Project Summary

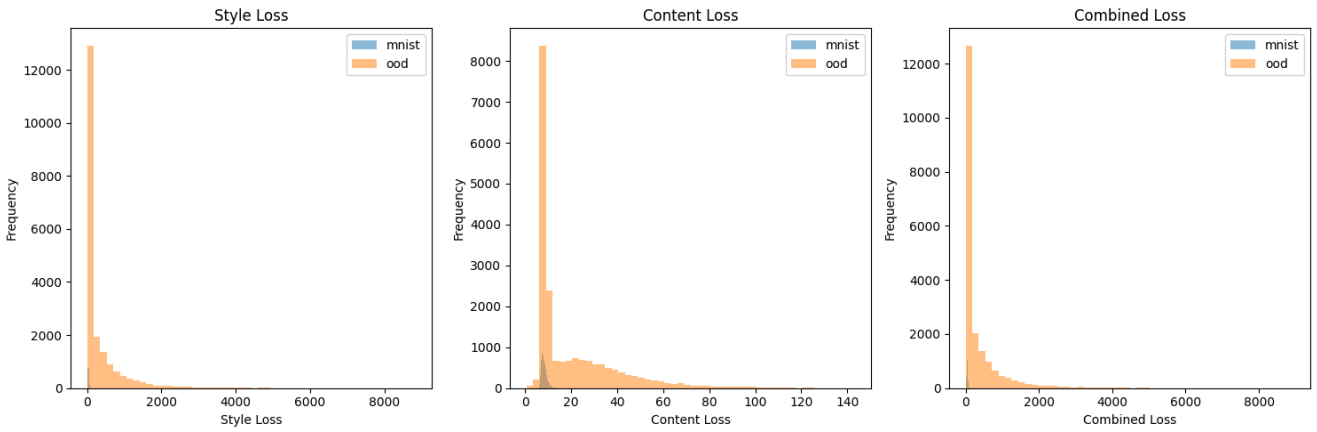
By: Yuval Reuveni, Maya Byle

previous attempts:

1. At first, we tried to implement a similar model to the method mentioned in C2AE: Class Conditioned Auto-Encoder for Open-set Recognition, but the training process needed to meet the available calculation resources. **We liked the approach of a multi-task auto-encoder and the comparison between the reconstructed and the original images,** so we tried to find a "cheaper" training process.
2. We found a threshold evaluation method that is being used in NST. The method combines two different metrics:
    a. Style loss - compares the patterns between the generated and original images by measuring the difference in their feature map correlations (represented by Gram matrices).
    b. Content loss- measures the difference between the feature representations of the generated and content images to preserve the original image's structure.

    We tried to use it to set a threshold that distinguishes between the MNIST and OOD. We used it at different stages of the reconstruction process (Content loss on the encoded vector, and Style loss on the image itself).

    The issue we had with this method was that even after many attempts, we did not manage to create a clear separation between the different distributions:



We did like the combination of two loss metrics to set a threshold to increase the amount of certainty and moved forward with this idea.

3. We also tried using a multi-task auto-encoder with the trained CNN as a classifier, we thought it would improve the model's capabilities of classifying MNIST over unknown

images. However, as we observed, it also improves the capabilities of identifying unknown images and cost computational resources. So we switched to a simpler classifier.

The Method - overview:

We chose multi-task AE that encodes the input image, and then classifies and reconstructs it. During the training, we use two loss metrics - cross entropy (for the classification task) and MSE (for the reconstruction task). The differences between the original and reconstructed images are used to identify OOD samples later on. For the OSR classification mission, we chose a threshold that combines two metrics (TBD) to increase the amount of certainty.

The model architecture:

The Multi-task Autoencoder (AE) model combines a simple classifier for classifying MNIST digits with an Autoencoder (AE) to help distinguish between in-distribution (MNIST) and out-of-distribution (OOD) samples.

First, the AE encodes the input images. Then, the AE reconstructs the encoded vector to an image (with the decoder), and classifies it into one of the known classes.

The baseline model:

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 28, 28]	1,664
Conv2d-2	[-1, 32, 14, 14]	18,464
Linear-3	[-1, 256]	8,448
Linear-4	[-1, 10]	2,570
LogSoftmax-5	[-1, 10]	0

Total params: 31,146

The AE model:

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 16, 28, 28]	160
BatchNorm2d-2	[-1, 16, 28, 28]	32
MaxPool2d-3	[-1, 16, 14, 14]	0
Conv2d-4	[-1, 32, 14, 14]	4,640
BatchNorm2d-5	[-1, 32, 14, 14]	64
MaxPool2d-6	[-1, 32, 7, 7]	0
Linear-7	[-1, 128]	200,832

Encoder-8	[-1, 128]	0
Linear-9	[-1, 1568]	202,272
ConvTranspose2d-10	[-1, 16, 7, 7]	4,624
BatchNorm2d-11	[-1, 16, 7, 7]	32
Upsample-12	[-1, 16, 14, 14]	0
ConvTranspose2d-13	[-1, 1, 14, 14]	145
Upsample-14	[-1, 1, 28, 28]	0
Decoder-15	[-1, 1, 28, 28]	0

==========

Total params: 412,801

Threshold Technique:

Our assumption: The MT AE was trained on MNIST data, so it would reconstruct the MNIST images better than OOD images and, hence will have a smaller difference between the origin and reconstructed MNIST inputs.

The model uses a thresholding technique to effectively separate in-distribution (MNIST) from out-of-distribution (OOD) samples. Two key metrics are computed: the norm loss (which measures the reconstruction error) and the difference loss (which measures the discrepancy between the original and reconstructed binary images).
Thresholds for these metrics are established based on the distribution of losses observed during training. During evaluation, if a sample's norm or difference loss exceeds these thresholds, it will be classified as OOD.

threshold for train data

```
[28] dist_norms, dist_diffs = measure_distance(model, train_loader)
     norm_threshold = np.mean(dist_norms) +  2* np.std(dist_norms)
     diff_threshold = np.mean(dist_diffs) +  2*np.std(dist_diffs)
     print(norm_threshold, diff_threshold)

     19.09636116027832 65.0808846685645
```

threshold for test ood data

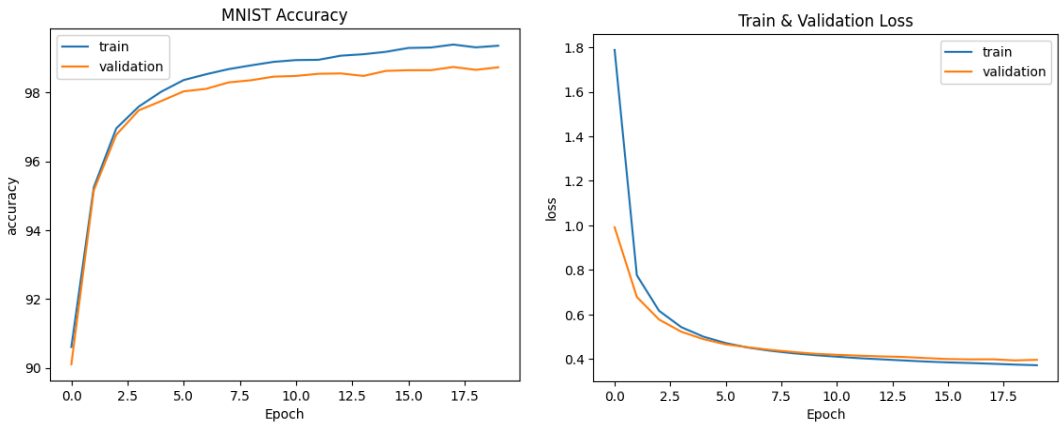
```
     normf, diff = measure_distance(model, test_ood_loader)
     norm_threshold = np.mean(normf) +  2* np.std(normf)
     diff_threshold = np.mean(diff) +  2*np.std(diff)
     print(norm_threshold, diff_threshold)

     34.1387357711792 580.345870028398
```

Hyper-Parameters:

- Batch Size: 1024 - we chose a relatively large batch size to speed up the training process.
- Number of Epochs: 20 - a balanced tradeoff between computing time and accuracy performance.
- Learning Rate: 0.0005 - after trying several parameters we observed this lr is optimal for training without diverging.
- optimizer: optim.Adam as we learned in class.
- **Activation Functions**: Relu as we learned in class.
- Number of Classes: 11.
- Norm Threshold: The threshold for determining if the reconstruction loss (norm loss) is indicative of an OOD sample. we compute it based on our MNIST train set.
- Diff Threshold: The threshold for determining if the difference loss between the original and reconstructed images indicates an OOD sample. we compute it based on our MNIST train set.

  Both norm and diff thresholds were chosen to be the sum of the mean and 2*std of their values. After trying several combinations, we chose the combination that achieved the best accuracy.

Train and Validation Results:



Baseline model Accuracy:

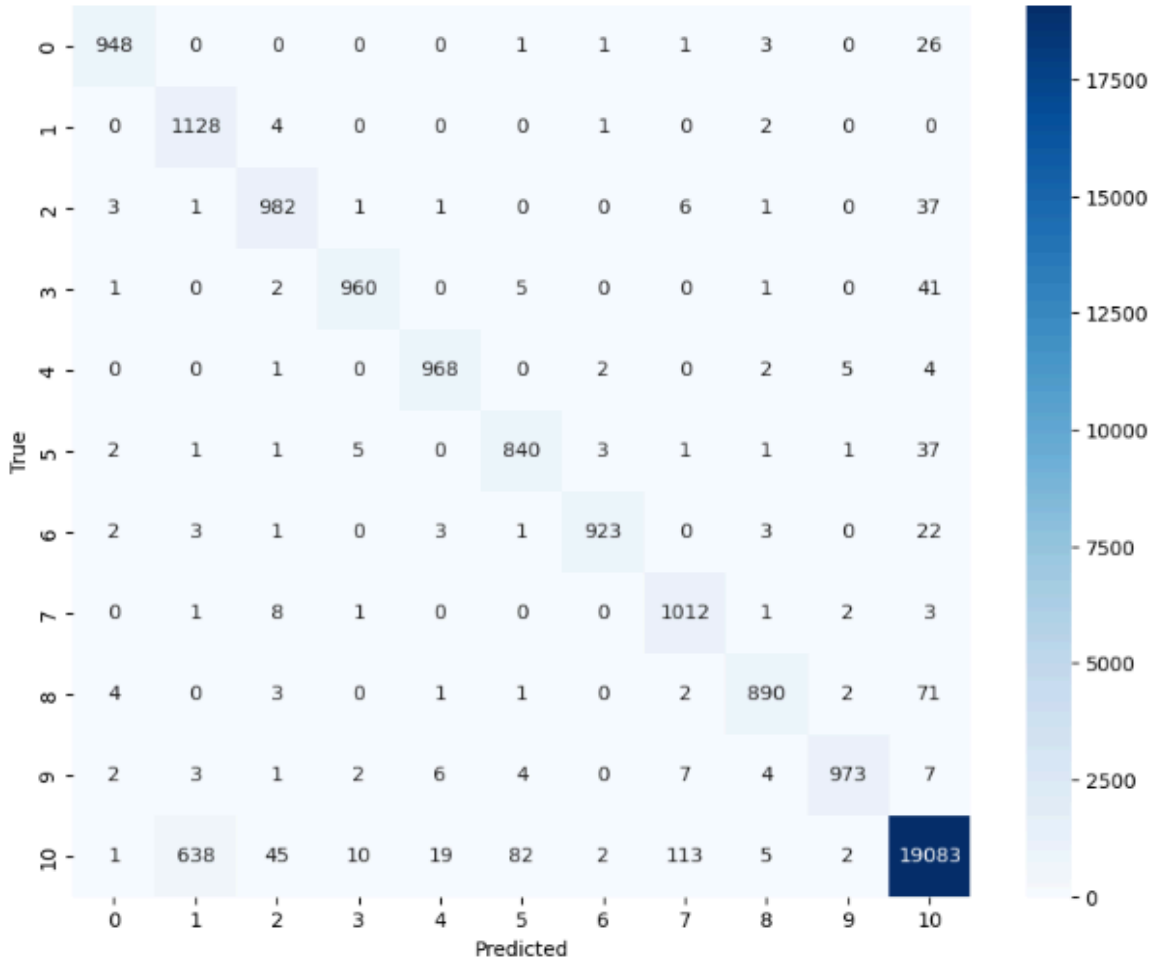
80.99%

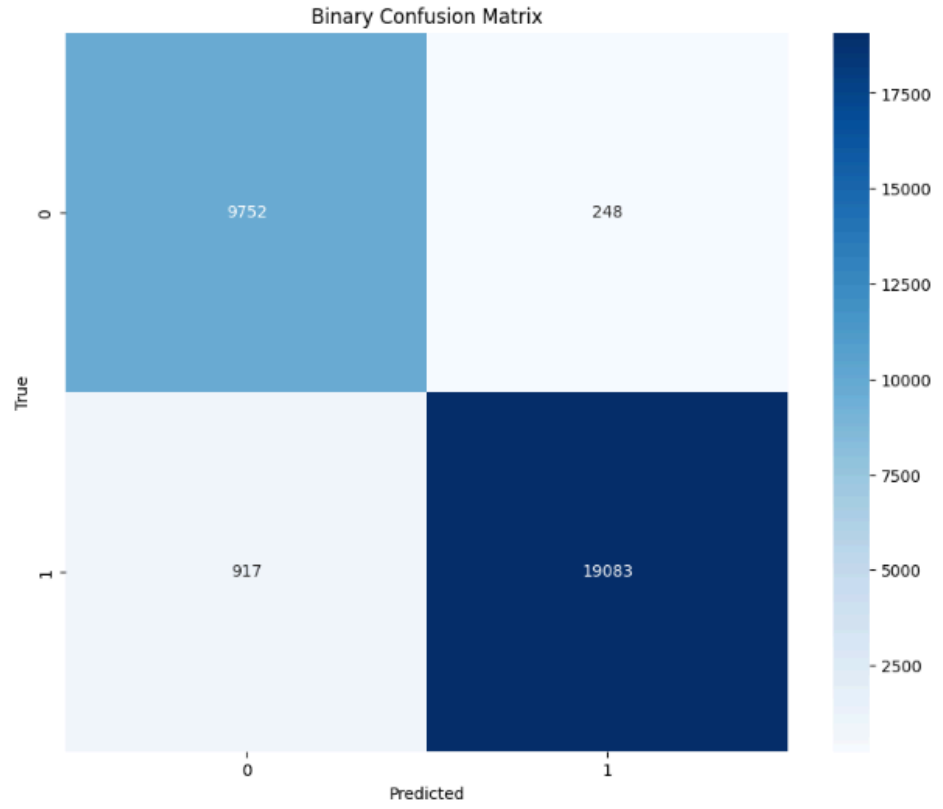
OSR results:

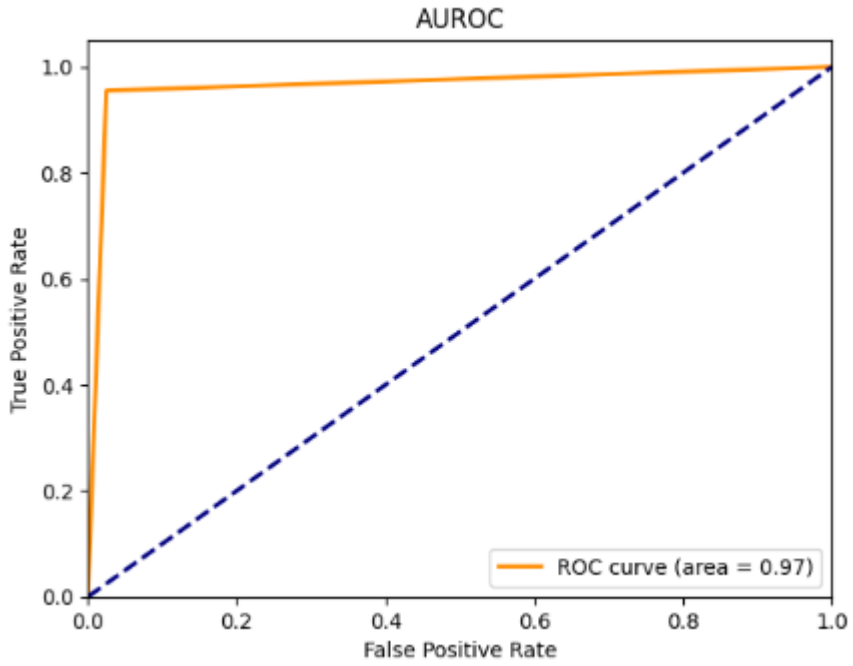
MNIST Accuracy: 96.24%

OOD Accuracy: 95.42%

Total Accuracy: 95.69%


0 948 0 0 0 0 1 1 1 3 0 26
1 0 1128 4 0 0 0 1 0 2 0 0
2 3 1 982 1 1 0 0 6 1 0 37
3 1 0 2 960 0 5 0 0 1 0 41
4 0 0 1 0 968 0 2 0 2 5 4
5 2 1 1 5 0 840 3 1 1 1 37
6 2 3 1 0 3 1 923 0 3 0 22
7 0 1 8 1 0 0 0 1012 1 2 3
8 4 0 3 0 1 1 0 2 890 2 71
9 2 3 1 2 6 4 0 7 4 973 7
10 1 638 45 10 19 82 2 113 5 2 19083
True
0 1 2 3 4 5 6 7 8 9 10
Predicted
17500
15000
12500
10000
7500
5000
2500
0


Binary Confusion Matrix
9752
248
917
19083
True
0
1
Predicted
0
1
17500
15000
12500
10000
7500
5000
2500



Style and content loss results from previous attempt:

MNIST Accuracy: 92.33%

OOD Accuracy: 88.99%

Total Accuracy: 90.66%

Strengths:

- In-distribution Data (MNIST): The model performs well on MNIST digit classification, achieving over 96% accuracy across different training durations. It excels at recognizing familiar patterns within the MNIST dataset, particularly due to CNN's strong feature extraction capabilities combined with the autoencoder.
- Out-of-distribution (OOD) Detection: With adequate training (e.g., 20 epochs), the model is effective at distinguishing OOD samples, achieving over 95% accuracy. This indicates that the model can detect significant deviations from the MNIST distribution, particularly when the OOD samples are visually distinct from MNIST digits.

Expected Performance and Limitations:

- our approach relies on a threshold that is based on the mean and std of the metrics described above. we expect that the model will struggle to classify correctly OOD images with small loss and MNIST images with large loss.

- Challenges with Complex Data: The model is designed for simple, MNIST-like data and may not perform well on complex or high-dimensional datasets, such as natural images or noisy data. In these cases, the autoencoder might misclassify OOD samples due to difficulties in reconstruction.
- Best Use Cases: The model is expected to excel on datasets similar to MNIST, where in-distribution data is straightforward and OOD samples are different.