ADL - Project Summary

Naor Guetta Tomer Varsanno

March 13, 2025

1 Final Method and Rationale

1.1 Our approach

Our final approach for this project is out-of-distribution (OOD) detection based on reconstruction error and distance from latent feature clustering. To achieve this, we implemented a wrapper class that follows a two-step verification process:

- It first checks whether the reconstruction error exceeds a predefined threshold.
- If the error is within the acceptable range, it then verifies whether the latent feature distance meets the clustering distance thresholds. Any sample that does not satisfy these conditions is considered OOD.

This method is built upon a multi-class AutoEncoder designed to learn compact latent representations while simultaneously performing classification. The model consists of three key components:

- Encoder – extracts latent features from input images.
- Decoder – reconstructs the input to ensure the learned representations remain meaningful.
- Classifier – uses the latent features to predict class labels.

By combining reconstruction-based anomaly detection with latent feature clustering, our approach enhances the reliability of OOD detection.

The rationale behind this approach lies in the dual functionality of the autoencoder: reconstructing known-digit patterns while identifying deviations in unknown samples. This combination helps the model learn robust representations that generalize well, making it capable of identifying out-of-distribution samples.

1.2 Previous attempts

Our initial approach involved ODIN [1](Out-of-Distribution Detector for Neural Networks), which uses input perturbations and temperature scaling to improve the detection of unknown inputs. Although ODIN improved performance in some cases, it requires manual threshold adjustment after training, using ROC analysis to determine the optimal value. Since this assignment requires reproducible results, a fixed threshold may not yield consistent performance across different runs. Although ODIN is faster than our autoencoder-based method, which makes it suitable for tasks with large datasets, its reliance on manual tuning and sensitivity to visually similar digits led us to prefer the autoencoder approach for a more reliable OSR.

Another approach we tested involved using a single autoencoder and a separate classifier neural network. In this method, we first calculate the reconstruction error of the input using the autoencoder. If the reconstruction error exceeded a predefined threshold, the sample was classified as OOD. Otherwise, the input was passed to the classifier neural network to predict the class label. While this approach worked reasonably well, it only relies on reconstruction error, and we feared that it would not work on similar data sets, i.e. letters. As such, we opted for a more robust system.

2 Hyper-parameters

Our method involves several key hyper-parameters, including:

- Learning Rate: Controls the step size for updating model weights during training.
- Batch Size: Determines the number of samples processed before updating model parameters.
- Alpha (Reconstruction Weight): Balances the reconstruction loss in the autoencoder.
- Beta (Contrastive Loss Weight): Controls the contribution of contrastive loss, which encourages distinct latent representations for different classes.
- Weight Decay: Regularization parameter in AdamW that prevents overfitting by penalizing large weights.
- Reconstruction Multiplier: Scales the standard deviation of reconstruction errors, effectively determining how lenient or strict the threshold is for identifying outliers based on how well a sample is reconstructed.
- Latent Multiplier: Scales the standard deviation of distances from data points to their class centroid in the latent space. This is used to compute per-class cluster thresholds.

To select the optimal hyper-parameters, we used Optuna, a framework for automated hyper-parameter optimization. Optuna allowed us to efficiently explore the search space by running multiple trials to identify promising configurations. Each trial trained the model with different hyper-parameter values and evaluated performance using validation accuracy and OOD detection metrics. The final configuration was chosen based on the best trade-off between classification accuracy on known digits and detection performance for unknown samples.

3 Results and Performance Evaluation

3.1 Training Process: Loss and Accuracy Over Epochs

Figures 1 and 2 show the training loss and classification accuracy over the course of training. The loss graph illustrates the convergence of both reconstruction and classification losses, indicating that the model successfully learned to minimize both errors. The accuracy graph shows steady improvement, eventually stabilizing as the model approached its optimal performance.

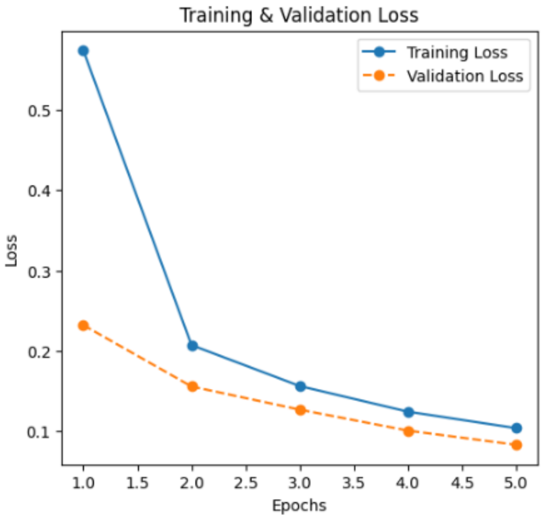


Figure 1: Training loss over epochs.

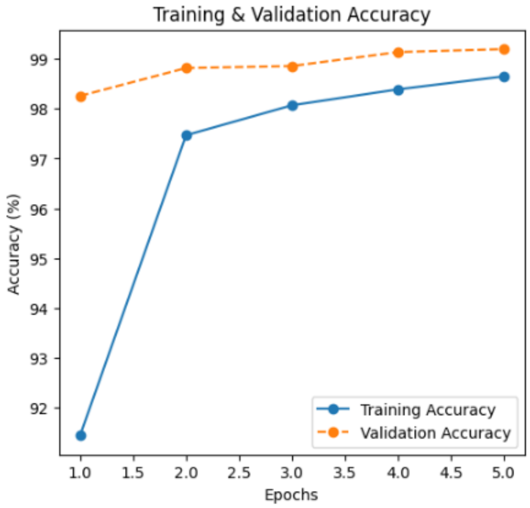
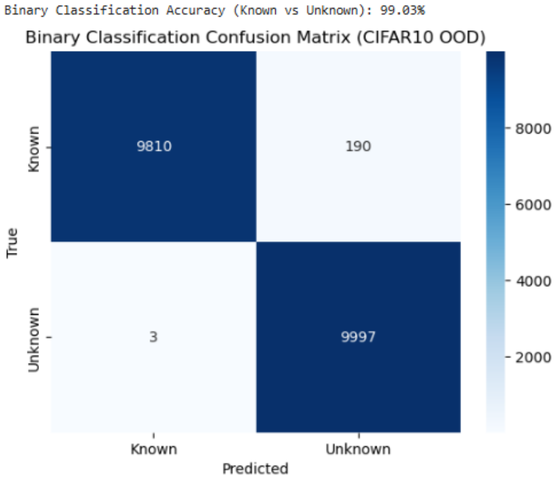


Figure 2: Classification accuracy over epochs.

3.2 Confusion Matrices (CM)

The confusion matrix in Figure 3 shows the performance of our model in distinguishing between in-distribution and out-of-distribution samples. Meanwhile, the confusion matrix in Figure 4 highlights the model's ability to accurately classify known digits while minimizing misclassification of unknown samples.

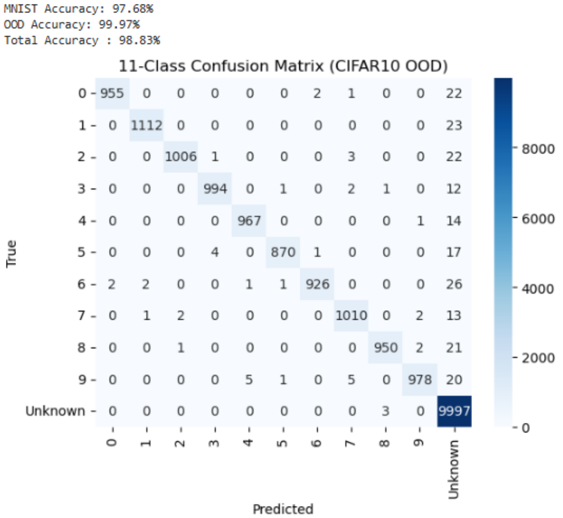


```
Binary Classification Accuracy (Known vs Unknown): 99.03%
```

Binary Classification Confusion Matrix (CIFAR10 OOD)

True \ Predicted	Known	Unknown
Known	9810	190
Unknown	3	9997

Figure 3: OOD and In-Distribution detection.



```
MNIST Accuracy: 97.68%
OOD Accuracy: 99.97%
Total Accuracy : 98.83%
```

11-Class Confusion Matrix (CIFAR10 OOD)

True \ Predicted	0	1	2	3	4	5	6	7	8	9	Unknown
0	955	0	0	0	0	0	2	1	0	0	22
1	0	1112	0	0	0	0	0	0	0	0	23
2	0	0	1006	1	0	0	0	3	0	0	22
3	0	0	0	994	0	1	0	2	1	0	12
4	0	0	0	0	967	0	0	0	0	1	14
5	0	0	0	4	0	870	1	0	0	0	17
6	2	2	0	0	1	1	926	0	0	0	26
7	0	1	2	0	0	0	0	1010	0	2	13
8	0	0	1	0	0	0	0	0	950	2	21
9	0	0	0	0	5	1	0	5	0	978	20
Unknown	0	0	0	0	0	0	0	0	3	0	9997

Figure 4: 11-class classification.

3.3 t-SNE Visualization of Latent Space

The t-SNE plot in Figure 5 illustrates the latent space representations of both known and unknown samples. Distinct clusters for each known digit indicate that the model learned well-separated latent features.The clear separation in the latent space is critical for reliable OOD detection and reflects the effectiveness of our training approach.

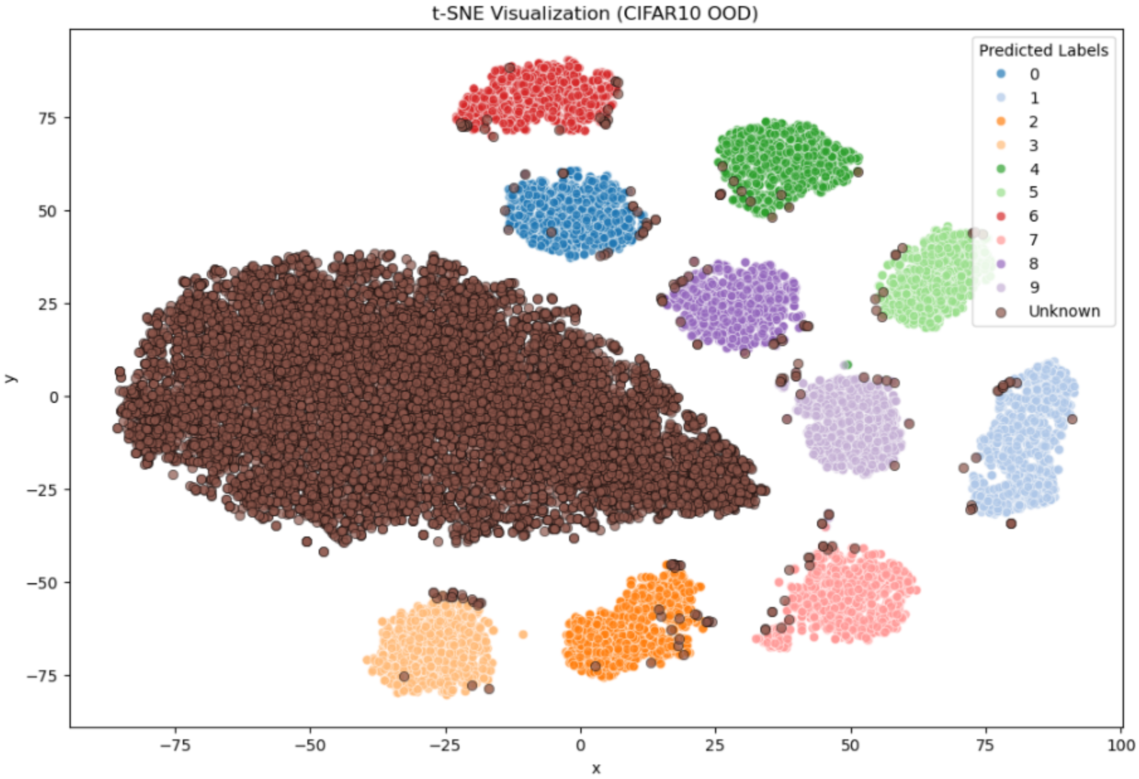


Figure 5: t-SNE visualization of latent space representations.

4 Limitation

While our approach demonstrates strong performance in recognizing both known and unknown samples, it has several limitations. The model's computational complexity arises from the need to calculate both reconstruction errors and latent distances, which can be time-consuming, especially when applied to large datasets. Additionally, the model's performance is sensitive to the selected reconstruction and latent distance thresholds. Although these thresholds were optimized using Optuna, they may require further adjustments when applied to different datasets.

Our approach performs well on structured datasets with clear class boundaries, This allows the autoencoder to learn consistent patterns, improving its ability to detect unknown inputs. However, its performance may degrade on highly unstructured data with significant intra-class variability, as this can increase reconstruction errors even for known samples. Due to its computational overhead, the model is less suitable for large-scale datasets. Despite these limitations, our method remains reliable for Open Set Recognition tasks with clear class distinctions and manageable dataset sizes, making it well-suited for applications like handwritten digit classification.

References

[1] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2018. Available at https://arxiv.org/abs/1706.02690.