A Multimodal Architecture : Seeing and Hearing Digits

Priyadharshini Ramesh Kumar
Computer Science Department
Texas A&M University
College Station, TX-77845
priyadharshini_r@tamu.edu

Abstract

This work explores the effectiveness of multimodal deep learning for classifying handwritten digits and audio MNIST dataset. We extend the traditional image-based MNIST by incorporating an audio channel where each digit is sonically represented. The model utilizes separate convolutional neural networks (CNNs) for both image and audio data, followed by a fusion layer to combine the extracted features. The final classification layer predicts the digit class. The experiments achieved an accuracy of approx 99% . This demonstrates the potential of multimodal deep learning in leveraging complementary information from different modalities to enhance classification performance on the MNIST dataset.

1 Introduction

In the world of machine learning, identification of handwritten digits has represented a fundamental challenge upon which methods for image classification can be evaluated. The MNIST dataset is made up of 60,000 such digits, and it is now widely used as a benchmark for comparing different algorithms. Traditionally, only visual information contained in the images has been taken into account by MNIST classifiers. Conversely, human brains process information through many sensors like eyes and ears that leads to understanding of things around us. This makes us wonder if we can improve the accuracy of recognizing handwritten numbers by using additional sensory data other than vision?

Multimodal deep learning offers benefits beyond potentially higher precision rates. Adding more inputs from different senses could make the model robust against noise and variations within any one modality. For instance, a digit which is partly blocked or blurred might be easily classified when accompanied with its corresponding auditory representation. In addition, learning multimodally can reveal intrinsic relationships among various modalities. In this case, between sound and visual shape of certain digits may be found by the model while dealing with hand written numbers. Thus future trends in machine

2 Your Method

This section explores the design and operation of a multi-modal deep learning model designed specifically for the categorization of digits. This novel method expands the model's comprehension of the input data and improves classification accuracy by integrating both image and audio data.The basic structure of the model is the utilization of Convolitional Neural Networks to classify both image and audio data separately and then fusing the output to get the final label.

2.1 Data Preprocessing

Here is where the data can be found on the Kaggle page of the course. The dataset is a multimodal MNIST dataset, which means it contains images of handwritten digits as well as audio files of spoken digits, paired with their corresponding label (0-9).

Using .npy files to store and manage NumPy arrays in Python is very useful; they are efficient for data manipulation because they simplify data storage during operations. Since these binary files save space compared to text-based formats like CSV, they offer a neat way of storing arrays when working with large datasets that have heavy disk space requirements. Additionally, it takes almost no time at all to load them! This means we can access array data quickly and directly — speeding up computations significantly. Furthermore,.npy files keep certain properties intact such as shape and dtype; this ensures that our saved data matches what we expect when loading back into memory later on without losing any precision or information.

Also known as one-hot encoding in machine learning applications, categorical variable preprocessing technique makes it easier to use and understand these types of variables by models. It represents each category with a binary number while keeping its semantic meaning so that algorithms can differentiate between different categories reliably.

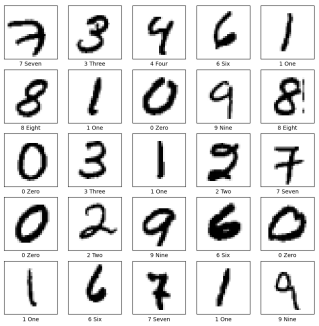


Figure 1: Image Data

In keeping with the data preprocessing, we use image representations to see the handwritten digits in Figure 1. The image is a grid with a single handwritten digit in each square cell. These numbers have accompanying text-formatted classes provided next to them. Before putting the data into a machine learning model, we can use this visualization to evaluate the image quality.

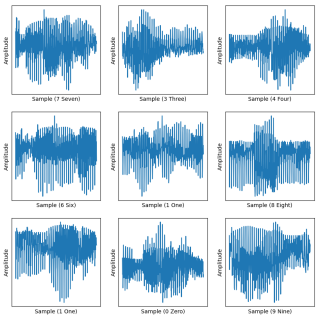


Figure 2: Audio Data

Building on from before, nine spectrograms—visual depictions of sound waves—are shown in Figure 2. With amplitude on the vertical axis and time on the horizontal axis (s), each spectrogram represents a unique sound. Stronger colors correspond to louder noises.

2.2 Model Design

Convolutional neural networks (CNNs) are used for image processing, and one-dimensional convolutional neural networks (CNNs with 1D convolutions) are used for audio processing in this

multi-modal deep learning architecture. The reason this model was chosen is that it can handle multi-modal data (pictures and audio), which makes it possible to extract characteristics from both modalities and combine them for classification. Since the MNIST dataset consists of relatively simple data, a complex model architecture wasn't necessary. This choice strikes a balance between effectiveness and computational efficiency.

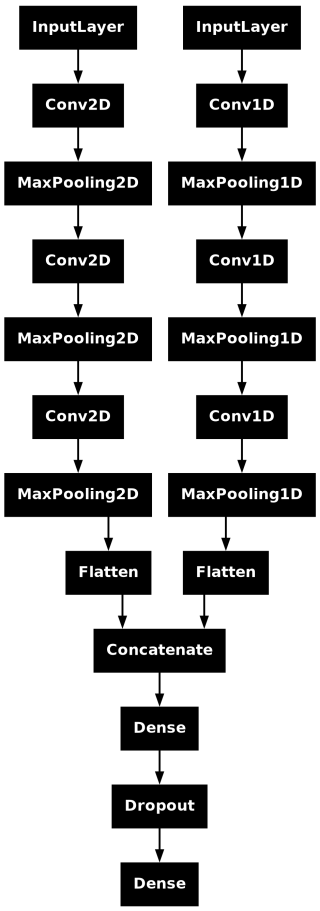


Figure 3: Multi Modal Model

The image processing branch of this architecture uses stacks of three convolutional layers (16, 32, 64) with increasing number of filters to learn hierarchical representations from raw image inputs. To reduce the computational load and downsample the feature maps, all convolutional layers are followed by a max-pooling layer depicted in the figure. Similarly, the audio processing branch of the model has three 1D convolutional with the same number of filters to capture temporal patterns as shown in the figure. The architecture of our model also has a dropout layer for regularization between the fully connected layers in the classification task to avoid overfitting. The output layer consists of ten units with softmax activation, which is appropriate for multi-class problems in terms of classification. We decided to follow best-practices in deep learning architectures in determining the parameters of the model architecture including the size of the kernels and the number of filters in the convolutional layers. In other words, the model was designed to efficiently extract relevant patterns and features from both the input modalities for the MNIST digit classifier using image and audio data.

2.3 Model Training

The image data from the MNIST dataset was merged with the audio data to train the algorithm. Input batches are passed through the model containing both types of data. Using backpropagation, the model adjusts the weights iteratively to minimize the loss function - categorical cross-entropy. Adam

optimizer was used to swap the model parameters during the training, with learning rate determined by hyperparameter optimization. Early halting is employed with a patience of three epochs for checking the validation loss and save the model from overfitting. That is, the training will stop after three consecutive epochs if the validation loss did not decrease. We used 80% of the training dataset as the training data and 20% for the validation set, this is to avoid overfitting to the training data and to track if the model is performing well on unknown data. After every epoch, the model is evaluated for accuracy and loss on the training and validation set to assess its performance and convergence. Our training runs until either the early stopping condition is met or 10 epochs have passed. Once the model was trained, we then used the validation data to test the performance and generalization of the model and how well it classifies MNIST digits.

2.4 Hyperparameter Tuning

The Optuna library, an effective tool for hyperparameter optimization, was used to tune the parameters. The methodology entails a methodical examination of a predetermined hyperparameter search space in order to determine the configuration that optimizes the selected objective function, specifically validation accuracy in this case. Optimizing the hyperparameters of the model will improve its overall performance and capacity for generalization.

The first step was defining a search space that included the range of values for every hyperparameter that needed to be tuned. This search space contained parameters including dropout rate, number of units in fully connected layers, number of filters in convolutional layers, and learning rate. Optuna provides versatility in search space definition, enabling the ranges and distributions for many kinds of hyperparameters to be specified.

Subsequently, an objective function was established to assess the model's performance for every collection of hyperparameters selected from the search space. This goal function usually entails assessing the model's performance on a validation dataset and training it with the given hyperparameters. In this instance, the model performance was evaluated using the validation accuracy as the metric.

Optuna started the optimization process by running a number of trials, each of which represented a unique set of hyperparameters sampled from the specified search space. The objective function was assessed for every trial by assessing the validation accuracy and training the model with the sampled hyperparameters. Subsequently, Optuna employed a very effective search algorithm to repeatedly examine the search space and pinpoint areas of potential interest that may result in increased validation accuracy.

Early stopping was used as a technique to reduce overfitting and boost efficiency throughout the optimization phase. By early halting trials before the validation accuracy improved for a predetermined amount of epochs, the optimizer was able to save computational resources and expedite the search process.

3 Results

A promising set of hyperparameters for the multimodal model were obtained through the Optuna optimization, providing important information on the architecture of the model.With 48 filters in the first layer to capture broad details, 128 filters in the second layer for more detailed features, and 64 filters in the third layer for even finer analysis, the image encoder appears to favor a progressive approach. In the same manner, the audio encoder gains from a customized filter configuration, beginning with 64 filters to capture basic audio properties, using another layer of 64 filters to go deeper, and using 128 filters in the final layer to extract more subtle information. This arrangement highlights how crucial it is to extract both general and specialized information from the audio data.

With 128 units, the dense layer seems to find a decent compromise between overfitting risk and model complexity. A dropout rate of 0.42 suggests that, by arbitrarily leaving out some activations during training, the model is kept from becoming overly accustomed to the training set, which may enhance its capacity to generalize to new data. Ultimately, it appears that the optimizer can effectively modify the model weights during training thanks to the learning rate of 0.000763.The other combination of hyperparameters are as shown in the table above.

Trial	Image Filters 1	Image Filters 2	Image Filters 3	Audio Filters 1	Audio Filters 2	Audio Filters 3
0	64	96	128	64	96	128
1	16	32	128	32	96	256
2	64	64	128	64	96	128
3	32	64	64	48	64	256
4	48	96	128	64	128	192
5	64	128	64	16	128	128
6	64	64	128	64	64	192
7	16	128	256	16	32	64
8	48	128	64	64	64	128
9	64	96	128	64	96	128

Table 1: Filter Parameters of Hyperparameter Optimization Trials

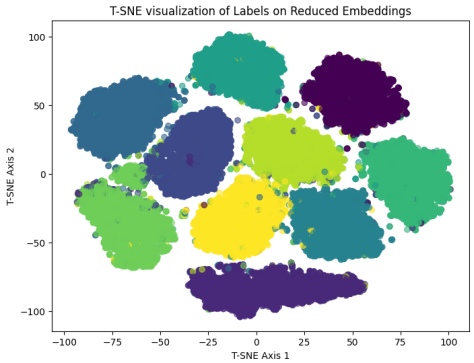
Metric	Value
Training Accuracy	0.9974
Testing Accuracy	0.9893
F1 Score	
Training Set	0.9974
Testing Set	0.9893

The above table represent the best values identified by the Optuna run. Since the hyperparameter search is fundamentally random, one must, of course, expect some slight variance in optimization results upon repetition. Nevertheless, the current data provide good grounds for this further investigation. In general, ablation studies are useful in understanding the impact of individual hyperparameters: individual hyperparameters are adjusted and the impact on model performance is tracked. It is possible to figure out which of the hyperparameters has the biggest impact on the model performance. Wider ranges of hyperparameter values would likely be expected to yield even larger performance benefits when optimized.

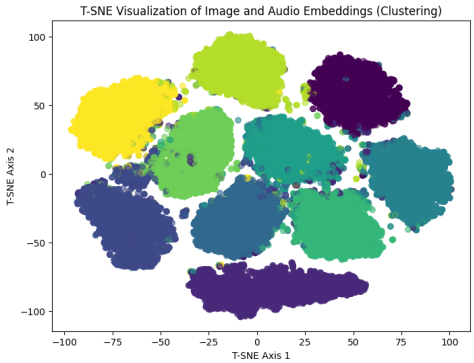
3.1 Visualisation

This section explores the properties of the learnt features through dimensionality reduction and clustering techniques, building on the high accuracy and F1 score attained by the model.

For every digit sample, the high-dimensional feature space recovered from the combined image and audio data is reduced using an t-SNE (t-distributed Stochastic Neighbor Embedding) visualization. Due to this reduction, two-dimensional display is possible, with each data point denoting a sample of digits. In an ideal world, the data points from the same digit class (0, 1, 2,..., 9) would group together in the t-SNE embedding, with unique groups denoted by various colors for each digit.



Labels on Reduced Embeddings



K-Means Clustering on Reduced Embedding

Examine the data points' arrangement for any patterns. Are particular digit forms dominating certain parts of the t-SNE plot (e.g., all rounded shapes crowded together)? Even while the individual digits may not be connected to colors directly, their relative placements can still reveal information about how the model interprets the similarities and contrasts among the digits.From the above Figures we can see that although the colors do not match in both, the cluster formations seem to be a close match.

This shows that the clusters are formed where the similar digits are clustered together.Examine the t-SNE plot to see if any separate data point clusters are present. This implies that the model has acquired the ability to distinguish between certain digit features. Better ability of the model to differentiate between various digits based on the combined image and audio input is indicated by tighter and more isolated clusters which can be seen in the above Figure.

The below image shows the K-means clustering of the Image and the Audio embeddings separately. Here, similar to the previous Figures the clusters from K-Means are tight which indicates that the similar features for each digits can be visualised.

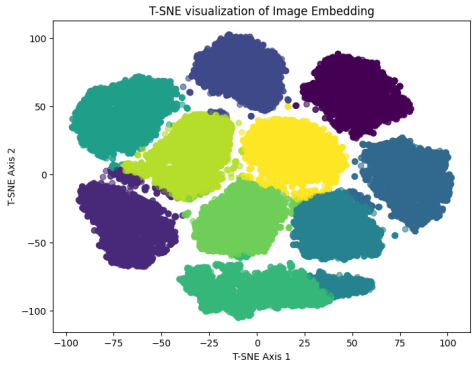
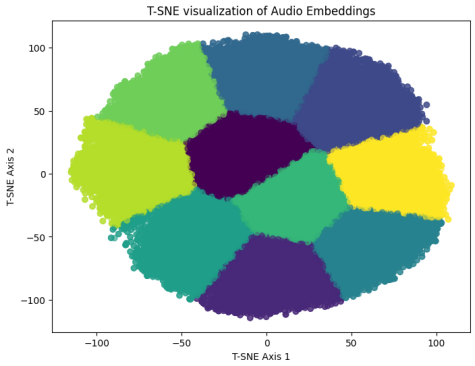


Image Embedding



Audio Embedding

4 Conclusion

To sum up, the multi-modal model created for combining audio and picture input from MNIST digits exhibits a well-thought-out training approach and an architecture. The model achieves a fine balance between complexity and generalization ability through the careful selection of convolutional layers for the extraction of spatial and temporal features, as well as the optimization of hyperparameters like filter sizes, units in the fully connected layer, dropout rates, and learning rates. The model benefits from these complementary data sources by combining both visual and auditory modalities, which improves its performance. Overfitting is prevented by early halting based on validation loss during training, and effective model training is encouraged by using moderate batch sizes and a restricted number of epochs. All things considered, the model's ability to achieve high accuracy on the MNIST digit picture and audio data can be credited to the combination of its multi-modal design, careful hyperparameter optimization, and reliable training approaches working.

Even with these design choices, there remains room for development. Rotations and translations are two examples of data augmentation techniques that could increase the diversity of the dataset. Further exploration of deeper architectures or transfer learning algorithms may reveal more complex correlations between image and audio data. Additionally, investigating more sophisticated feature engineering techniques can yield richer input representations for the model, which could result in additional performance improvements.