Project Summary (Section 5.8)

Lee Ben Gigi (208312686)
Ron Gurevich (318772456)
Omer Ben Arie (211602842)

1. Method Description

Our approach to the Open Set Recognition (OSR) problem combines embedding space analysis with triplet loss learning, probability thresholds, and density-based clustering. The core philosophy behind our method is that samples from known classes (MNIST digits) should form distinct clusters in the embedding space, while unknown samples would likely fall outside these clusters.

We implemented a multi-stage classification pipeline:

1. **Feature Extraction with Combined Losses**
   We trained a CNN architecture on MNIST using a combination of cross-entropy (CE) and triplet loss to learn discriminative embeddings. While the triplet loss explicitly encourages samples from the same class to cluster together and pushes samples from different classes apart, leading to well-formed clusters in the embedding space, the cross-entropy loss plays a complementary role by maximizing the predicted probabilities of the correct classes. This dual-objective approach ensures that the learned features are both highly discriminative and optimized for classification accuracy**.**

2. **DBSCAN Clustering**
   We applied DBSCAN (Density-Based Spatial Clustering of Applications with Noise) to the embedding space of each class. In our experiments, the chosen hyperparameters typically resulted in the formation of two clusters per class. This approach enables us to capture intra-class variability by modeling the class distribution with two distinct clusters—each represented by its centroid and radius—rather than assuming a single prototype per class. Such a configuration allows our system to better handle complex and multi-modal data distributions in the open-set recognition scenario.

3. **Two-Stage Classification**
   Our classification strategy enhances open-set recognition through two sequential stages:

   Stage One – Softmax Probability Check:
   After extracting the sample's embedding, the model computes logits for 10 known classes and converts these into probabilities using softmax. The class with the highest probability is considered the predicted class. If this maximum probability falls below a predefined threshold, the sample is immediately classified as "Unknown." This is implemented by adjusting the logits suppressing the predicted class's score and boosting the "Unknown" label so that the subsequent softmax strongly favors "Unknown."

Stage Two – Cluster Distance Check:

If the sample passes the softmax threshold, the DBSCAN clustering results for the predicted class are retrieved. (In our experiments, each class generally yields two clusters.)

- If no clusters are available for the predicted class, the sample is classified as "Unknown."
- If clusters do exist, the Euclidean distance between the sample's embedding and the centroid of each cluster within the predicted class is computed. The minimum distance among these is selected and compared against the corresponding cluster's radius (scaled by a factor, dist_factor).
- If the minimum distance exceeds the scaled radius, it implies that the sample does not conform well to any of the clusters for the predicted class, and thus, it is reclassified as "Unknown."
- Otherwise, the sample retains its predicted class label.

This two-stage approach allows us to effectively distinguish ambiguous in-distribution samples (such as poorly written but valid MNIST digits that may have low confidence yet still fall within cluster boundaries) from truly out-of-distribution samples that might exhibit deceptively high confidence scores but do not match the known spatial distribution patterns.

2. Hyperparameters and Configuration

Our method involves several key hyperparameters that we tuned carefully to balance the accurate classification of in-distribution samples (MNIST) with the effective detection of out-of-distribution (OOD) samples. The main hyperparameters and their configurations are as follows:

1. **Embedding Dimension:**
   - **Setting:** 64
   - **Rationale:** This dimension provides sufficient expressive power for feature representation while keeping computational requirements manageable.
2. **Triplet Loss Parameters:**
   - **Margin:** 1.0
     - *Purpose:* Ensures that the distance between an anchor and a negative sample is at least 1.0 greater than the distance between the anchor and a positive sample. This encourages samples of the same class to cluster tightly while pushing apart samples of different classes.
   - **Alpha (Weight):** 1.0
     - *Purpose:* Balances the contribution of the triplet loss relative to the cross-entropy loss during training, ensuring that both objectives contribute equally.
3. **DBSCAN Parameters:**
   DBSCAN is used to identify natural clusters within each class's embedding space. The key parameters are:
   - **eps:**
     - *Definition:* The maximum distance between two samples for them to be considered neighbors.
     - *Tuning:* We performed a grid search over values [1.6, 1.8, 2, 2.2].
   - **min_samples:**
     - *Definition:* The minimum number of samples required to form a cluster.
     - *Tuning:* Values [5, 7, 9] were evaluated. In our experiments, these parameters typically led to the formation of two clusters per class.
4. **Classification Thresholds:**
   The classification process involves a two-stage decision:
   - **Probability Threshold:**
     - *Definition:* The minimum softmax confidence required for the model to trust its predicted class before proceeding to the spatial (cluster-based) check.
     - *Tuning:* We experimented with values [0.6, 0.8].
   - **Distance Factor:**
     - *Definition:* A scaling factor applied to the cluster radius when checking if a sample’s embedding is sufficiently close to the nearest cluster centroid.

    - *Tuning:* Factors [1.0, 1.2] were explored.

A grid search was conducted across all parameter combinations (4 eps values × 3 min_samples values × 2 probability thresholds × 2 distance factors = 48 combinations) to optimize for two primary objectives:

- **Maintaining High Accuracy on MNIST:** Minimizing false "Unknown" classifications to ensure that valid MNIST samples are correctly recognized.
- **Effective OOD Detection:** Minimizing false "Known" classifications by correctly identifying out-of-distribution samples as "Unknown."

Achieving the right balance was critical. For example, setting the distance factor too low would cause many valid MNIST samples to be misclassified as "Unknown," while setting it too high could allow truly OOD samples to be incorrectly labeled as known.

Additionally, The code updates the best hyperparameter configuration only if:

1. Total accuracy improves.
2. Out-of-distribution (OOD) accuracy is above 80%.
3. The total accuracy gain outweighs any OOD accuracy loss.

This ensures the model improves overall performance while maintaining good generalization to unseen data. By requiring high OOD accuracy and prioritizing total accuracy improvement over potential OOD accuracy loss, it prevents overfitting to known data. This favours a balanced configuration with high accuracy on both known and unknown data, leading to a more robust and reliable model.

Best Configuration:

Based on our grid search, the optimal configuration was found to be:

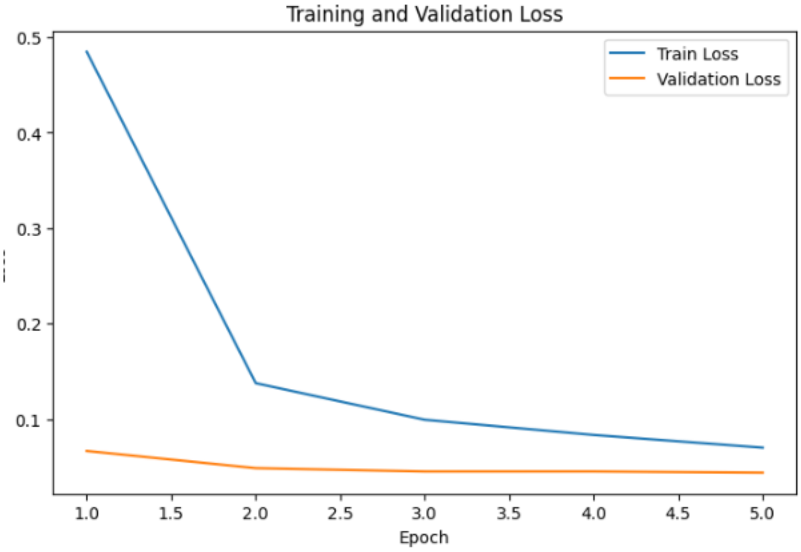
- **eps = 2.2**
- **min_samples = 9**
- **Probability Threshold = 0.6**
- **Distance Factor = 1**

This configuration achieved a validation accuracy of 95.58 %.

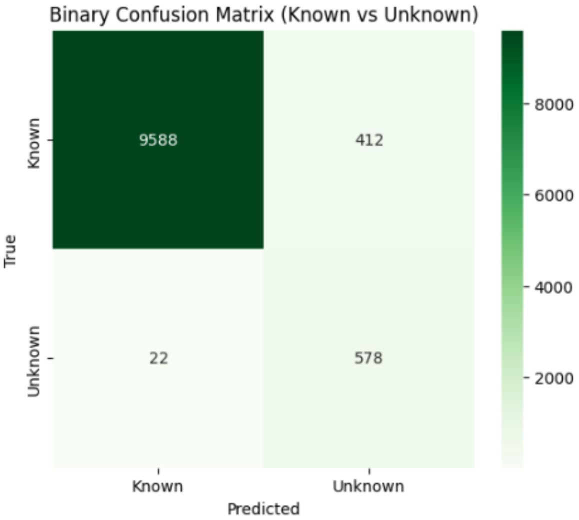
3. Results Analysis

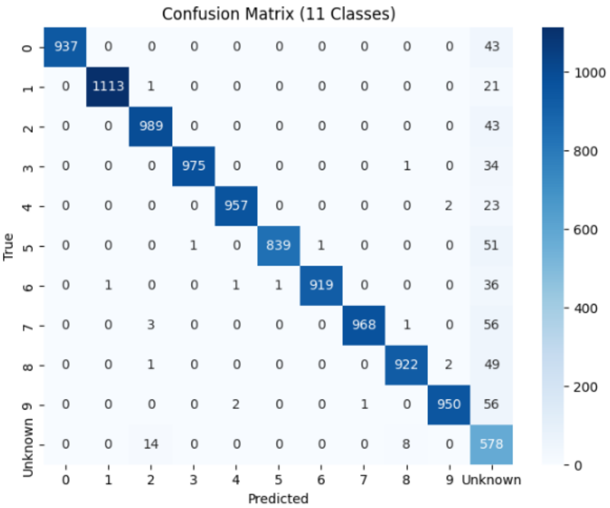
Our experiments evaluated the model on MNIST (in-distribution) and FashionMNIST (out-of-distribution). The following key observations emerged:

- Training and validation loss:



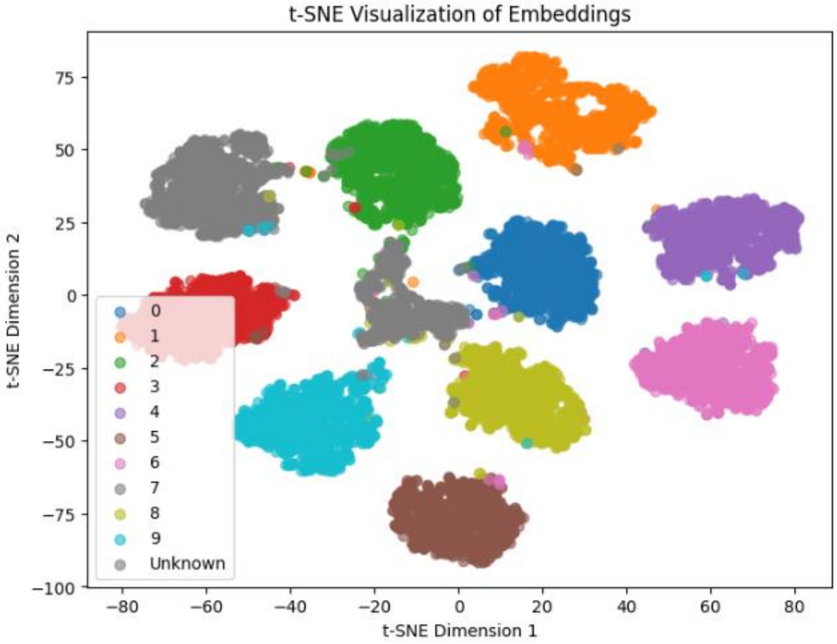
- **Confusion Matrix Analysis**
  Asymmetric Error Pattern: The binary confusion matrix (Known vs. Unknown) and the 11-class confusion matrix reveal that, initially, our model was more prone to misclassifying MNIST digits as "Unknown" than to misclassify FashionMNIST samples as "Known."
  Threshold Tuning: This finding was crucial in guiding the tuning of the probability and distance thresholds, ultimately helping us reduce false "Unknown" classifications without substantially increasing false "Known" errors.





- **Distance Distribution**

  MNIST vs. FashionMNIST Clusters: Analyzing the distribution of distances between each sample and its corresponding cluster centroid showed that MNIST digits generally clustered more tightly around their centroids compared to FashionMNIST samples.
  Overlap for Ambiguous Digits: Despite the tighter clustering, there was still partial overlap—particularly for poorly written or ambiguous digits—emphasizing the importance of a distance-based cutoff.



- **Hyperparameter Tuning Impact**
  Trade-off Between Accuracy and OOD Detection: Varying the probability threshold and distance factor illustrated the balance needed between maintaining high accuracy on MNIST and effectively recognizing FashionMNIST as OOD.
  Optimal Operating Point: Through systematic grid search, we identified a threshold configuration that minimized false "Unknown" classifications (maintaining MNIST accuracy) while still rejecting truly OOD FashionMNIST samples.
- **Clustering Effectiveness**
  Handling Intra-Class Variance: DBSCAN helped the model capture complex intra-class patterns, especially for digits with high variability (e.g., '1' and '7').

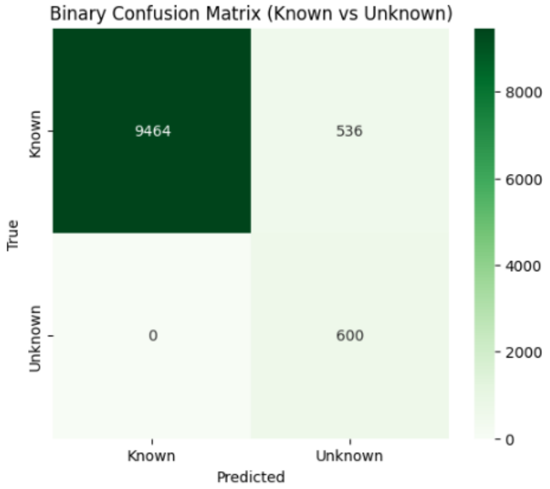
Distinguishing OOD Samples: By comparing embedding distances to multiple cluster centroids, the model could differentiate between atypical MNIST digits and genuinely out-of-distribution FashionMNIST samples.
Reduced False "Unknown" Rates: The distance-based approach, combined with DBSCAN, significantly decreased the likelihood of labeling valid MNIST samples as "Unknown," while only minimally affecting the detection of FashionMNIST as OOD.

Overall, the final configuration provided a strong balance between correctly classifying in-distribution digits and rejecting OOD samples, as evidenced by the confusion matrices and the t-SNE visualization of the learned embeddings.

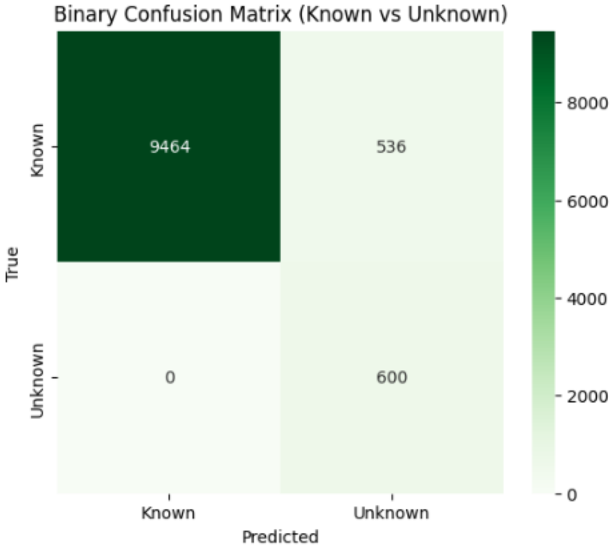
Final OSR Model and Extended Results

In our final OSR model, the optimal hyperparameter configuration was derived using FashionMNIST as the out-of-distribution (OOD) dataset. This model was subsequently evaluated on additional OOD datasets to assess its generalization capabilities. The results are summarized below:

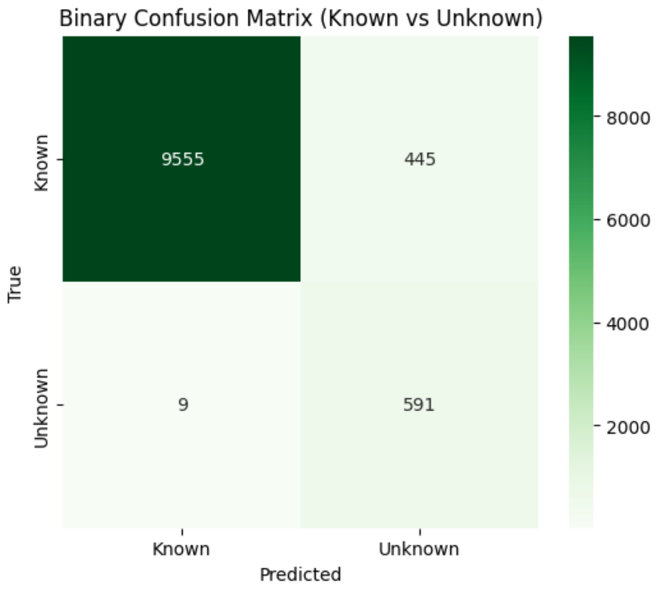
- **CIFAR10:**
  - Accuracy on MNIST: 95.32%
  - Accuracy on OOD (CIFAR10): 100.00%
  - Total Accuracy: 95.58%



- **SVHN:**
  - Accuracy on MNIST: 95.32%
  - Accuracy on OOD (SVHN): 100.00%
  - Total Accuracy: 95.58%



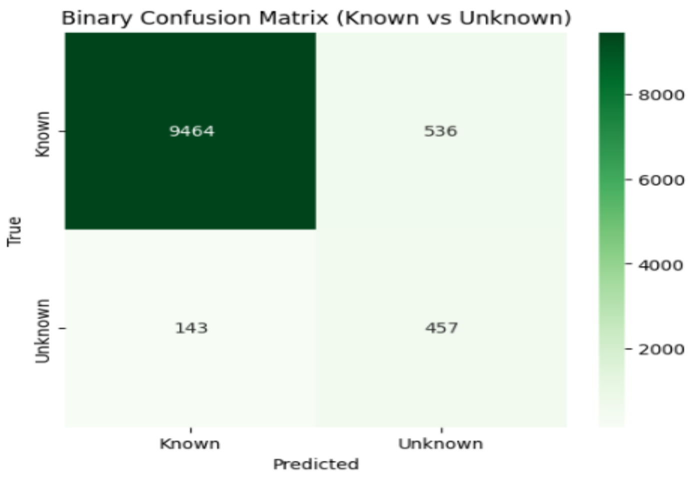
- **USPS:**
  - Accuracy on MNIST: 95.32%
  - Accuracy on OOD (USPS): 98.50%
  - Total Accuracy: 95.5%



- **EMNIST:**
  - Accuracy on MNIST: 95.32%
  - Accuracy on OOD (EMNIST): 81.50%
  - Total Accuracy: 94.54%

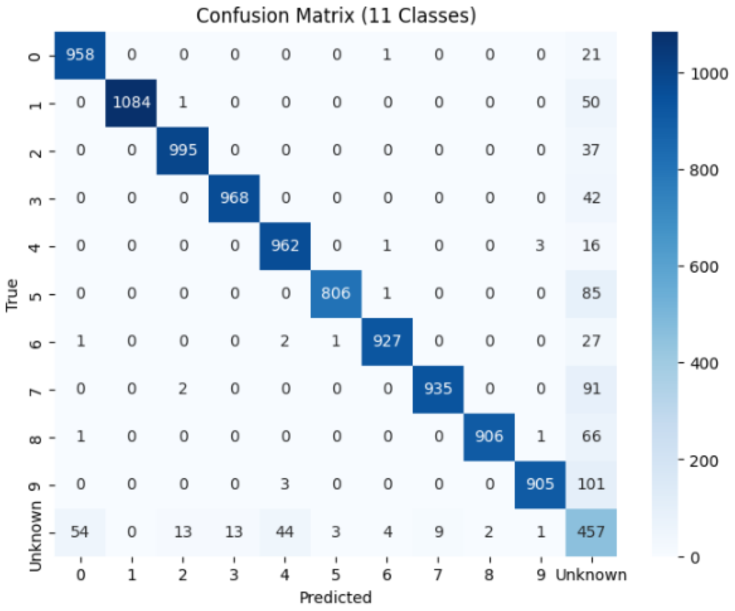
  Although our final OSR model was tuned using FashionMNIST as the out-of-distribution (OOD) dataset, we additionally evaluated it on EMNIST to assess its performance on a more similar handwriting-based dataset. The figures above include:

  1. **Binary Confusion Matrix (Known vs. Unknown)**
     - Shows that while most MNIST digits are correctly recognized as "Known," a noticeable number of EMNIST samples are misclassified as "Known" (and vice versa).
     - This reflects the higher similarity between EMNIST letters and MNIST digits compared to other OOD datasets like CIFAR10 or SVHN.



2. **11-Class Confusion Matrix (0–9 + Unknown)**

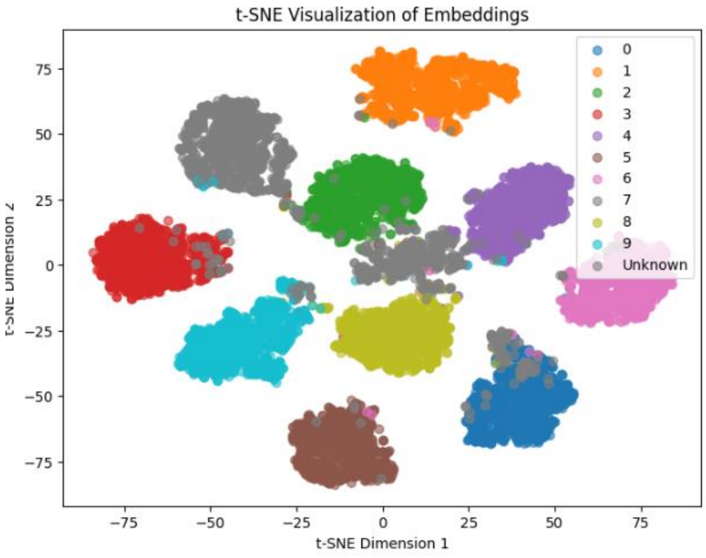
- Illustrates how the model distributes predictions across the 10 digit classes plus the “Unknown” label.
- A key observation from the matrix is that many misclassifications involve the digit ‘0,’ which often gets confused with the letter ‘O’ in EMNIST. This overlap is intuitively reasonable, given the visual similarity between these characters.



True \ Predicted	0	1	2	3	4	5	6	7	8	9	Unknown
0	958	0	0	0	0	0	1	0	0	0	21
1	0	1084	1	0	0	0	0	0	0	0	50
2	0	0	995	0	0	0	0	0	0	0	37
3	0	0	0	968	0	0	0	0	0	0	42
4	0	0	0	0	962	0	1	0	0	3	16
5	0	0	0	0	0	806	1	0	0	0	85
6	1	0	0	0	2	1	927	0	0	0	27
7	0	0	2	0	0	0	0	935	0	0	91
8	1	0	0	0	0	0	0	0	906	1	66
9	0	0	0	0	3	0	0	0	0	905	101
Unknown	54	0	13	13	44	3	4	9	2	1	457

3. **t-SNE Visualization**

- Depicts the learned embedding space, where each color represents a different digit class, and gray (or the last color) represents “Unknown.”
- While the majority of MNIST points cluster distinctly, some EMNIST samples fall within or near these clusters, explaining the classification errors.



4. Limitations

Our approach should perform well on OOD data that is significantly different from MNIST digits (like natural images or text) but might struggle with more similar distributions (like handwritten letters or other handwritten symbols) that could occupy similar regions in the embedding space.

Despite these limitations, our method achieves a good balance between maintaining high accuracy on known classes while effectively identifying unknown samples, all without violating the constraint of not training on out-of-distribution data, and while ensuring that the training and tuning time remains within the defined limits.