Contents

[Project Report 3](#_Toc173012812)

[Task 1 – Classification Problem (Single label Classification) 3](#_Toc173012813)

[Introduction 3](#_Toc173012814)

[Dataset Overview and Pre-processing 3](#_Toc173012815)

[Data Loading and Initial Exploration 3](#_Toc173012816)

[Pre-processing Steps 3](#_Toc173012817)

[Handling Missing Data 3](#_Toc173012818)

[Challenges 4](#_Toc173012819)

[Methodology and Techniques 4](#_Toc173012820)

[Results and Discussion 4](#_Toc173012821)

[Visualizations 5](#_Toc173012822)

[Model Comparison and Selection 7](#_Toc173012823)

[Hyperparameter Tuning on Random Forest classifier 7](#_Toc173012824)

[Task 2 – Triple MNSIT (Multi labels dataset) 8](#_Toc173012825)

[Dataset Overview and Pre-processing 8](#_Toc173012826)

[Data Loading and Initial Exploration 8](#_Toc173012827)

[Pre-processing Steps 9](#_Toc173012828)

[Challenges 9](#_Toc173012829)

[Methodology and Techniques 9](#_Toc173012830)

[Model Comparison and Selection 9](#_Toc173012831)

[Hyperparameter Tuning 9](#_Toc173012832)

[Results and Discussion: 10](#_Toc173012833)

[Visualization 13](#_Toc173012834)

[Conclusion 18](#_Toc173012837)

[Ethical, Legal and Social Considerations in AI 18](#_Toc173012838)

# Project Report

## Task 1 – Classification Problem (Single label Classification)

### Introduction

In this assignment, we demonstrate several machine learning techniques to solve classification problems.

By processing the datasets (both numerical and categorical features), we explore the process of data pre-processing, model building, and evaluation.

This assignment showed the importance of data preparation, model selection, and fine-tuning to achieve the most optimal performance in machine learning.

### Dataset Overview and Pre-processing

Dataset1 consisted of 925 rows and includes the features var1, var2, var3, var4, var5, var6, and var7, with the target column as the label.

Among these features, var3 and var6 are categorical, while var7 contains datetime values. The other features such as var1, var2, var4, and var5, are numerical.

### Data Loading and Initial Exploration

The dataset was loaded and initial exploration was conducted to understand its structure and identify any data quality issues.

The first step is checking for missing values, which revealed a significant amount of missing data in the var4 column. Missing Data in var4 about 600 missing values (about 65% missing).

### Pre-processing Steps

For the categorical features, we converted var3 and var6 into numerical values using label encoding.

Regarding the datetime feature, var7, we converted it to a datetime format and split it into var7\_date and var7\_time.

### Handling Missing Data

Since var4 had around 65% missing values, we explored to impute missing values in a dataset using a KNN imputer.

The KNN (K-Nearest Neighbours) imputer is a method that replaces missing values based on the mean or median values of the nearest neighbours.

Since KNN works with distance computations, first we scale the dataset.

### Challenges

We faced several challenges during the pre-processing phase of the dataset.

One major issue was handling a significant amount of missing data in var4, which required careful consideration of imputation techniques or potential removal. Ensuring the correct encoding of categorical features, such as var3 and var6, was important to maintain the integrity of the data.

Properly managing and splitting datetime values in var7 also posed a challenge, as it was need to accurately separate and format the date and time components for further analysis.

### Methodology and Techniques

Model Selection: Applied Logistic Regression, Decision Trees, Random Forests, and Artificial Neural Networks (ANN) to the classification problem.

Hyperparameter Tuning: Use Grid Search and cross-validation to fine-tune model parameters.

Evaluation Metrics: Assessed models using accuracy, precision, recall, and F1-score.

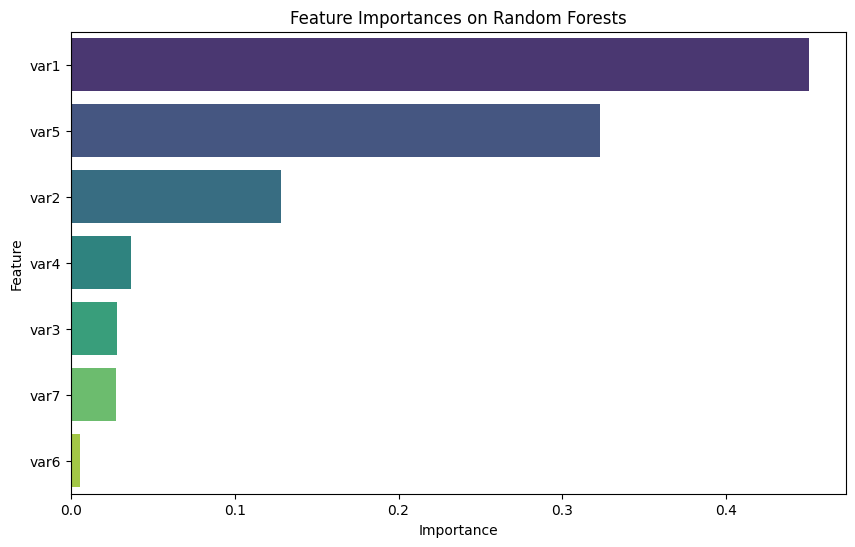
### Results and Discussion

Random Forests model with or without normalization outperformed other models and achieved the highest accuracy, around 96.2%.

Important features for the Random Forest model are var1, var5, and var 2. We plotted feature importance on Random Forests, Limited Depth Decision Trees, and Partial Dependence Plots to compare the features.

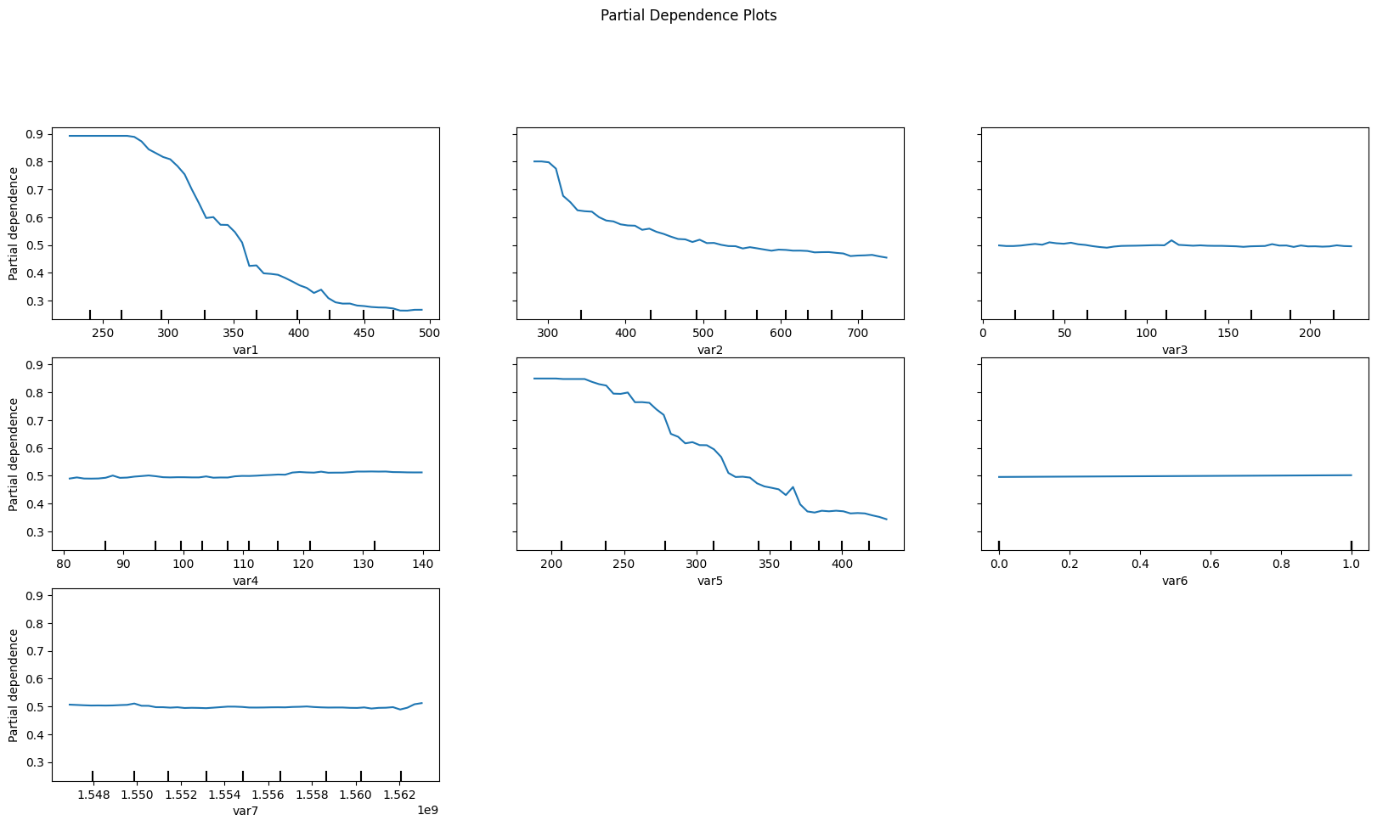
### Visualizations

#### Feature importance



#### Limited Depth Decision Tree

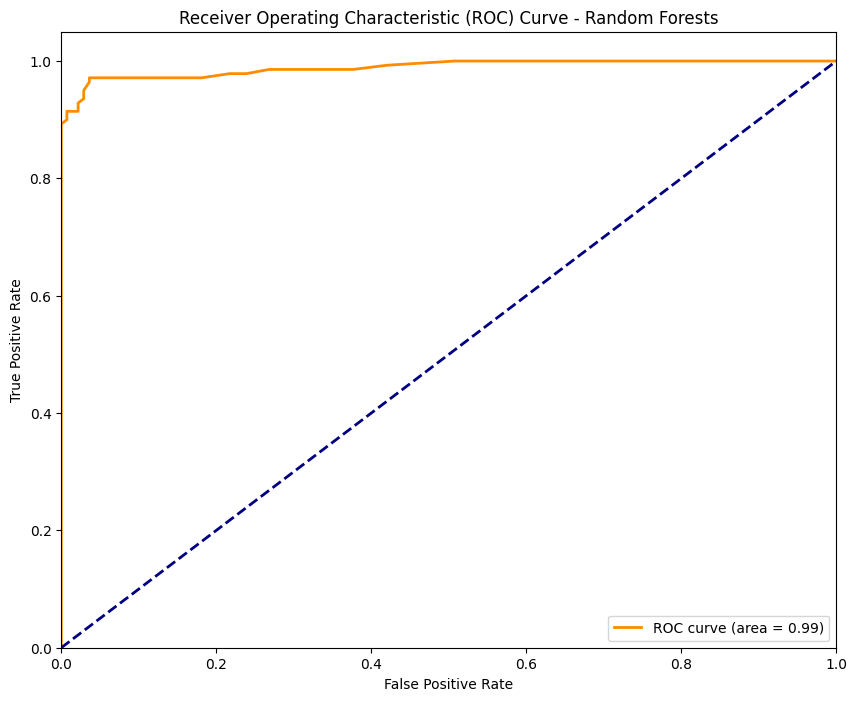
#### Partial Dependence Plots



Since var4 is not a significant feature, imputing the mean or removing the column does not affect the accuracy of the model.

Confusion matrix and heatmap for the final selected model shows that the model has high and reliable performance.

#### ROC Curve using Random Forests

****

### Model Comparison and Selection

|  |  |
| --- | --- |
| Classifier | Accuracy |
| Random Forests | 0.9621621621621622 |
| Artificial Neural Networks | 0.9621621370315552 |
| Logistic Regression | 0.9513513513513514 |
| Support Vector Machines | 0.9424460431654677 |
| Decision Trees | 0.9405405405405406 |

### Hyperparameter Tuning on Random Forest classifier

Best Parameters:

{

    'bootstrap': [False],

    'max\_depth': [10],

    'max\_features': ['sqrt'],

    'min\_samples\_leaf': [2],

    'min\_samples\_split': [2],

    'n\_estimators': [300]

}

**Conclusion**

Effective data pre-processing is important for model performance. Ensemble methods like Random Forest can provide significant improvements. Fine-tuning and cross-validation are necessary steps to optimize model performance.

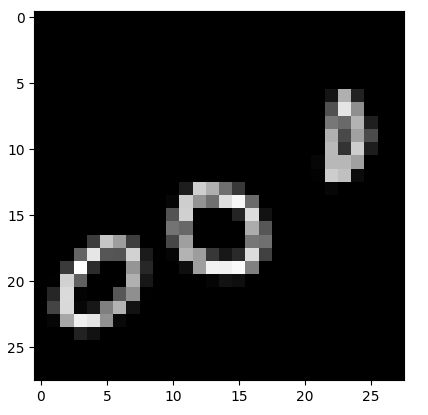
In future, we can explore more advanced techniques for handling missing data and feature engineering. Experiment with other methods and deeper neural networks, and investigate the impact of feature selection and dimensionality reduction techniques.

## Task 2 – Triple MNSIT (Multi labels dataset)

### Dataset Overview and Pre-processing

Task 2 assignment is focusing on the objectives and importance of multi-label classification on multiple digits from images, where each image contains several digits.

The dataset for this project consists of images each labelled with three digits.



### Data Loading and Initial Exploration

To load and explore the dataset, we define the load\_images\_from\_folder function, which reads images from subdirectories, resizes them, converts them to grayscale, and assigns labels based on subdirectory names. Directories for training, validation, and test datasets are specified to organize the data.

We print the contents of these directories to verify their structure. We load the datasets into arrays of images and corresponding labels. This process prepares the dataset for further processing and model training.

### Pre-processing Steps

Pre-processing steps taken to prepare the data for modelling.

This includes reshaping and normalizing pixel values, one-hot encoding of labels to convert labels to categorical, creation of a custom data generator for efficient data handling, set up learning rate scheduler to adjust the learning rate during training, and configure call-backs for early stopping to prevent overfitting by stop the training when the validation loss stops improving.

### Challenges

The challenges during pre-processing involved reshaping the images to fit the model's input requirements, normalizing pixel values to the range [0, 1], and encoding the labels for a multi-label setup using one-hot encoding.

We faced several challenges when we need to choose which activation function and which loss function, we need to use.

### Methodology and Techniques

CNN models were developed, comparing categorical cross entropy with Binary cross entropy as loss functions and Sigmoid versus SoftMax as activation functions.

Techniques like Early Stopping, Drop out were used to improve training performance and avoid overfitting.

### Model Comparison and Selection

Binary cross entropy as the loss function outperformed categorical cross entropy, showing lower loss and higher accuracy for multi-label classification, demonstrating its suitability for this task.

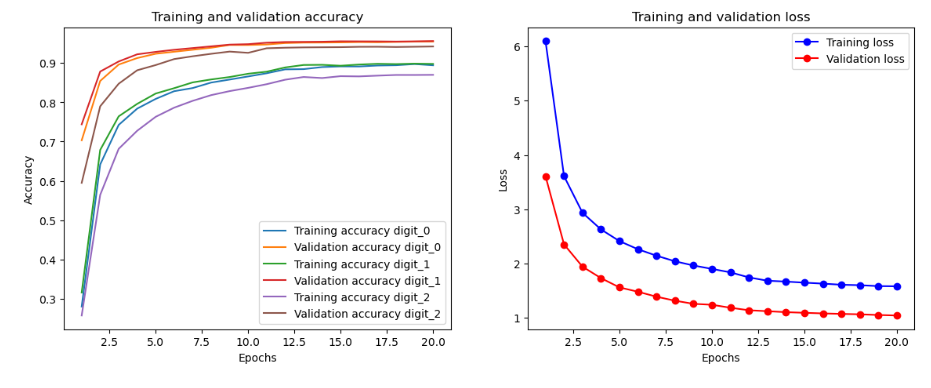
While recommendations suggest Sigmoid for multi-label and SoftMax for multi-class, our experiment shows the use of SoftMax in output layer achieved the highest accuracy.

### Hyperparameter Tuning

Hyperparameter tuning performed using Keras-Tuner to find the best parameters, beside adjust the model parameters like learning rate and increase K-fold cross-validation to more accurately evaluate model performance across different subsets of data, and implement batch normalization in the model to normalize the inputs for each mini-batch, reducing internal covariate shift and stabilizing training.

### Results and Discussion:

Initial models using **Sigmoid activation and categorical cross entropy** as the loss function achieved high accuracy (around 95%) and loss (around 1.00), indicating no underfitting or overfitting.



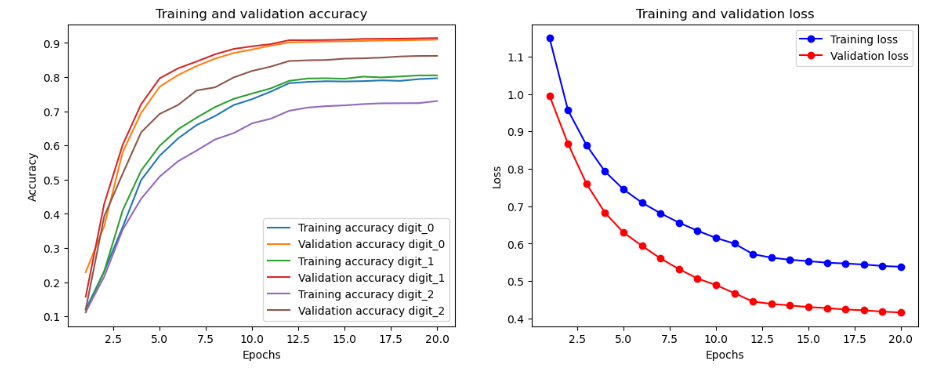
Test loss: 1.0049712657928467

Test accuracy for digit\_0: 0.9526000022888184

Test accuracy for digit\_1: 0.9549999833106995

Test accuracy for digit\_2: 0.9416499733924866

Switching to **Sigmoid activation and binary cross entropy** as the loss function reduced losses (around 0.3), but lowered accuracy. The final evaluation on the test dataset showed accuracies (above 90%) for all digits.



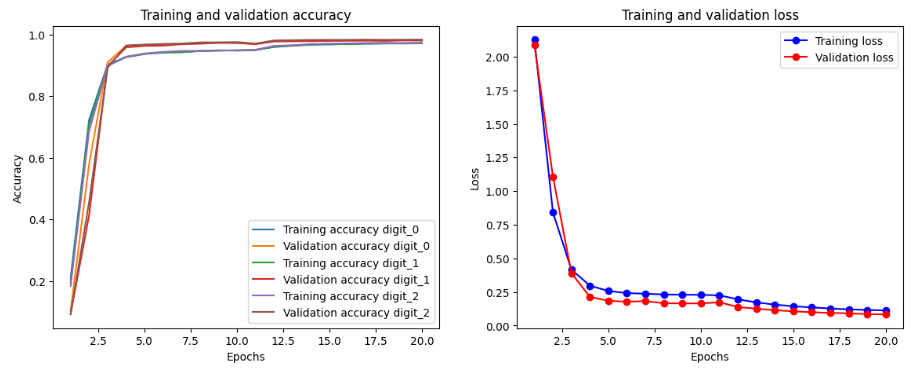
Test loss: 0.3763981759548187

Test accuracy for digit\_0: 0.907800018787384

Test accuracy for digit\_1: 0.9002500176429749

Test accuracy for digit\_2: 0.9164999723434448

The model using **Softmax activation and Binary Cross Entropy** as the loss function showed better performance with higher accuracy (around 98%) and loss (around 0.07).



After hyperparameter tuning, the model demonstrated improvement and better performance.

Using **Keras-Tuner**, we found the best parameters and implemented into the model:

val\_loss: 0.07059618085622787

Best val\_loss So Far: 0.06173119693994522

Total elapsed time: 05h 47m 46s

{'conv\_1\_filters': 96, 'conv\_2\_filters': 128, 'conv\_3\_filters': 128, 'dense\_units': 256}

Final testing on unseen data after Hyperparameter Tuning on Softmax activation function and Binary Crossentropy as the loss function confirmed the model's highest accuracy (around 98.7%) and lowest loss (around 0.007).

Test loss: 0.06563404202461243

0.008683374151587486

0.008130848407745361

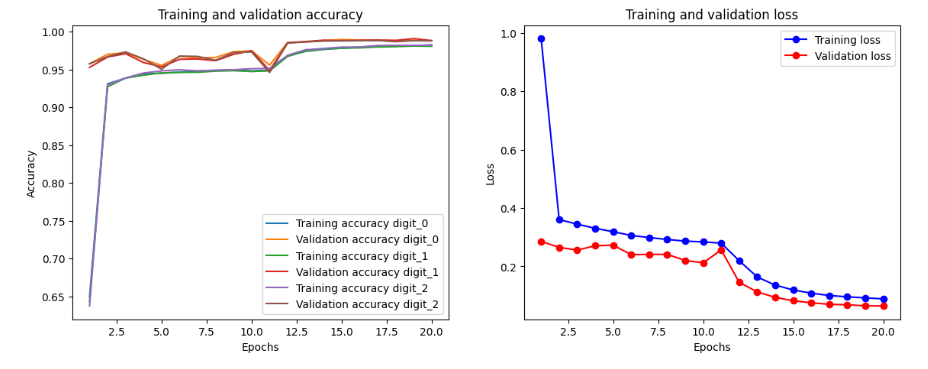
0.007283779326826334

0.986299991607666

0.9872499704360962

0.9888499975204468

There is high accuracy on the test set which the model has never seen before, that means the model can predict on the new unseen data. The model regularized well and trained appropriately, no overfitting nor underfitting.



### Visualization

#### Confusion Matrix

### 

A graph of blue squares

Description automatically generated

A graph of a diagram

Description automatically generated with medium confidence

The diagonal elements represent the correctly classified ones, with the highest values along the diagonal showing correct predictions. Off-diagonal values show where the model making errors.

### The model is confused when distinguishing between digits 0 and 3, and also between digits 0 and 8. Their features are similar to each other and are not distinguished well by the model.

#### Visual Representation of The Filters/ Kernels and Feature map

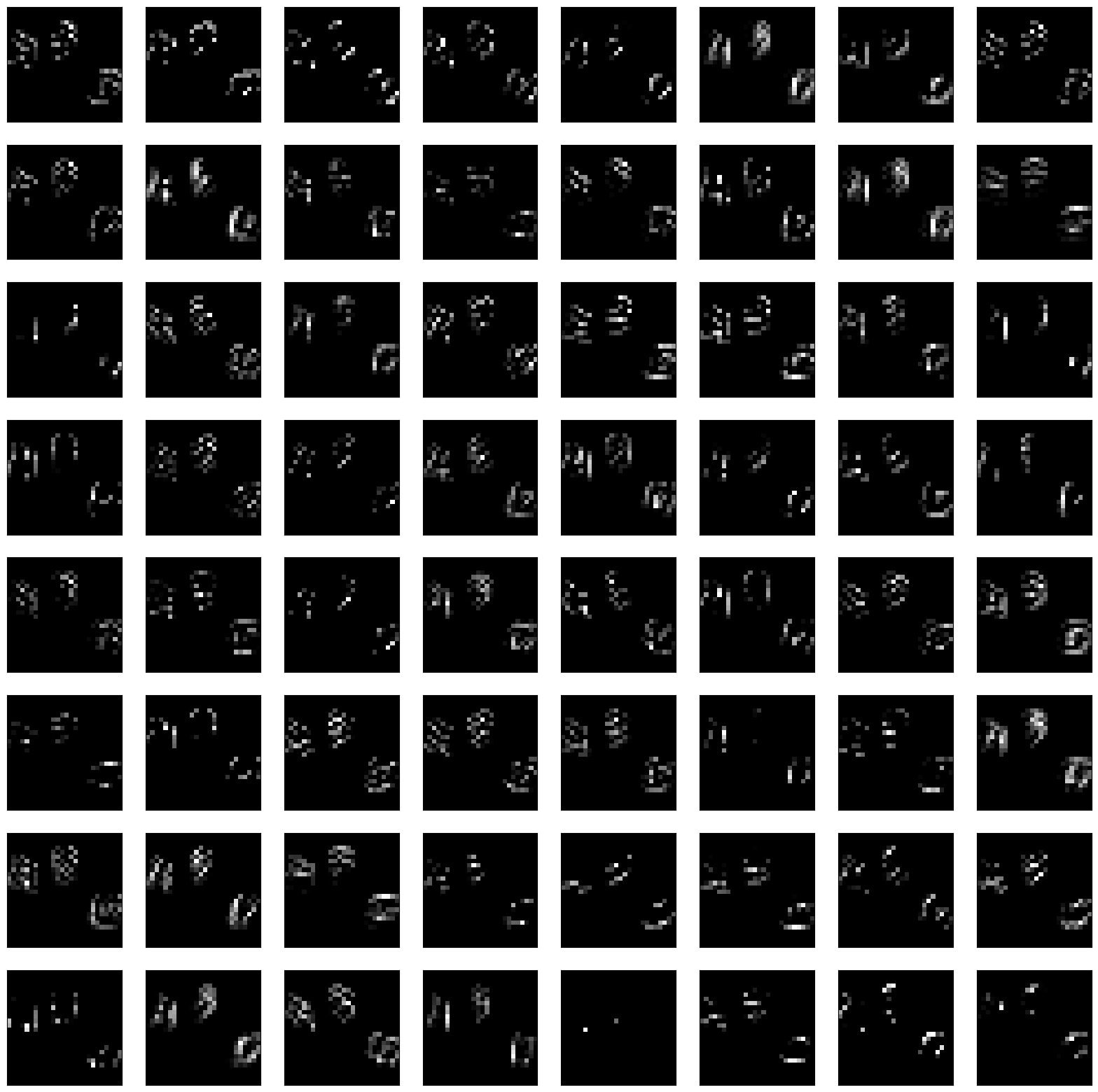
Filters/Kernel is small matrices that slide over the input image to detect specific features such as edges, textures, or patterns. Each filter is applied across the entire input image, and the result is a feature map.

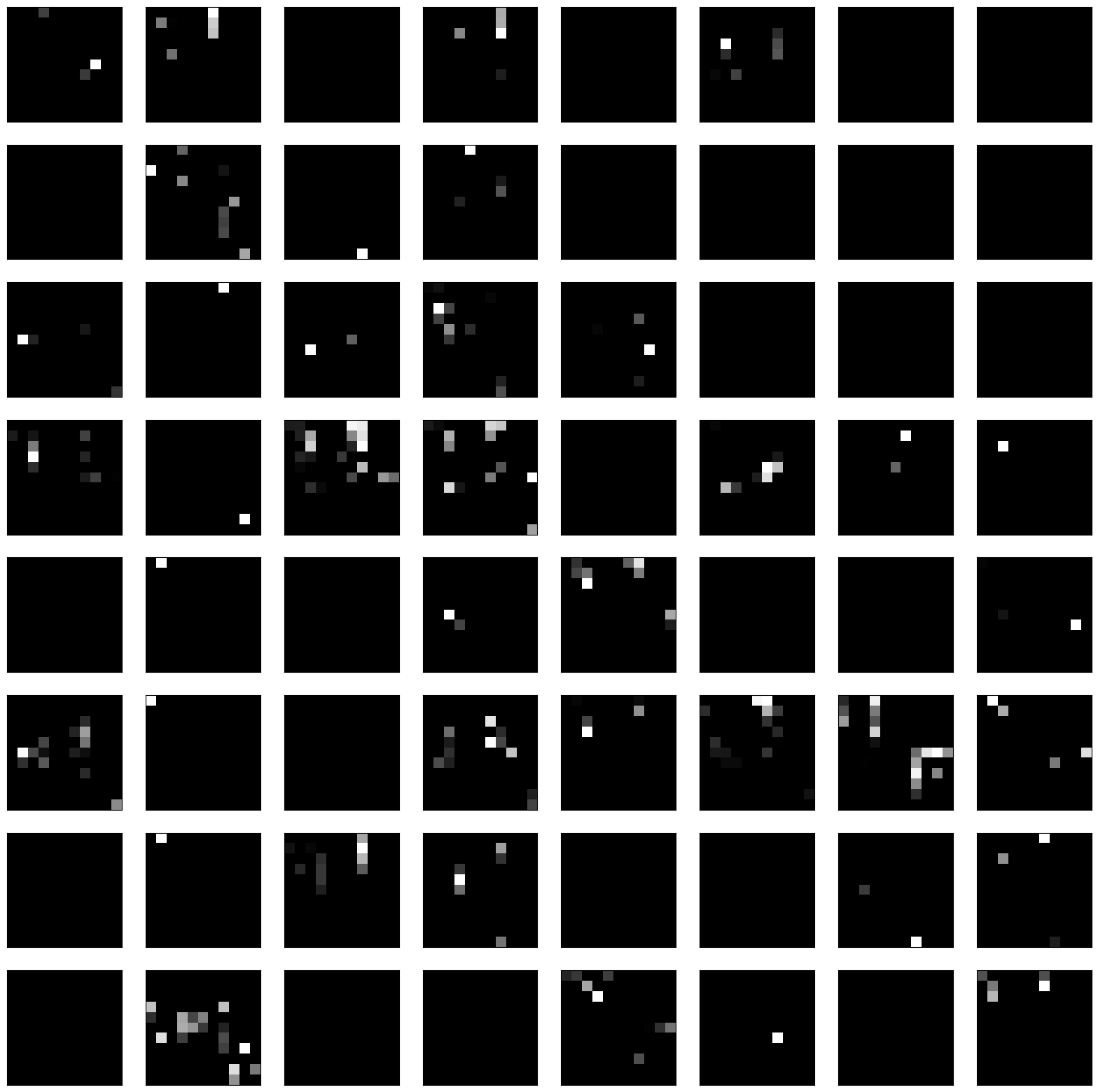
The visual representation of the filters helps to understand what each filter is focusing on in the input images. The filters have dimensions (3, 3, 1, 96), where 3 and 3 are the height and width of each filter. 1 is the number of input channels for grayscale image, there is only one channel). 96 is the number of filters in this layer.

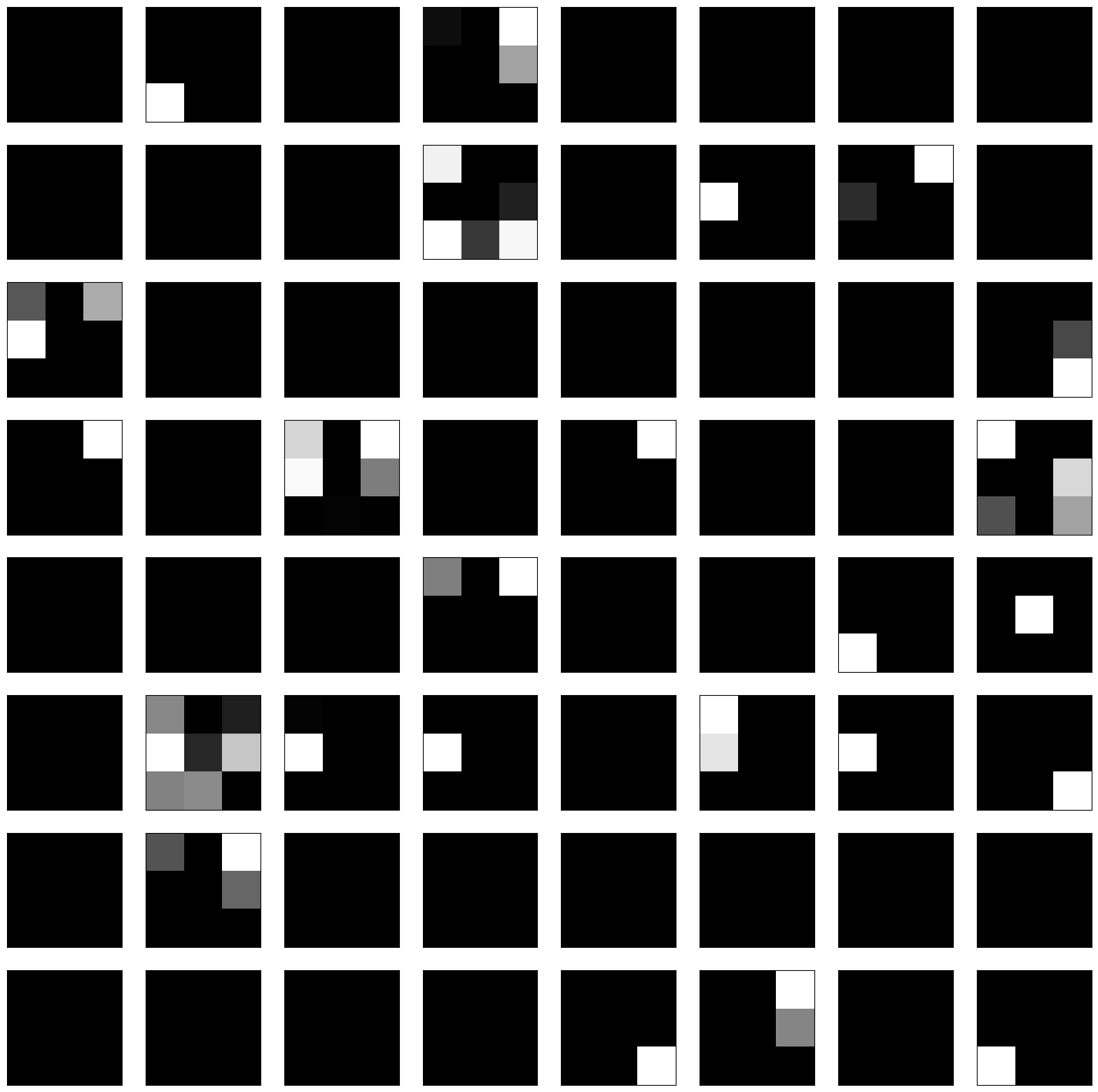
The filters have purpose to detect specific features within the input images. During training, the weights of these filters are adjusted to minimize the loss function, thereby learning the features that are most relevant for the task. Visualizing the filters can give insights into what the model is learning.

In this example, we chose a random handwritten image (number 430) from the train dataset, and predicted the feature outputs.









The patterns seen in the feature show of what each filter has learned. For example, some may detect edges, other textures, and some specific shapes. Early layers earn simple features, then deeper layers learn more complex features and patterns.

Saliency Maps highlight the most important areas of the input image in pixel level, showing which parts of the input image contribute most to the output prediction.

A red and yellow pixelated image

Description automatically generated

Similar purpose with Saliency Map but smoother, Gradcam highlight the most important areas of the input image in region level, showing which parts of the input image contribute most to the output prediction.

A group of yellow dots

Description automatically generated

### Conclusion

The project successfully demonstrated the application of CNNs to multi-label image-based digit classification, with SoftMax activation and binary cross entropy as the loss function showing the highest

performance over Sigmoid activation and categorical cross entropy as loss function. They are more improving after hyperparameter tuning. This is showing how importance in selecting appropriate loss functions and optimizing model architectures.

Future work could explore more complex architectures, additional regularization techniques, and further hyperparameter tuning to enhance model performance and robustness, potentially experimenting with different convolutional layer configurations to improve focus on relevant parts of the image.

### Ethical, Legal and Social Considerations in AI

When developing AI models for multi-label image classification, such as with triple MNIST digits, it's important to address ethical, legal, and social issues.

Data protection laws like GDPR or CCPA must be adhered to, ensuring that any personal data is handled with strict privacy measures.

The ownership of the AI model, its training data, and its outputs need to carefully consider ensuring responsible and beneficial AI deployment.