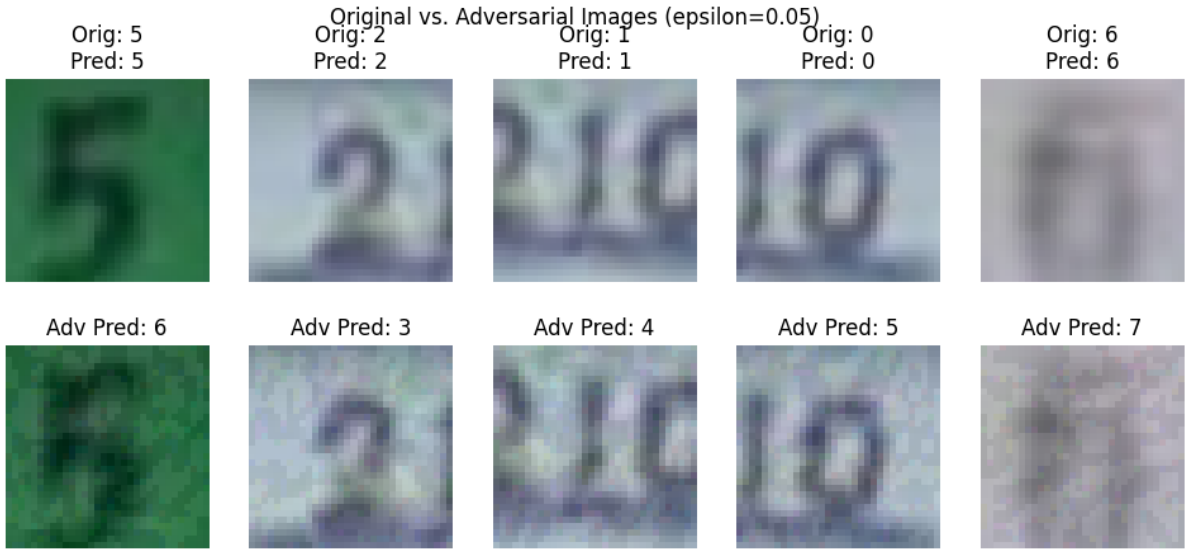
2.1) Visualize some images that the model got right before the perturbation and wrong after the attack. Create a confusion matrix of the output on the entire test set.

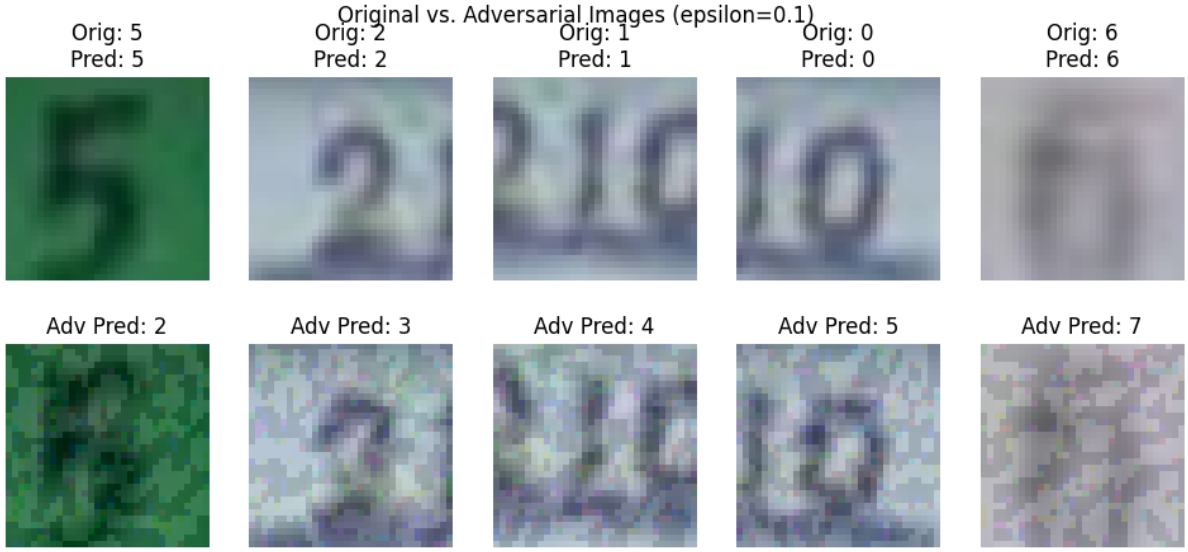


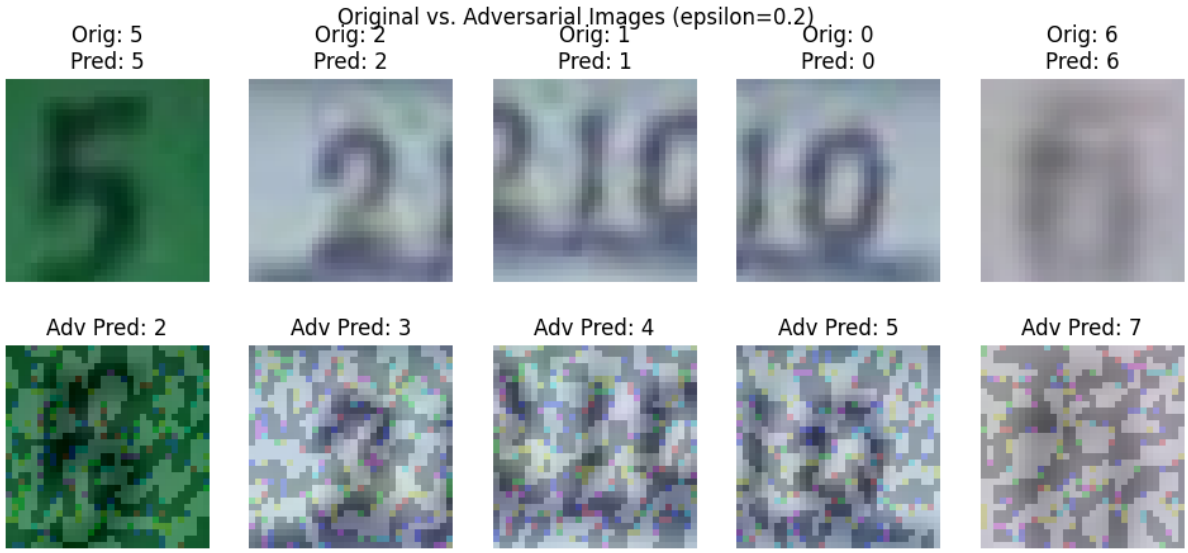


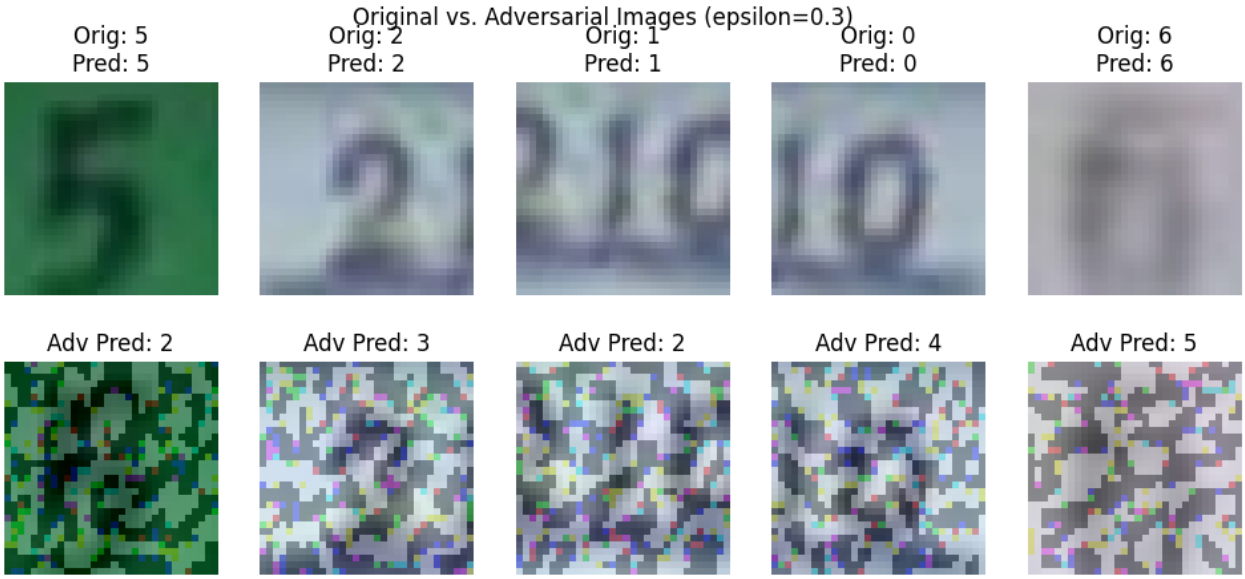
2.2) Test the function with different values of epsilon (at least 5) and plot the accuracy as a function of epsilon. For each epsilon, display the perturbed images with the model's classification. At what epsilon does it become harder for the human eye to correctly classify?




Original vs. Adversarial Images (epsilon=0.05)
Orig: 5
Pred: 5
Orig: 2
Pred: 2
Orig: 1
Pred: 1
Orig: 0
Pred: 0
Orig: 6
Pred: 6
Adv Pred: 6
Adv Pred: 3
Adv Pred: 4
Adv Pred: 5
Adv Pred: 7


Original vs. Adversarial Images (epsilon=0.1)
Orig: 5
Pred: 5
Orig: 2
Pred: 2
Orig: 1
Pred: 1
Orig: 0
Pred: 0
Orig: 6
Pred: 6
Adv Pred: 2
Adv Pred: 3
Adv Pred: 4
Adv Pred: 5
Adv Pred: 7

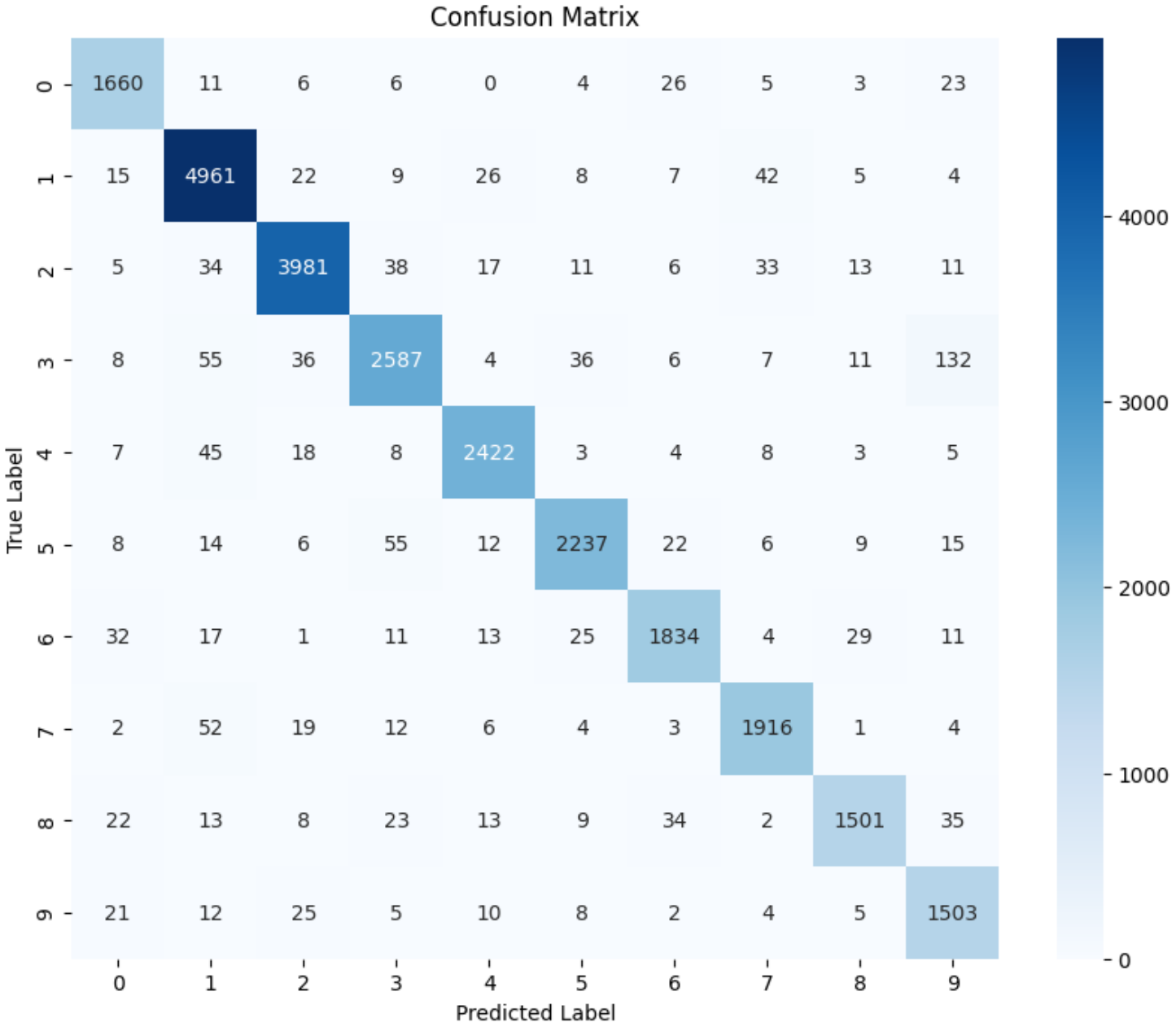

Original vs. Adversarial Images (epsilon=0.2)
Orig: 5
Pred: 5
Orig: 2
Pred: 2
Orig: 1
Pred: 1
Orig: 0
Pred: 0
Orig: 6
Pred: 6
Adv Pred: 2
Adv Pred: 3
Adv Pred: 4
Adv Pred: 5
Adv Pred: 7



As we can see above, from epsilon equals to 0.1, it becomes very harder for the human eye to correctly classify.

Part 3: Training our model using adversarial training

3.1) Display the confusion matrix along with some examples of images that the model classified incorrectly. Discuss the performance of the model now compared to before.



The model is now more robust because he went through adversarial training. The initial model achieved a poor accuracy of 10.21% in the adversarial setting with FGSM attack with epsilon = 0.1. Now, the model is achieving 80.90% on the adversarial setting. We can notice that the robust model succeeds to classify digits containing common shapes as 6 and 0 or 1 and 4 whereas the initial model failed to achieve this as discussed before.

Part 4: Contrastive Learning

4.1) When training an unsupervised contrastive learning model such as SimCLR, would we prefer to have a large or small batch size?

With SimCLR, we prefer to use large batch size as discussed in tirgoul. Contrastive Learning relies on “negative” examples. In contrastive frameworks, each image in the batch has augmented “positive” views of itself and many “negative” examples from other images in the batch. Having a large batch provides more negative samples, improving the quality of the contrastive signal.

4.2) In general, what possible evaluation metrics could be used in this task (unsupervised representation learning) to measure our model's performance?

Since SimCLR is trained in an *unsupervised* way, we do not optimize directly for labeled performance. We can apply a clustering algorithm like k-means for example on the learned embeddings, then measure unsupervised metrics (like Silhouette Score) or, if labels are available, measure how pure each cluster is.

4.3) When creating embeddings for images in the test set, how does the process differ from what we do in training?

When training SimCLR, we apply random augmentations to create multiple views of each image. We also pass each augmented view through the encoder and projection head to compute the contrastive loss.

When we want to obtain embeddings on test data, we either disable or use minimal deterministic augmentation. We forward pass the test image through the same encoder network. We use the encoder output before the projection head as the "embedding," or use the output of the projection head if that is the desired representation.

So, the process differs in augmentations and no contrastive loss at test time.

4.4) For each of the following image augmentations, explain whether or not we would like to use them in the SimCLR framework:

- Randomly cropping a fixed-size window in the image.

Yes. Random crops encourage spatial invariance. They also help the network learn robust features by seeing multiple cropped views of the same image.

- Enlarging the image to 128x128.

No. If the input dataset is 96×96, arbitrarily upsampling to 128×128 might introduce unwanted interpolation artifacts.

- Random rotation of the image.

May be useful but the rotation should be small because it can negatively impact the semantics in the picture

- Adding Gaussian noise.

Yes. Gaussian noise adds another channel of variation, forcing robustness to pixel-level perturbations. It's not the most common transformation in the *original* SimCLR setup compared to color jitter and random crop, but it's still a plausible augmentation.

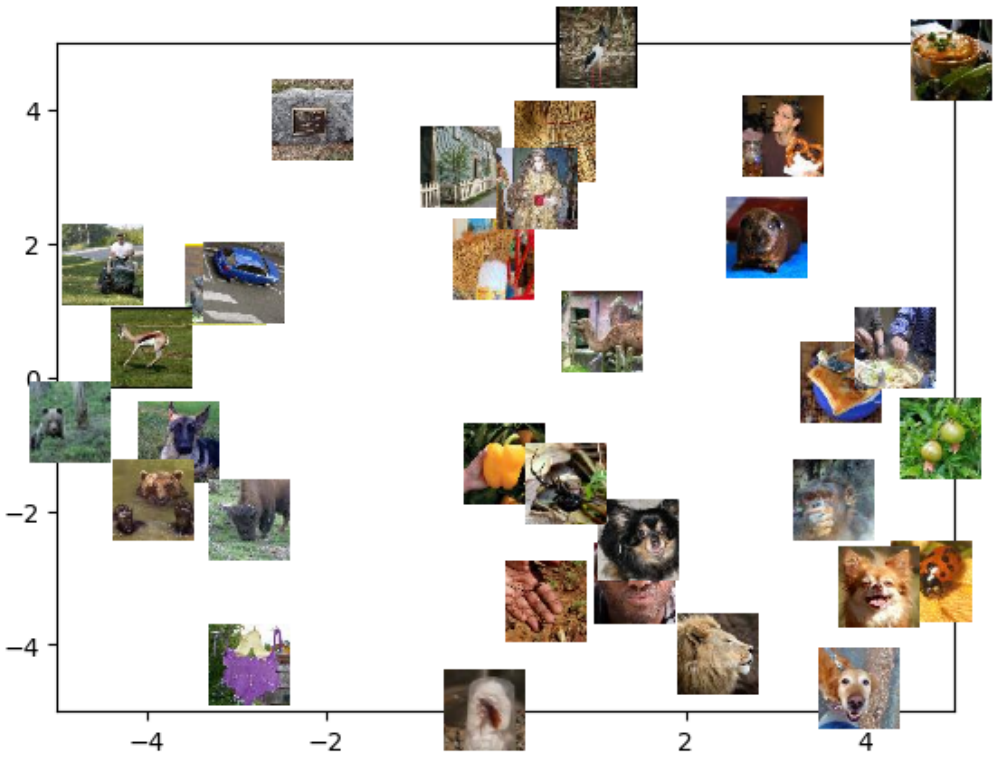
- Randomly changing the image's dimensions.

No. It may distort the image and change semantics.

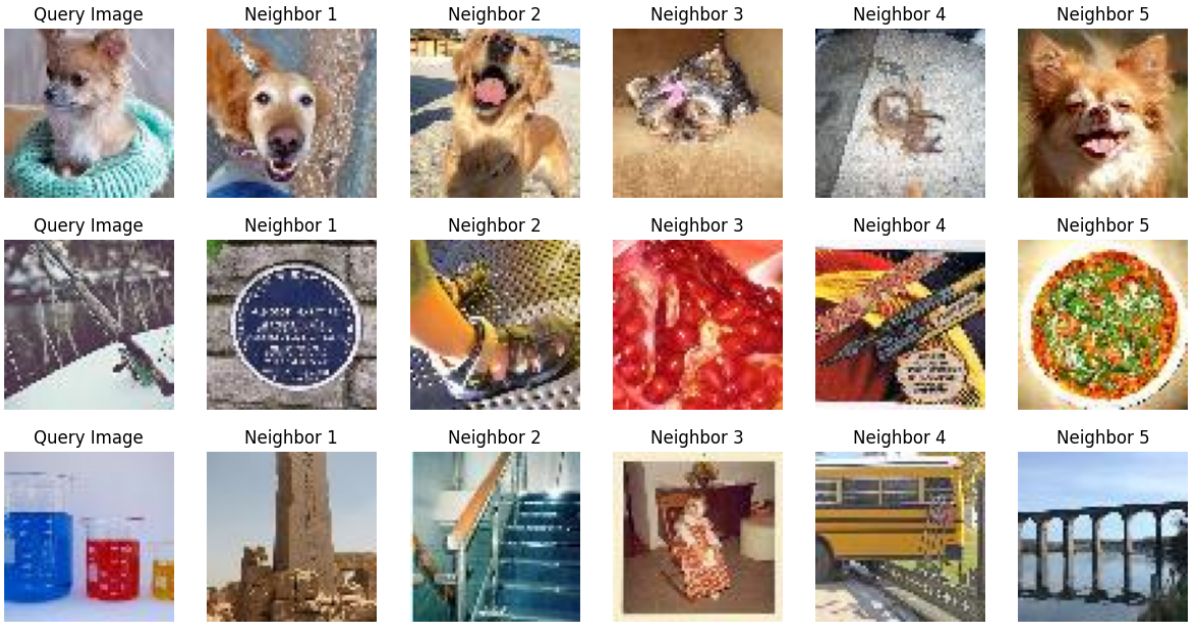
- Randomly converting the image to grayscale.

Yes. It forces the model to learn color-invariant features. This helps the model not to rely solely on color cues.

4.5) Show the model's performance on the test data



4.6) For some batch of the test loader, take 3 images in the batch. For each image, find and display the 5 images that have the closest embeddings to them. Do the chosen images make sense? If not, what could have possibly gone wrong with your model?



1st query : Nearest neighbors are mostly dogs, which is a good. However, some retrieved images contain different breeds or dogs in different contexts. The embeddings might capture some superficial features rather than deep semantic understanding.

2nd query : The retrieved images do not share an obvious visual similarity. The model may not be properly distinguishing texture-based, color-based, or contextual information. The model may focus too much on the shape, long and thin (query, 2 and 4) or round (2 and 5).

3rd query : There is no clear thematic or visual similarity to laboratory equipment or scientific imagery : the nearest neighbors include a ruin, a staircase, an old photo of a child, a school bus, and a bridge. The embeddings may not properly distinguish categories and are failing to cluster images based on meaningful semantics.