**ADL final project - project summary**

**1. Method and reason:**

The project involves building an Open Set Recognition (OSR) model using a convolutional multi task autoencoder architecture. The model is designed to classify known classes while detecting unknown samples by leveraging reconstruction error. The architecture comprises:

* Encoder: Extracts features from input images using convolutional layers.
* Classifier: Predicts class probabilities using a linear layer followed by LogSoftmax.
* Decoder: Reconstructs the input from encoded features, facilitating the reconstruction error calculation.

The baseline model is adapted into an OSR model by introducing a reconstruction threshold. This threshold determines whether a sample is from an unknown class based on the reconstruction error. During evaluation, samples exceeding the threshold are assigned to the "unknown" class.

* **Previous Attempts:**
* Hyper-parameter Tuning:  
  When determining the reconstruction threshold, various percentile values were tested. A lower percentile increased false positives, causing more known samples to be misclassified as unknown. Conversely, a higher percentile led to more false negatives, where unknown samples were incorrectly labeled as known. The final percentile was chosen to balance this trade-off, as observed in the confusion matrix.
* Data Augmentation:  
  Initially, the model exhibited overfitting, with the training accuracy reaching 100% and minimal loss, while the validation accuracy remained approximately 3% lower which indicated lack of generalization.

To improve robustness, data augmentation techniques were applied, specifically using rotation transformations. Affine transformations were considered but ultimately dropped due to their negative impact on runtime, the extended epoch durations resulted in lower accuracy within the set runtime limitations.

By focusing on rotation-only augmentation, the model achieved better generalization and improved performance on the test set.

**2. Hyper-parameters and Configuration:**

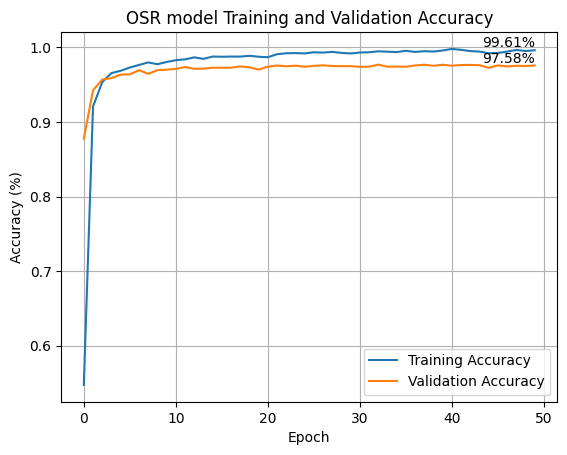
Key hyper-parameters include:

* Learning Rate of 0.1: Set during the optimizer configuration, A lower learning rate stabilizes training.
* Batch Size: Defined by the DataLoader configuration. A batch size of 512 was chosen to balance memory usage and gradient stability.
* Number of Epochs: Set to 50 to allow sufficient learning regarding runtime limitation.
* Reconstruction Threshold: Determined using the 98.5th percentile of reconstruction errors on the validation set. This threshold provides a balance between false positives and false negatives for unknown samples.

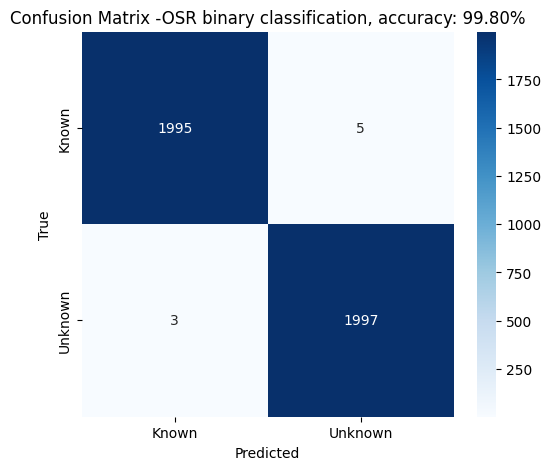
These parameters were selected through experimentation, focusing on achieving high classification accuracy while minimizing misclassification of unknown samples.

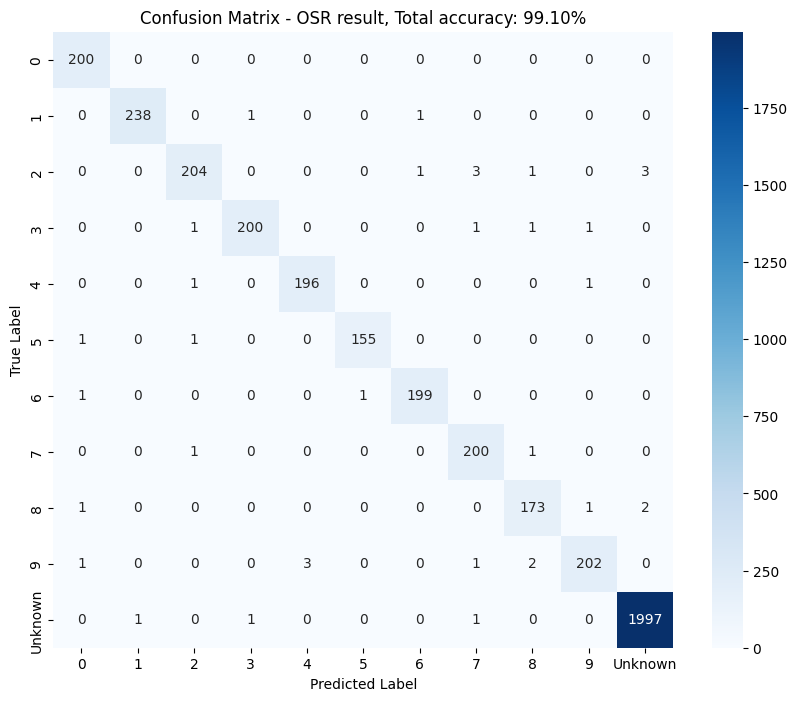
**3. Figures and Performance Indicators:**

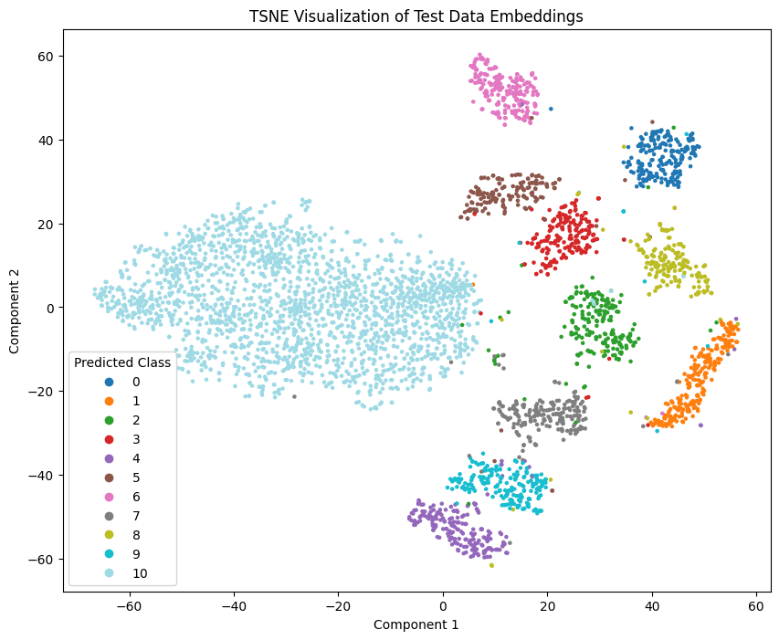
* Training & Validation Loss:
* Accuracy Curves



* Confusion matrixes:





* Tsne visualization:

Performance is evaluated using:

* Confusion matrixes: the binary classification matrix indicates how good the model label known and unknown classes, without consideration of the actual labels of the known classes (which helped us determine the threshold percentile).  
  The OSR result matrix indicates how the model classify the labels, and where is his mistakes and why.
* Unknown Detection Rate: The proportion of unknown samples correctly identified by exceeding the reconstruction threshold.
* Tsne visualization: The t-SNE visualization provides insights into the performance of the model by showing how well it separates different classes in the lower-dimensional space, which indicates that the model has learned to effectively distinguish between them.

**4. Limitations:**

The limitations of our OSR model are:

* Data Dependency: The model relies heavily on reconstruction error, which might not generalize well to all types of data distributions.
* Threshold Sensitivity: The choice of percentile for the reconstruction threshold directly impacts the balance between false positives and false negatives.

The model is expected to perform well on:

* Simple Image Data: Such as MNIST, where the reconstruction error is meaningful.
* Low Complexity Backgrounds: Where the autoencoder can accurately reconstruct known samples.
* High-quality datasets as OOD: High-quality datasets that are inherently more challenging to reconstruct, the method may perform well due to increased reconstruction errors across unknown samples.
* Low- quality datasets as main dataset: The approach demonstrates strong performance on lower-quality datasets, such as the dataset used in our experiments, where the reconstruction loss more effectively differentiates between in-distribution and out-of-distribution samples.

However, performance may degrade with:

* High Variability Data: Complex natural images where reconstruction error alone may not effectively distinguish unknown samples.
* Noisy Data: Where the autoencoder might fail to reconstruct even known samples accurately, leading to false positives for unknowns.
* Easier reconstruct Data sets: Datasets that are easier to reconstruct than MNIST, the lower percentile of reconstruction loss may not reliably correspond to unknown samples, potentially leading to misclassification.
* High-quality datasets as main data set: The approach will not demonstrate strong performance on data sets where the reconstruction error on the main data set is relatively small and effectively differentiates between in-distribution and out-of-distribution samples.
* Low-quality datasets as OOD: The approach will not demonstrate strong performance on Low-quality datasets as OOD because the reconstruction error should be quite small and it will be hard to differentiate between in-distribution and out-of-distribution samples.