A close up of a sign

Description automatically generated

**Project Report**

**On**

**Handwritten Digit Identification**

**Fundamentals of Machine Learning Lab**

**Subject Code: DSE 2242**

| **Names** | **Registration No** |
| --- | --- |
| Nayavanath Yogi | **220968344** |
| Mehul Garg | **220968332** |
| Ranveer Singh Sayal | **220968340** |

**Department of Data Science & Computer Applications,**

**Manipal Institute of Technology,**

**Manipal**

**JAN -MAY 2024**

**Table of Contents**

Abstract

1. Introduction
2. Methodology
   1. Flowchart/ block diagram of the proposed method
   2. Explain each phase of the block diagram.
3. Experimental Setup
4. Dataset
5. Results, and Discussion
6. Conclusion

**ABSTRACT**

Handwritten digit recognition is a classic problem in the field of machine learning, with the MNIST dataset serving as a benchmark for evaluating various algorithms' performance. In this project, we undertake a comprehensive investigation into three widely-used classification techniques: k-Nearest Neighbours (KNN), Support Vector Machine (SVM), and Ensemble Methods, applied to the MNIST handwritten digit classification task. Throughout the project, we conduct extensive experiments and analyses to compare the performance of KNN, SVM, and ensemble methods. We employ cross-validation techniques to ensure the reliability and generalizability of our results, and visualize the models' performance using confusion matrices and ROC curves.

Our findings shed light on the relative strengths and weaknesses of KNN, SVM, and ensemble methods in the context of MNIST handwritten digit classification. By comparing these approaches, we aim to provide insights that can guide the selection of appropriate algorithms for similar pattern recognition tasks in practical applications. This study contributes to advancing our understanding of machine learning algorithms' efficacy in image classification and underscores the importance of empirical evaluation in model selection and development.

**CHAPTER 1**

**INTRODUCTION**

The project begins with data preprocessing, where the MNIST dataset, comprising 28x28 pixel grayscale images of handwritten digits ranging from 0 to 9, is prepared for training and testing. We then partition the dataset into training and testing subsets to facilitate model development and performance evaluation.

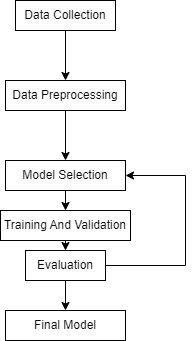
Our first endeavour involves implementing the KNN algorithm, a non-parametric method that classifies data points based on their proximity to the nearest neighbours in the feature space. We explore different values of k, the number of nearest neighbours considered during classification, and rigorously evaluate the model's performance using standard metrics such as accuracy, precision, recall, and F1-score.

Subsequently, we employ the Support Vector Machine (SVM) algorithm, a powerful supervised learning technique known for its effectiveness in high-dimensional feature spaces. By tuning various hyperparameters and kernel functions, we aim to optimize the SVM model's performance for accurately classifying handwritten digits.

In addition to individual classifiers, we delve into ensemble learning approaches, which combine multiple base classifiers to enhance predictive accuracy and robustness. Specifically known for their ability to mitigate overfitting and improve generalization performance.

**CHAPTER 2**

**METHODOLOGY**

****

1. Data Collection
   1. The MNIST data set is a very well known data collected by MNIST comprising of over 60000 images of handwritten images
   2. It is easy to download as .gz file and can be unzipped and extracted in the jupyter notebook environment
2. Data Preprocessing
   1. The data consists of 4 files x\_train, x\_test, y\_train and y\_test. x\_train and x\_test are in the form of y x 28 x 28 array consisting of the image pixel values.
   2. We transform it into a dataframe by reducing its dimensionality
   3. We compile all the data into one single data frame and export as csv
   4. Columns with only 0 values are omitted as they have no effect on the model
   5. Data is split randomly into x\_train, x\_test, y\_test and y\_train
3. Model Selection
   1. 3 models are selected to classify the dataset namely KNN, SVM and ensemble max voting with models- KNN and SVM
   2. The models are trained with x\_train and y\_train data
4. Evaluation
   1. The performance of the models are compared by various measures namely-
      1. accuracy score
      2. classification report
      3. confusion matrix
5. Result Analysis
   1. Analyze classification reports and performance metrics for insights into model strengths and weaknesses.
   2. Identify areas for improvement and optimization.

**CHAPTER 3**

**EXPERIMENTAL SETUP**

● Software: Google Collab, Python, scikit-learn, OpenCV

● Dataset: MNIST handwritten digits dataset

**CHAPTER 4**

**DATASET**

The MNIST dataset is stored in a structured format that facilitates easy access and manipulation for machine learning tasks. Here's a detailed explanation of how the data is stored:

1. Image Data: The core of the MNIST dataset consists of grayscale images of handwritten digits. Each image is represented as a matrix of pixel values, where each pixel corresponds to a grayscale intensity ranging from 0 to 255. These images are stored as 28x28 matrices, resulting in a total of 784 pixels per image.
2. Label Data: Alongside the image data, the MNIST dataset also includes corresponding labels for each image, indicating the digit that the image represents. These labels are stored as integers ranging from 0 to 9, with each integer representing a specific digit. For example, the label 0 corresponds to the digit '0', the label 1 corresponds to the digit '1', and so on.
3. Data Format: The MNIST dataset is often stored in a common file format such as CSV (Comma-Separated Values) or HDF5 (Hierarchical Data Format version 5). These formats provide a standardized way to store and access the image and label data, making it easy to import the dataset into various machine learning frameworks and libraries.

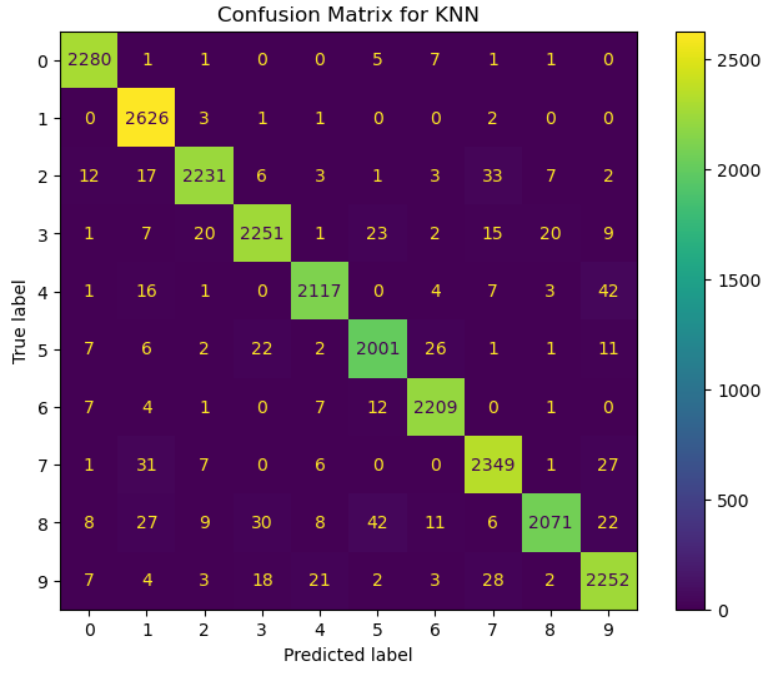
Overall, the structured format of the MNIST dataset enables researchers and practitioners to easily access and manipulate the data for tasks such as image classification, digit recognition, and algorithm development. Its simplicity and accessibility have made it a cornerstone of the machine learning community, serving as a benchmark for evaluating the performance of different models and techniques.

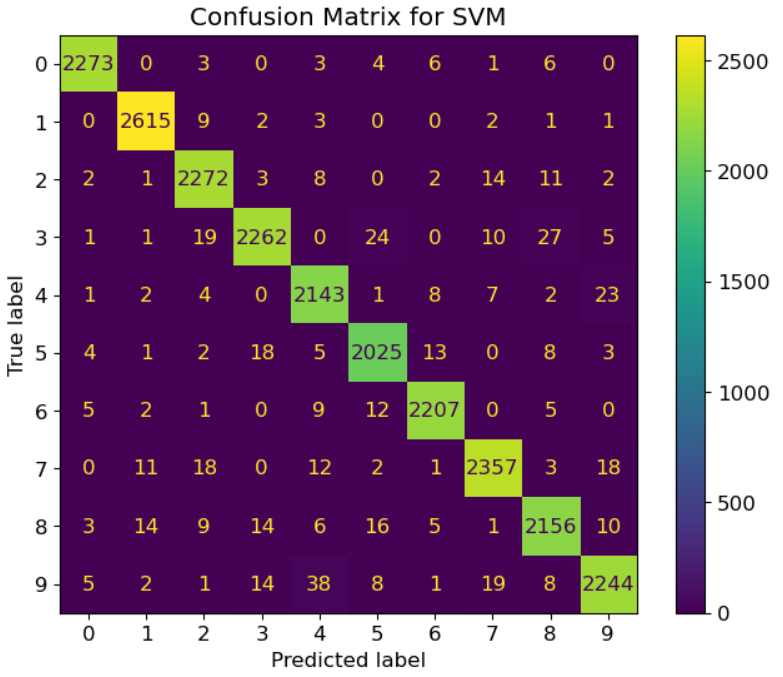
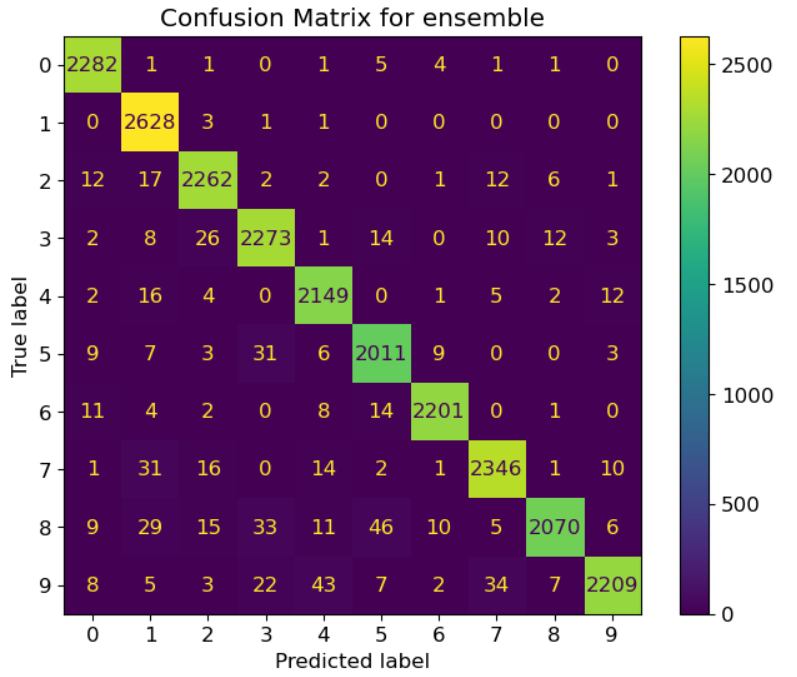
**CHAPTER 5**

**RESULT AND DISCUSSION**

The project involved assessing the performance of three distinct machine learning algorithms, namely k-Nearest Neighbors (KNN), Support Vector Classifier (SVC), and Ensemble of max voting between KNN and SVC in the task of numerical value detection. Results revealed that the Support Vector Classifier (SVC) algorithm excelled with an accuracy rate of 0.9764, followed by Ensemble and k-Nearest Neighbours algorithms, with accuracies of 0.9710 and 0.9691 respectively.

Additionally, the classification reports offered detailed insights into the precision, recall, F1-score, and support for each class, facilitating a comprehensive examination of the algorithms' performance across different metrics. Such detailed analysis aids in understanding the strengths and weaknesses of the algorithms, thus providing valuable information for further optimization and refinement of the face mask detection system.





**CHAPTER 6**

**CONCLUSION**

The project results clearly establish the practicality of harnessing machine learning algorithms to detect numeric values effectively. Among the array of classifiers tested, the Support Vector Classifier (SVC) algorithm emerges as the most effective classifier for this task, achieving the highest accuracy among the evaluated models. However, there remains a pressing need for further investigation to delve into cutting-edge methodologies and fine-tune the numeric value detection system's resilience to meet the demands of real-world scenarios. This necessitates a deep dive into exploring innovative approaches and enhancing the overall reliability and adaptability of the system to ensure seamless integration into practical applications.