From Pixels to Predictions:

Implementing and Comparing Machine Learning Models for the MNIST Dataset



Aikaterini Antoniou

EC Utbildning

Course: Machine Learning

2024-03

# 

# Abstract

This report delves into the development, evaluation and deployment of various machine learning models for the task of handwritten digit recognition using the MNIST dataset as a standard. Among the models compared a particular focus is placed on the Convolutional Neural Network (CNN), which is assessed against traditional machine learning algorithms like Random Forest and SVM. Through thorough testing, the CNN demonstrated superior accuracy, showcasing its effectiveness in image-based classification tasks. The report further explores the deployment of the CNN model within a Streamlit web application, enabling interactive user engagement and real-time digit prediction. Key findings underline the CNN's robustness and the potential for machine learning models to enhance practical applications in digit recognition.

# Abbreviations and Terms

ML: Machine Learning

CNN: Convolutional Neural Network

SVM: Support Vector Machine

RF: Random Forest

MNIST: Modified National Institute of Standards and Technology database

Tables of Contents

[Abstract 2](#_Toc162203427)

[Abbreviations and Terms 3](#_Toc162203428)

[1 Introduction 1](#_Toc162203429)

[1.1 Purpose and Question Research 1](#_Toc162203430)

[2 Theory 2](#_Toc162203431)

[2.1 Overview of Machine Learning 2](#_Toc162203432)

[2.2 Insights into the MNIST Dataset 2](#_Toc162203433)

[2.3 The Models Used 2](#_Toc162203434)

[2.3.1 Convolutional Neural Networks (CNNs) for Image Recognition 2](#_Toc162203435)

[2.3.2 Support Vector Machines (SVM) in Classification 2](#_Toc162203436)

[2.3.3 Random Forests 2](#_Toc162203437)

[2.4 Why these models and why not other models 3](#_Toc162203438)

[2.4.1 Why Random Forest? 3](#_Toc162203439)

[2.4.2 Why SVM? 3](#_Toc162203440)

[2.4.3 Why CNN? 3](#_Toc162203441)

[2.4.4 Why Not Regression Models or Lasso? 3](#_Toc162203442)

[2.5 Streamlit's Role in Machine Learning Deployment 4](#_Toc162203443)

[3 Method 5](#_Toc162203444)

[3.1 Data Acquisition and Preprocessing 5](#_Toc162203445)

[3.2 Model Implementation 5](#_Toc162203446)

[3.3 Model Training and Evaluation 5](#_Toc162203447)

[3.4 Application Deployment 5](#_Toc162203448)

[3.5 Model Validation 5](#_Toc162203449)

[4 Results and Discussion 7](#_Toc162203450)

[4.1 Results 7](#_Toc162203451)

[4.2 Discussion 7](#_Toc162203452)

[4.3 Streamlit Application 8](#_Toc162203453)

[4.3.1 Deployment Process 8](#_Toc162203454)

[4.3.2 Application Interface and Functionality: 8](#_Toc162203455)

[4.3.3 Results in practice 8](#_Toc162203456)

[4.3.4 Significance and Broader Implications of the Application 10](#_Toc162203457)

[5 Conclusion 11](#_Toc162203458)

[6 Theoritical Questions 12](#_Toc162203459)

[7 Self-evaluation 17](#_Toc162203460)

[Appendix A: Data Acquisition and Preprocessing 18](#_Toc162203461)

[Appendix B: Random Forest Model 18](#_Toc162203462)

[Appendix C: SVM Model 18](#_Toc162203463)

[Appendix D: CNN Model 18](#_Toc162203464)

[Appendix E-1:CNNs Results 19](#_Toc162203465)

[Appendix E-2: CNNs Results 19](#_Toc162203466)

[Appendix F-1: Preview of the code in Spyder 20](#_Toc162203467)

[Appendix G 1-10: Uploaded images, preprocessed and predicted digits 21](#_Toc162203468)

[References 31](#_Toc162203469)

# Introduction

In the realm of machine learning, the MNIST dataset has been recognized as a cornerstone for benchmarking the performance of algorithms in the domain of image recognition. Comprising 70 000 images of handwritten digits, this dataset serves not only as a foundational tool for learning and experimentation but also as a critical benchmark for assessing the advancements in machine learning techniques (Makariev, 2023). The significance of handwritten digit recognition, as exemplified by the MNIST dataset, extends across various practical applications, from postal mail sorting (Naga Sai Nischal et al., 2020) to bank check processing, highlighting its role in automating tasks that require visual perception.

The objectives of this project were numerous. Initially, the focus was to develop, evaluate, and compare various machine learning models to assess their efficacy in recognizing handwritten digits. While traditional algorithms such as Support Vector Machines (SVM) and Random Forests (RF) have demonstrated considerable success, an emphasis was placed on exploring Convolutional Neural Networks (CNN), renowned for their prowess in image analysis (Wikipedia, 2024). Furthermore, the project sought to investigate the models' deployment within a practical, user-oriented application, utilizing the Streamlit framework to create an interactive web application. This phase aimed to demonstrate the models' real-world applicability and provide insights into the challenges and considerations involved in their deployment.

## Purpose and Question Research

The primary goal of this project is to explain the mechanisms and effectiveness of various machine learning strategies in digit recognition, emphasizing the transition from theoretical models to practical applications. However, in order to achieve this goal, the following questions must be answered: How do various machine learning models, including Convolutional Neural Networks, perform in the development, evaluation, and deployment for recognizing handwritten digits, and what distinguishes CNNs in this comparative analysis?

# Theory

This section offers a comprehensive narrative that sketches the theoretical foundations and justifies the selection of specific models and tools used in my project, positioning them within the context of image classification and machine learning deployment. Here are some theoretical concepts on which this work was based:

## Overview of Machine Learning

Machine learning stands as a pivotal technology within artificial intelligence, empowering systems to derive insights and make decisions based on data. This field primarily segregates into supervised, unsupervised and reinforcement learning paradigms. The present project is anchored in supervised learning, wherein predictive models are trained using labeled datasets (Géron, 2019,p.7).

## Insights into the MNIST Dataset

The Modified National Institute of Standards and Technology (MNIST) dataset is a large collection of handwritten digits (0-9) commonly used for training various image processing systems. The dataset contains 70 000 , 28x28 pixel grayscale images of digits, divided into a training set of 60 000 examples and a test set of 10 000 examples. It is widely used for benchmarking classification algorithms (Makariev, 2023).

## The Models Used

### Convolutional Neural Networks (CNNs) for Image Recognition

CNNs are a class of deep neural networks, most commonly applied to analyzing visual imagery. They are specialized in parsing through image data and leverage unique properties such as spatial hierarchy in images to effectively identify patterns and characteristics. These networks employ layers of convolutions alongside pooling and nonlinear operations, culminating in the ability to discern intricate structures within images which makes them exceptionally suited for tasks like digit recognition in the MNIST dataset (Datagen, ‘’Image Classification Using CNN: Introduction and Tutorial’’).

### Support Vector Machines (SVM) in Classification

SVMs are supervised learning models that analyze data used for classification and regression analysis. They are renowned for their robustness in classification scenarios, particularly when dealing with high-dimensional data. They work by finding the hyperplane that best divides a dataset into classes, which is particularly useful for clear categorization of data points. By constructing an optimal hyperplane, SVMs aim to achieve a clear margin of separation between different class labels. Their effectiveness in discerning boundary lines between categories renders them a valuable contender in the realm of image classification (Kanade, 2022).

### Random Forests

Random Forest is an ensemble learning method for classification and regression that operates by constructing multiple decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Embodying the concept of 'wisdom of the crowd,' Random Forest algorithms amalgamate the predictions from multiple decision trees to forge a final verdict. This ensemble approach mitigates the risk of overfitting, common in individual decision trees, thereby ensuring more reliable and generalizable classification outcomes, pertinent to the analysis of image data (ebrary.net, ‘’Random Forests and Ensemble Classifiers: The Wisdom of the Crowd’’).

## Why these models and why not other models

The selection of Random Forest, SVM and CNN is motivated by their compatibility with the image classification task. Other models like pure regression or Lasso are either not designed for classification or are too simplistic to capture the necessary detail in image data effectively. Particularly:

### Why Random Forest?

Random Forest is an ensemble learning method effective for classification tasks. It can handle high-dimensional data well and is less prone to overfitting, making it a robust choice for image classification. Being an ensemble of decision trees, it benefits from the collective decision-making of multiple estimators, which generally improves its accuracy and robustness over a single decision tree (AIML.com, 2023).

### Why SVM?

SVM is particularly adept at handling high-dimensional data. It works well for binary and multiclass classification problems and can model complex boundaries thanks to its use of kernel functions. It is effective in cases where the number of dimensions exceeds the number of samples, which can be advantageous in image recognition tasks, albeit MNIST is quite balanced in this regard (GeeksforGeeks, 2023).

### Why CNN?

CNN is specifically designed for tasks involving image data. It can capture the spatial hierarchy of features in an image (like edges in lower layers, simple shapes in middle layers, and complex objects in higher layers), which is extremely beneficial for image classification. CNNs reduce the number of parameters through weight sharing and pooling, making them more efficient and less prone to overfitting compared to fully connected networks on image data (Datagen,‘’Convolutional Neural Network: Benefits, Types, and Applications’’).

### Why Not Regression Models or Lasso?

Regression models, including linear regression or logistic regression, are not ideal for image classification because they do not capture the spatial hierarchies in image data. They treat each pixel as an independent feature without considering the correlation between adjacent pixels. So linear regression is not suitable for classification at all since it's meant for predicting continuous outcomes, not discrete classes. Logistic regression can be used for classification but is generally less effective for complex image data like MNIST compared to more specialized approaches like CNNs or SVMs with appropriate kernels (Kumar, 2021; Morin, 2023).

Lasso is a regression analysis method that is typically used to create simpler, more interpretable models that perform feature selection. While one could apply Lasso regularization within the context of logistic regression for classification, the method is primarily aimed at reducing overfitting in regression models, not at capturing complex patterns in image data (Kavlakoglu, 2024; Quora,"Pros and Cons of Lasso Regression,").

## Streamlit's Role in Machine Learning Deployment

Streamlit is an open-source app framework specifically designed for machine learning and data science teams. It allows for the rapid creation of interactive, data-driven web applications, enabling users to interact with machine learning models and visualize their outputs effectively. Streamlit emerges as a potent tool for deploying machine learning models into accessible web applications, facilitating direct user interaction with the predictive systems. By integrating a model into a Streamlit application, users can intuitively interact with the model, inputting their own data and visualizing the results in real time, which is instrumental in demonstrating the practical utility and user-centric design of machine learning solutions (Makhijani, 2023).

# Method

The study utilized the MNIST dataset, acclaimed for its application in machine learning research for handwritten digit recognition. This dataset comprises 70 000 grayscale images, evenly distributed across ten digit classes. The preprocessing and analytical methodologies employed to evaluate the effectiveness of various machine learning models are detailed below.

## Data Acquisition and Preprocessing

The MNIST dataset, sourced from the OpenML repository, includes 60 000 training images and 10000 test images. The raw data from the MNIST dataset, fetched via the fetch\_openml function from the scikit-learn library, underwent several preprocessing steps to ready it for analysis. Initial preprocessing involved normalizing the pixel intensity values to a range of 0 to 1, using the MinMaxScaler. This step ensures that the model inputs have a uniform scale, which is crucial for effective model training and convergence. For CNN compatibility, the training and testing sets were reshaped into a 4-dimensional array to match the input shape required by Keras, maintaining the original 28x28 pixel size of the images (Mirz, 2023). A part of the code depicting the above is available in the appendices section below (see Appendix A).

## Model Implementation

Three distinct models were evaluated:

- Random Forest (RF): Implemented with 100 estimators, the RF model was trained on the flattened, normalized image data (see Appendix B).

- Support Vector Machine (SVM): Configured with a 'scale' gamma value, the SVM was also trained on the normalized dataset, demonstrating its capability in high-dimensional data classification (see Appendix C).

- Convolutional Neural Network (CNN): consisted of two convolutional layers, each followed by a max-pooling layer, a flattening layer and two dense layers, culminating in a softmax activation function for classification. This model was specifically designed to leverage the spatial hierarchy of pixel data in images, trained over 5 epochs with a batch size of 64 and validated on a 10% split of the training data (see Appendix D).

## Model Training and Evaluation

Each model underwent training using the preprocessed MNIST training data. Performance evaluation focused on accuracy metrics, calculated by comparing the models' predictions against the actual labels in the test dataset. The CNN model, in particular, demonstrated superior accuracy, highlighting its efficacy in image-based classification tasks.

## Application Deployment

The best-performing model, the CNN, was integrated into a Streamlit web application, designed to allow users to upload images of handwritten digits and receive predictive classifications in real-time. This deployment process involved saving the trained CNN model and confirming its availability for the application's use.

## Model Validation

In addition to accuracy measurements, the CNN model's predictions were visually inspected through a series of sample images from the test set, demonstrating the model's practical effectiveness in real-world digit classification tasks.

# Results and Discussion

## Results

The results of the evaluation of the three models selected are summarized in the table below. Then follows the analysis of these results.

|  |  |
| --- | --- |
| **Accuracy for different models** | |
| Random Forest (RF) | 0.9704 |
| Support Vector Machine (SVM) | 0.9792 |
| Convolutional Neural Network (CNN) | 0.9905999898910522 |

Table 1: Accuracy results for the three selected models.

The evaluation of the three machine learning models on the MNIST dataset for handwritten digit recognition yielded insightful findings. The Random Forest model achieved an accuracy of 97.04%, and the Support Vector Machine (SVM) model slightly outperformed it with an accuracy of 97.92%. However, the Convolutional Neural Network (CNN) demonstrated superior performance, reaching an accuracy of 99.06% on the test set. This marked difference highlights the CNN's robust capability in image-based classification tasks, leveraging its complex architecture to effectively capture and classify the nuanced patterns in handwritten digits.

The training process of the CNN model further underscores its efficiency and effectiveness. Over the course of five epochs, the model showed a steady increase in accuracy from 86.88% to an impressive 99.26% on the training set, with validation accuracy closely mirroring this upward trend. This progression indicates not only the model's ability to learn and adapt to the dataset but also its potential in generalizing well to unseen data. The training and validation losses concurrently decreased, reinforcing the model's learning efficacy (See Appendix E-1).

The architecture of the CNN, consisting of convolutional layers followed by max-pooling layers, a flattening step, and dense layers, was instrumental in this performance. With a total of 365,792 parameters, the model was able to learn detailed features from the image data, which is critical in distinguishing between the different digits. The distinction between trainable parameters (121,930) and non-trainable parameters underscores the tailored approach taken in the model's design, optimizing it specifically for the task at hand (See Appendix E-2 ).

## Discussion

The CNN's outstanding accuracy in this study serves as a testament to the potential of deep learning models in image recognition tasks. Compared to traditional machine learning algorithms like Random Forest and SVM, CNNs offer a more nuanced and effective method for processing and classifying image data. This is particularly evident in tasks requiring the identification of subtle patterns and variations, as is the case with handwritten digit recognition.

The implications of these findings are significant for the development of automated systems that require accurate and reliable digit recognition, such as postal mail sorting and bank check processing. The deployment of the CNN model within a Streamlit web application further demonstrated the practical applicability of these models, offering an interactive platform for real-time digit prediction and engaging user experience.

This study also highlights the importance of model architecture and training strategies in achieving high performance. The incremental improvements observed during the CNN's training process emphasize the need for careful consideration of model parameters, layer configurations, and optimization techniques to harness the full potential of deep learning in practical applications.

## Streamlit Application

Following the comprehensive evaluation of machine learning models for digit recognition and giving a particular emphasis on the Convolutional Neural Network (CNN), the project advanced into the application phase. This involved deploying the trained CNN model within a Streamlit web application, thereby transforming the model from a theoretical framework into a practical, user-interactive tool. The deployment process, the application's functionality and the application's results are described as follows:

### Deployment Process

Having chosen to utiliz the Streamlit framework in creating interactive web applications, the CNN model, which exhibited superior performance in recognizing handwritten digits, was integrated into the application to allow real-time digit prediction. This integration was facilitated by loading the pre-trained CNN model from a specified path, ensuring the application could leverage the model's predictive capabilities.

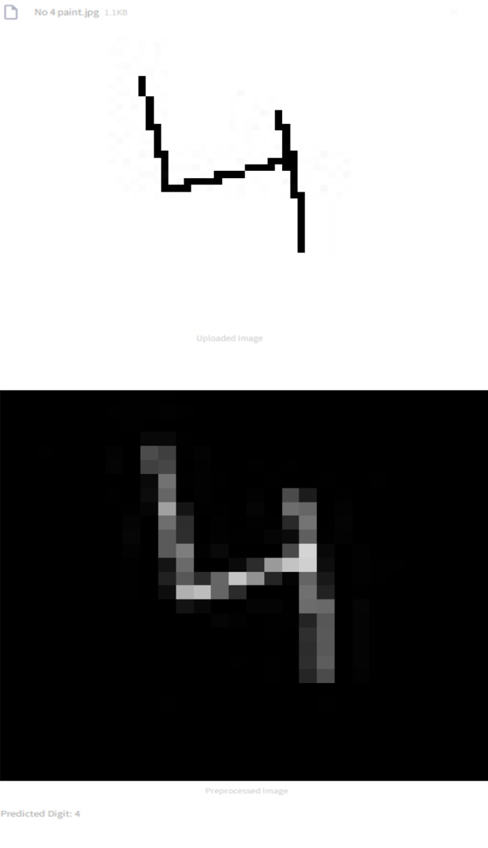
### Application Interface and Functionality:

The application titled "MNIST Digit Recognizer," invites users to upload images of handwritten digits. Upon uploading an image, the application employs a preprocessing function to convert the image to grayscale, resize it to the required 28x28 pixels, invert colors, normalize the pixel values and reshape it to match the input format of the CNN model. This preprocessing is crucial for maintaining consistency with the data preparation steps used during the model's training phase.

Once the image is preprocessed, the application utilizes the CNN model to predict the digit represented in the image. The prediction process continues in the display of the uploaded image alongside its preprocessed version, providing users with visual feedback on how the image was prepared for prediction. Finally, the application reveals the predicted digit, offering an interactive experience that demonstrates the model's real-world applicability (See Appendix F-1,F-2).

### Results in practice

Below is an image of a predicted number following the process as described above:



It should be noted that the numbers are drawn in the paint program of the computer, using different fonts and pencil styles (like natural pencil, calligraphy pen, marker, watercolor brush, calligraphy brush and size 1px,3px), making the number hard to read even in its original form. But despite this, a successful prediction of all the digits from 0 to 9 was achieved. This procedure was performed for all digits from 0 to 9 with successful prediction and the images are presented in the Appendix G (1-10).

### Significance and Broader Implications of the Application

Deploying the CNN model within a Streamlit application exemplifies the transformative journey from data science research to practical application. This venture not only validates the model's capability in a real-world context but also democratizes access to sophisticated machine learning predictions, enabling users without technical expertise to benefit from cutting-edge research.

The application symbolizes a significant stride towards integrating machine learning models into everyday tools and services, providing a blueprint for future projects aimed at bridging the gap between theoretical research and practical utility. It serves as a compelling example of how machine learning can be made approachable and useful for a wide audience, thereby enhancing its impact and relevance in society (Sharma, 2022).

# Conclusion

This study compared several machine learning models, including Convolutional Neural Networks (CNNs), on the MNIST dataset for handwritten digit recognition. The investigation revealed CNNs to be superior, achieving an accuracy of 99.06%, outperforming traditional models like SVM and RF. This highlights CNNs' effectiveness in image classification, attributable to their deep learning architecture designed to capture complex patterns in data.

Addressing the research question, it's clear that CNNs stand out in development, evaluation, and real-world application for digit recognition tasks. Their architectural advantages enable more accurate and generalizable predictions compared to SVM and RF models. Furthermore, the successful deployment of a CNN in a Streamlit web application demonstrated the practical potential of machine learning models to enhance user engagement and application accessibility.

In summary, CNNs' superior performance in this study underscores their significant potential in digit recognition and beyond. This underscores the value of continuous exploration and application of deep learning models in addressing complex challenges across various fields.

# Theoritical Questions

1.In machine learning, the data is typically split into three sets: training, validation, and test. The purpose of each is:

Training set: This is used to train the machine learning model. The model learns from this data by adjusting its parameters to minimize the error in its predictions.

Validation set: This data set is used to tune the hyperparameters of the model and provide an unbiased evaluation of a model fit on the training dataset while tuning the model's architecture. It helps in preventing overfitting.

Test set: This set is used to provide an unbiased evaluation of a final model fit on the training dataset. It gives us the performance metric of the model after it has been trained and validated (Wikipedia, 2024).

2. Without an explicit validation set, Julia can use several approaches to evaluate and choose the best model among the three she has trained. One of them is:

Cross-validation: Julia can use k-fold cross-validation on her training data. This technique divides the training data into k smaller sets (or folds). The model is trained on k-1 of these folds and validated on the remaining part, repeating this process k times (using each fold once as a validation set). The model's performance is then averaged over the k iterations to give an estimation of its effectiveness (Wikipedia contributors, 2024).

3. A regression problem in machine learning is one where the objective is to predict a continuous outcome variable (dependent variable) based on one or more predictor variables (independent variables). The goal is to model the relationship between the predictor and outcome variables so that we can make accurate predictions for new, unseen data.

Examples of regression models include:

Linear Regression: This model assumes a linear relationship between the dependent and independent variables. It's often used as a baseline model in many predictive tasks.

Polynomial Regression: An extension of linear regression where the relationship between the independent variable and the dependent variable is modeled as an n-thdegree polynomial. It's useful when the relationship is not linear.

Ridge Regression: An extension of linear regression that includes a regularization term to prevent overfitting, especially useful when dealing with multicollinearity or when the number of predictors exceeds the number of observations.

Lasso Regression: Similar to Ridge Regression, Lasso also includes a regularization term, but it can shrink some coefficients to zero, effectively performing variable selection.

Random Forest Regression: A type of ensemble learning method where multiple decision trees are constructed during the training phase and output the mean prediction of individual trees for regression tasks.

Potential application areas for regression models include:

Real Estate: Predicting house prices based on features like location, size, and number of bedrooms.

Finance: Forecasting stock prices or market trends based on historical data and various economic indicators.

Healthcare: Estimating medical costs or predicting patient outcomes based on their clinical parameters.

Energy: Predicting electricity consumption or solar power generation based on weather data and other relevant factors (Appier, 2024).

4. The RMSE (Root Mean Square Error) is a standard way to measure the error of a model in predicting quantitative data. It represents the square root of the average squared differences between the predicted values and the observed actual outcomes over the test set.

Interpreting RMSE:

Scale-dependent: The RMSE value is in the same units as the predicted and observed values, making it relatively easy to relate to the scale of the data.

Sensitive to outliers: RMSE gives a relatively high weight to large errors because it squares the errors before averaging, so it is very sensitive to outliers.

Zero means perfect fit: An RMSE of zero indicates no error, meaning the predictions are perfect, which is almost never the case in real-world scenarios.

RMSE is used for: Evaluating models: To compare the performance of different regression models; lower RMSE values indicate a better fit to the data.

Hyperparameter tuning: During the model training process, to find the best hyperparameters that minimize RMSE.

Model validation: To validate the model's predictive power, often in conjunction with cross-validation techniques (Olumide, 2023).

5. A classification problem is a type of supervised learning problem where the task is to predict the discrete class labels of new instances, based on a training dataset containing instances with known labels. Examples of classification models include:

Logistic Regression: Despite the name, it is a classification algorithm, often used for binary classification problems (e.g., spam or not spam).

Decision Trees: A model that uses a tree-like graph or model of decisions and their possible consequences; it's intuitive and easy to visualize.

Support Vector Machines (SVMs): A powerful classifier that works well on a wide range of classification problems, including those with complex boundaries.

Random Forest: An ensemble method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees.

Neural Networks: Versatile and powerful models that can be used for a variety of classification tasks, especially complex ones with large datasets.

Potential application areas for classification models include:

Email filtering: Classifying emails as spam or not spam.

Medical diagnosis: Diagnosing diseases based on patient symptoms and test results.

Credit scoring: Assessing the risk of loan applicants.

Image recognition: Identifying objects within images.

Sentiment analysis: Determining whether the sentiment of a piece of text is positive, negative, or neutral (Kumar, 2023).

A "Confusion Matrix" is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa).

It includes true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

The diagonal elements represent the number of points for which the predicted label is equal to the true label.

The off-diagonal elements are those that were labeled incorrectly by the algorithm.

Metrics such as accuracy, precision, recall, and F1-score can be derived from the confusion matrix (Narkhede, 2018).

6. The K-means algorithm is an unsupervised machine learning model used for clustering analysis. The goal of K-means is to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

Here's how the K-means algorithm works:

-Select k initial centroids (the means), usually chosen randomly.

-Assign each data point to the closest centroid based on some distance measure, creating k clusters.

-Recalculate the centroids as the mean of all points assigned to that cluster.

-Repeat steps 2 and 3 until the centroids no longer change significantly, which means the model has converged.

Example of an application:

Customer Segmentation: Businesses often use K-means to segment their customers based on purchase history, behaviors, or demographics to target marketing campaigns more effectively. For example, a retail chain could use K-means to categorize shoppers into different groups and tailor specific promotions to each group's shopping habits (Frost, 2024).

7. Ordinal encoding, one-hot encoding, and dummy variable encoding are methods used to convert categorical variables into numerical form so that they can be provided to machine learning algorithms.

Ordinal Encoding: This method converts categorical variables into integer codes ranging from 0 to (number of categories - 1). The numbers are assigned in an ordered manner based on the category. This method assumes an order or hierarchy in the categories.

Example: Imagine a variable "Education Level" with three categories: "High School," "Bachelor's," and "Master's." They could be encoded as 0, 1, and 2, respectively.

One-Hot Encoding: Each category value is converted into a new binary column and assigned a 1 or 0 (notation for true/false) across all columns. This representation does not assume any order of the categories and prevents the model from assuming a natural ordering between categories that may not exist.

Example: If we have a "Color" variable with three categories: "Red," "Green," and "Blue," one-hot encoding will create three variables "isRed," "isGreen," and "isBlue." A red item will be coded as [1, 0, 0], green as [0, 1, 0], and blue as [0, 0, 1].

Dummy Variable Encoding: Similar to one-hot encoding, dummy variables create a binary column for each category of a categorical variable. The key difference is that it creates N-1 variables to avoid the "dummy variable trap" (perfect multicollinearity), which is redundant and can cause issues in regression models.

Example: Using the "Color" variable from above, dummy encoding (assuming "Red" is the reference category) would create two variables: "isGreen" and "isBlue." A red item would be coded as [0, 0], green as [1, 0], and blue as [0, 1] (Prgomet, 2024).

8. We know that categorical data can be either ordinal or nominal:

Nominal data are categories without an intrinsic order. In Göran's view, color names like {red, green, blue} are typically considered nominal since they don't have a natural order.

Ordinal data, on the other hand, have a clear order or ranking. Julia’s example implies a ranking or order of desirability with respect to the color of a shirt, making it ordinal in that specific context.

Julia is highlighting the important point that the designation of data as nominal or ordinal can depend on how the data is used or interpreted. While colors are inherently nominal, they can take on ordinal properties if they are used in a context where there is a ranking or order.

9. Streamlit is an open-source Python library that is used to create web applications quickly and with minimal coding. It's specifically designed for machine learning and data science professionals to turn data scripts into shareable web apps. Key features of Streamlit include:

Ease of use: Streamlit apps are created with Python scripts. It requires no callbacks or HTML knowledge, which simplifies the app development process.

Rapid prototyping: It provides a fast way to build interactive and aesthetically pleasing user interfaces for Python programs.

Interactivity: Widgets like sliders, buttons, and text input allow users to interact with the app and change the output or visualizations in real-time.

Data Science Integration: Easily integrates with Pandas, NumPy, Matplotlib, Plotly, and other data science libraries to display data and results (Młynarek, 2024).

Streamlit can be used to:

Visualize Data: Quickly create interactive plots and graphs to explore and present data.

Build ML Tools: Develop custom machine learning tools, from model training interfaces to results analysis dashboards.

Create Data Applications: Turn analyses into interactive web apps that can be shared with colleagues or the world (<https://www.datacamp.com/tutorial/streamlit>, <https://www.youtube.com/watch?v=ggDa-RzPP7A&list=PLgzaMbMPEHEx9Als3F3sKKXexWnyEKH45&index=12>).

# Self-evaluation

1. Challenges

I faced two main challenges: mastering CNN complexities for digit recognition and deploying the model in a Streamlit app. The most challenging was mastering the complexities of Convolutional Neural Networks (CNNs) for digit recognition. The intricate architecture and the need for precise data preprocessing required a deep understanding of both theoretical concepts and practical implementation. To overcome this, I dedicated time to studying relevant literature and online resources, and conducted extensive experiments to observe the impacts of various configurations on model performance. Especially utilizing Streamlit documentation for practical implementation enhanced my problem-solving skills.

1. Proposed Grade

I believe my work merits a high grade for several reasons. First of all, the project successfully achieved its objective of comparing machine learning models, particularly highlighting the superiority of CNNs in handwritten digit recognition, analyzing the rigorous methodology, from data preprocessing to model evaluation and demonstrating a comprehensive understanding and application of machine learning principles. Then I spent many hours working on it and I think I covered what this report should have covered. However, because beyond the successful outcome of this research work I believe that there will be errors in the structure, the analysis or even possible omissions, my grade would say that I expect it to be somewhere between G and VG.

1. Note to Antonio

Antonio,

Thank you for your invaluable guidance throughout this project and lections. Your support was crucial in overcoming challenges and deepening my understanding of machine learning supplying us with such rich and valuable material. This experience has sparked a greater interest in the field, for which I am sincerely grateful.

# Appendix A: Data Acquisition and Preprocessing

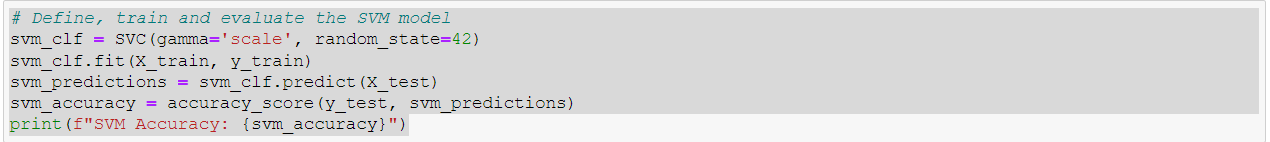
# 

# 

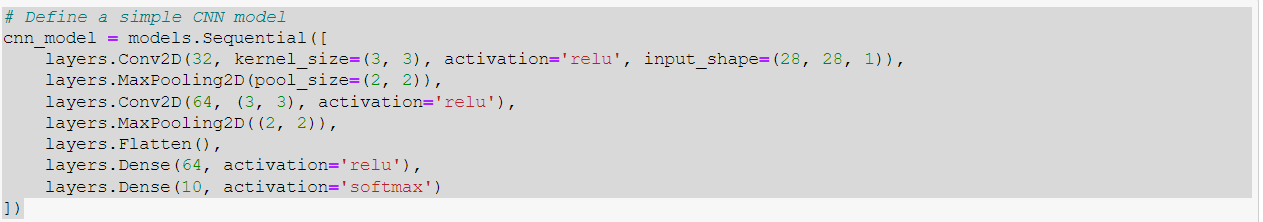
# Appendix B: Random Forest Model

# 

# Appendix C: SVM Model



# Appendix D: CNN Model



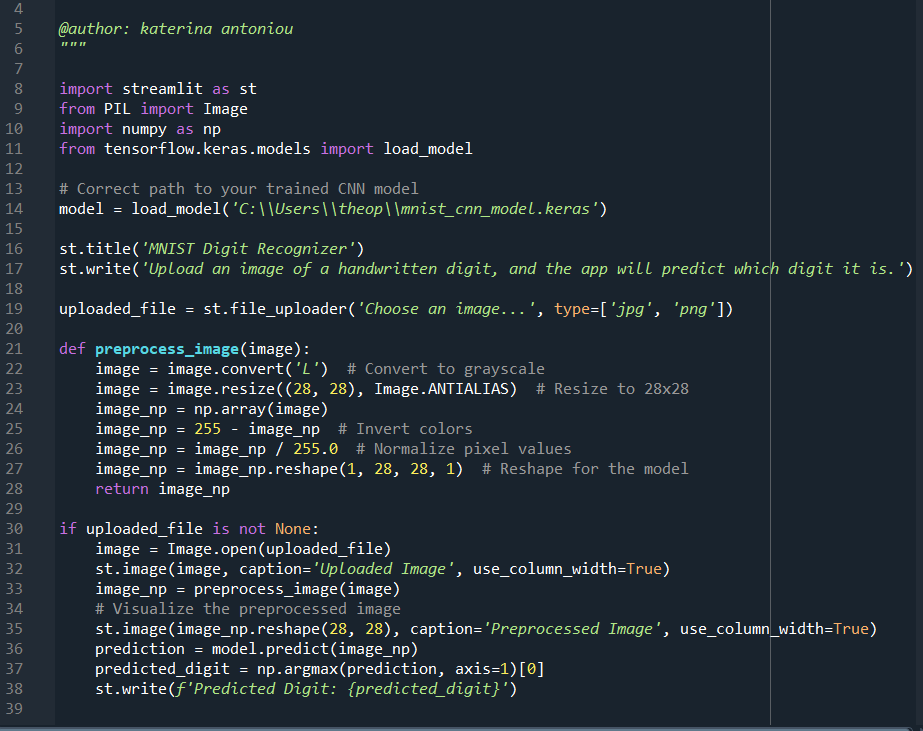
# Appendix E-1:CNNs Results

# 

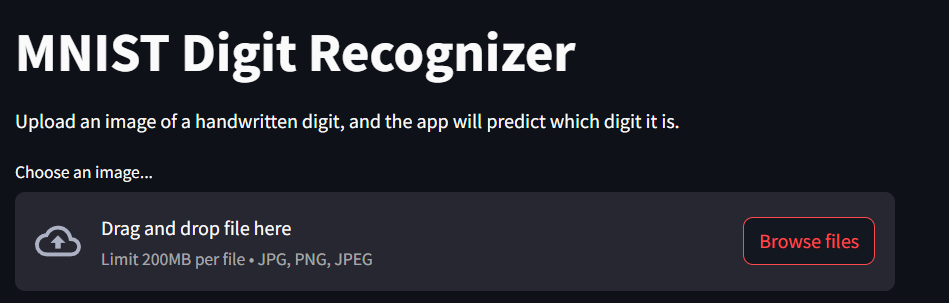
# Appendix E-2: CNNs Results

# 

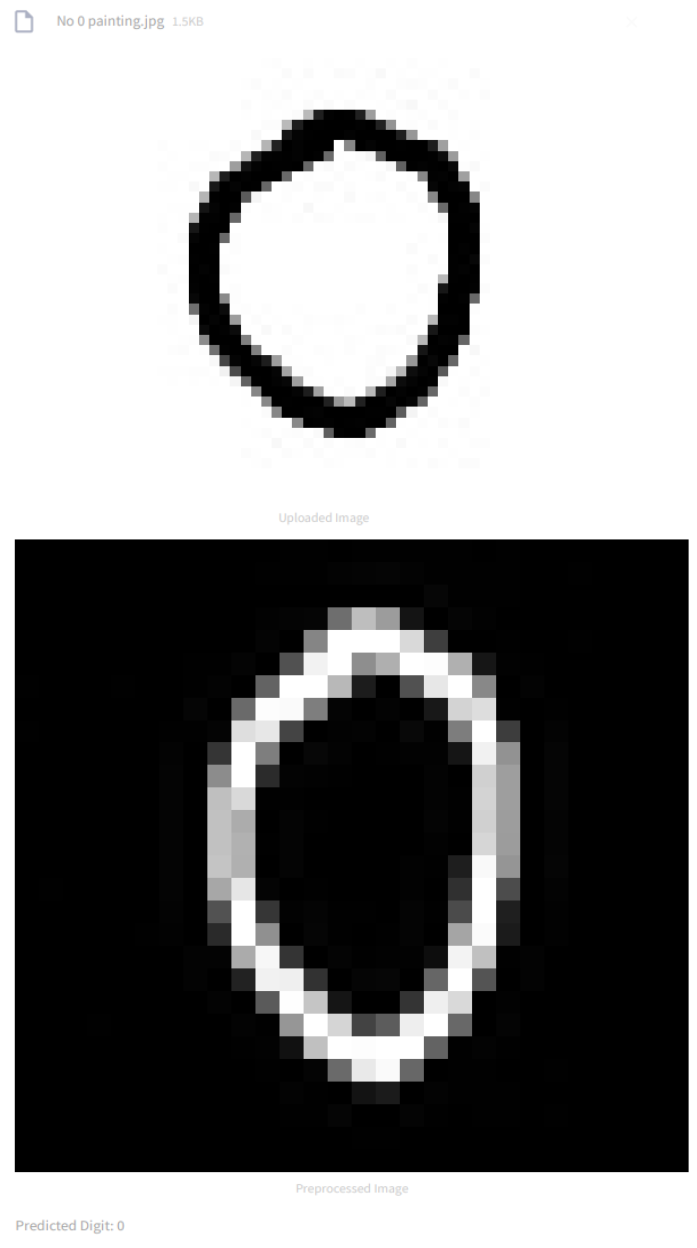
# Appendix F-1: Preview of the code in Spyder

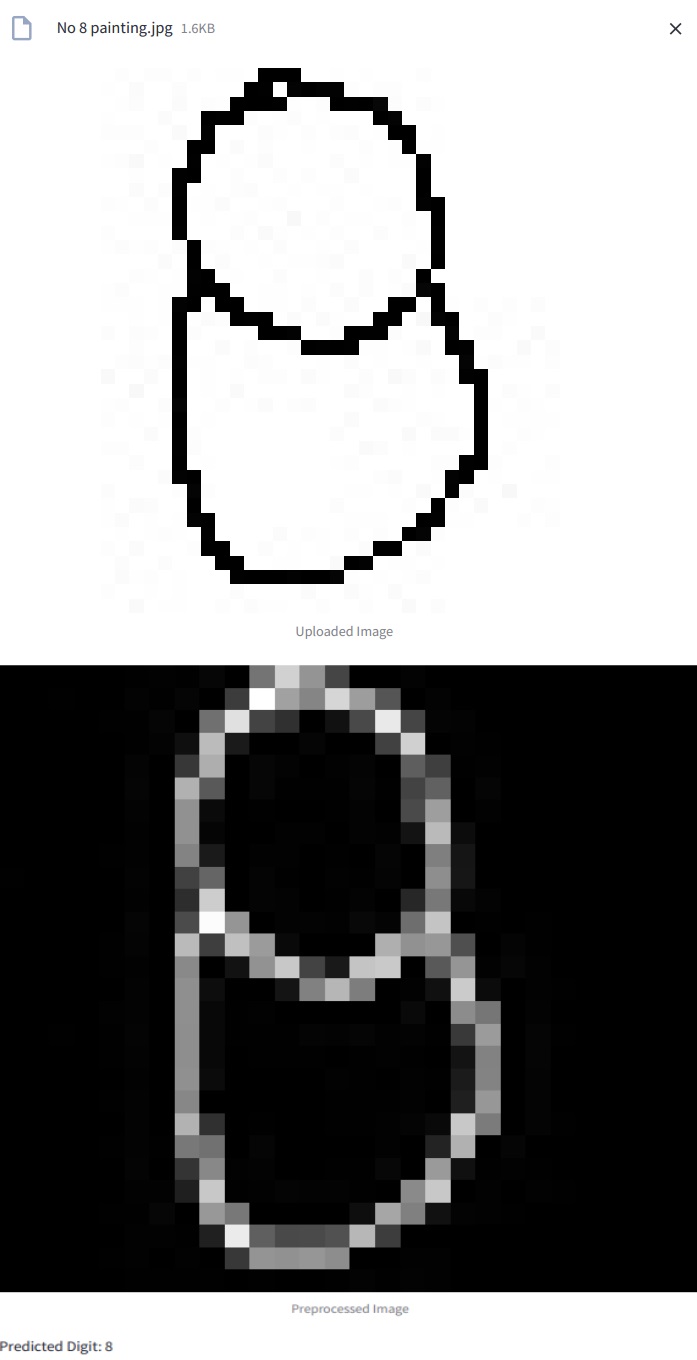
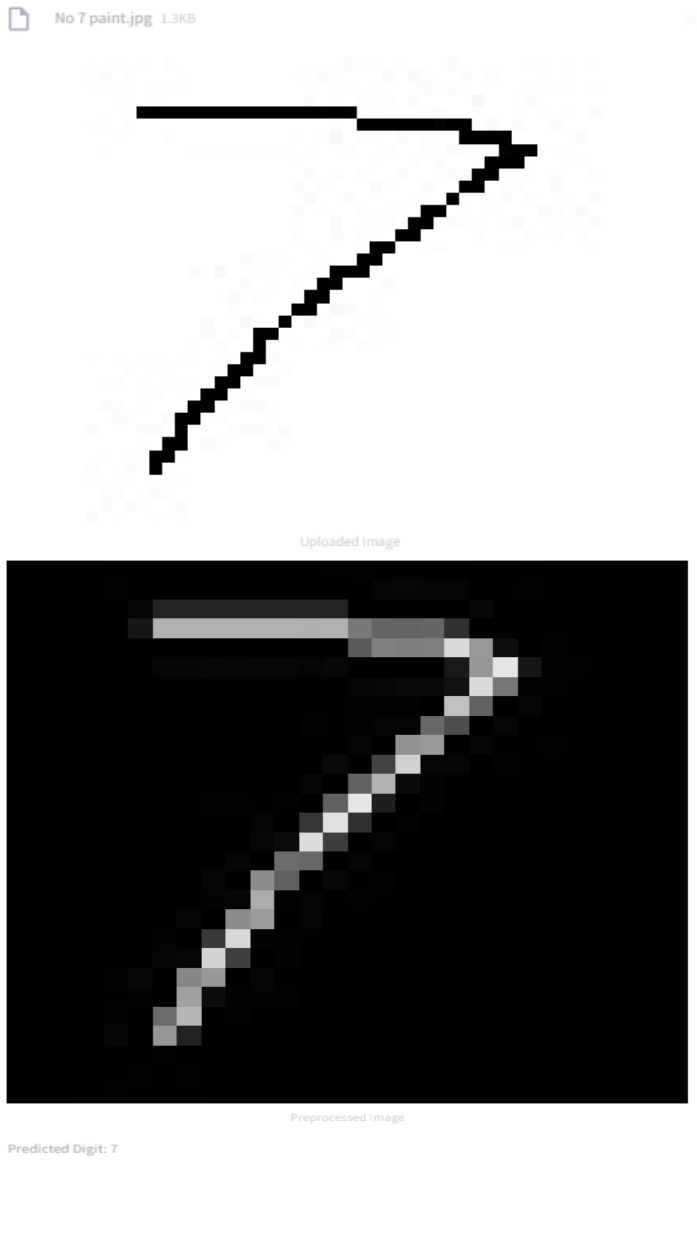
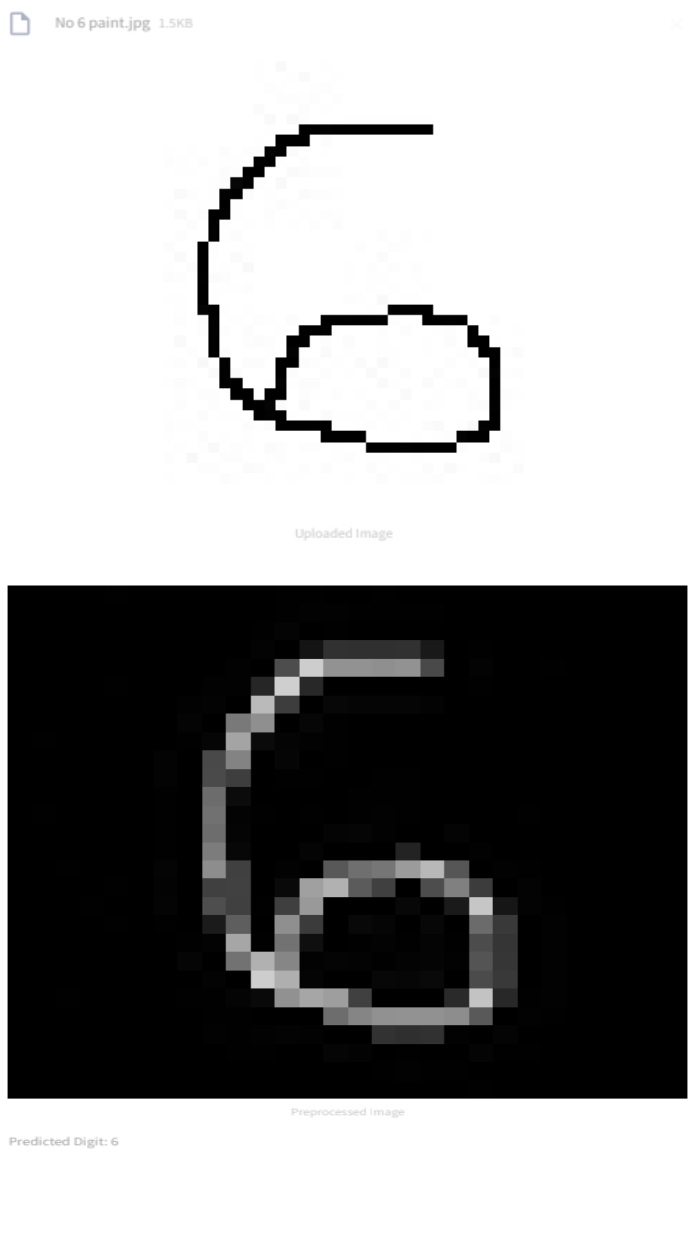
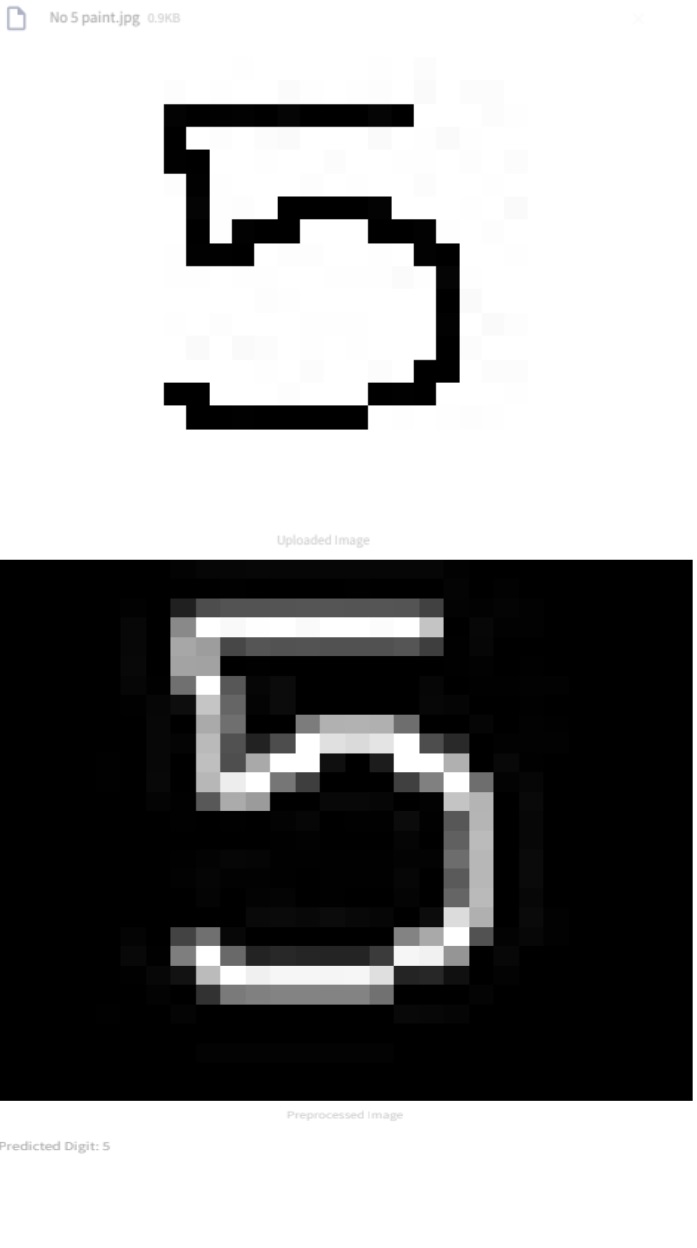
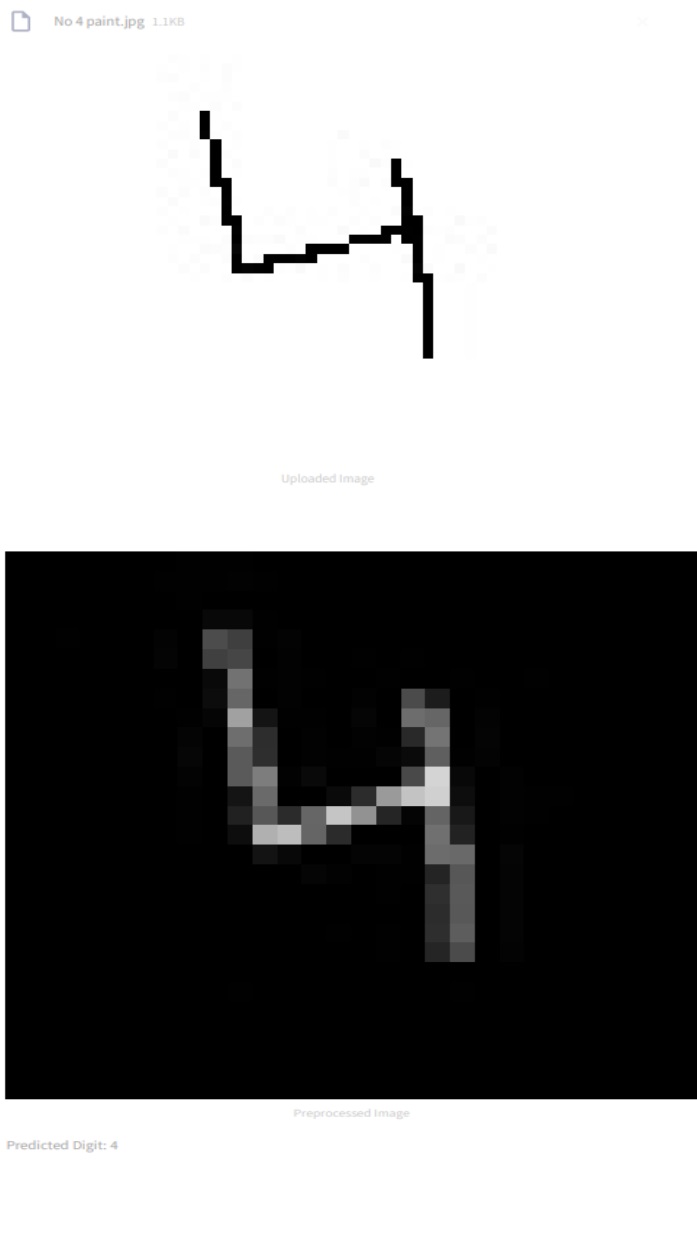
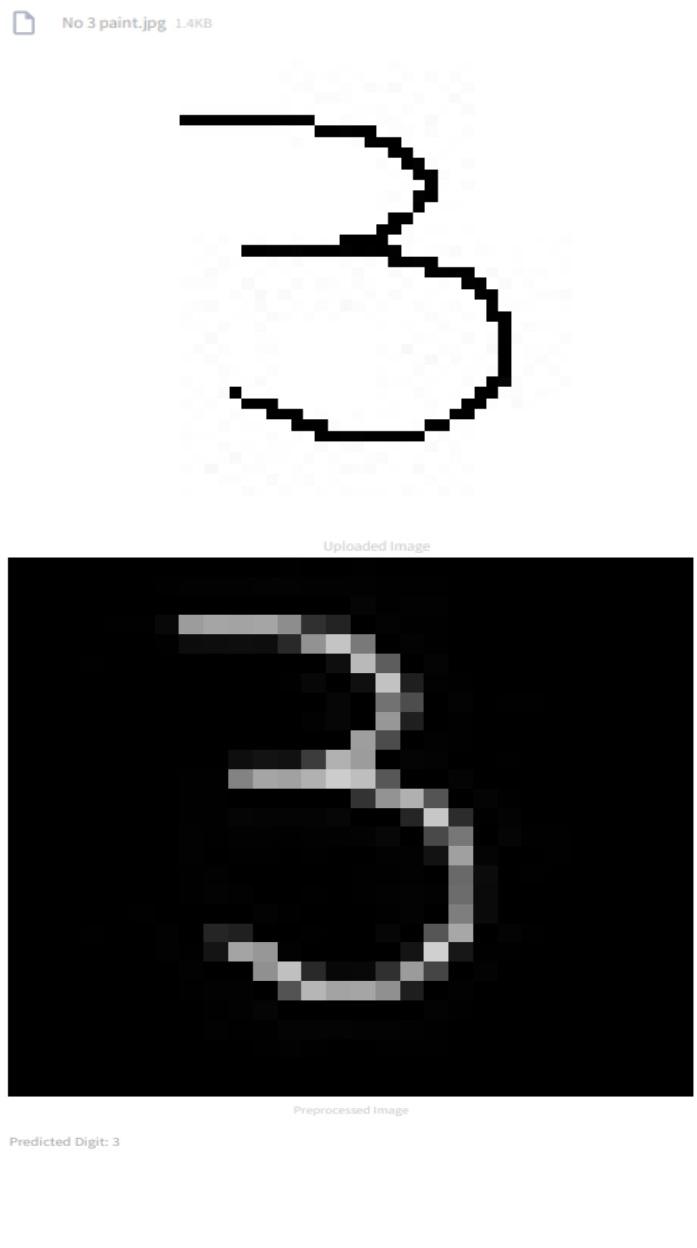
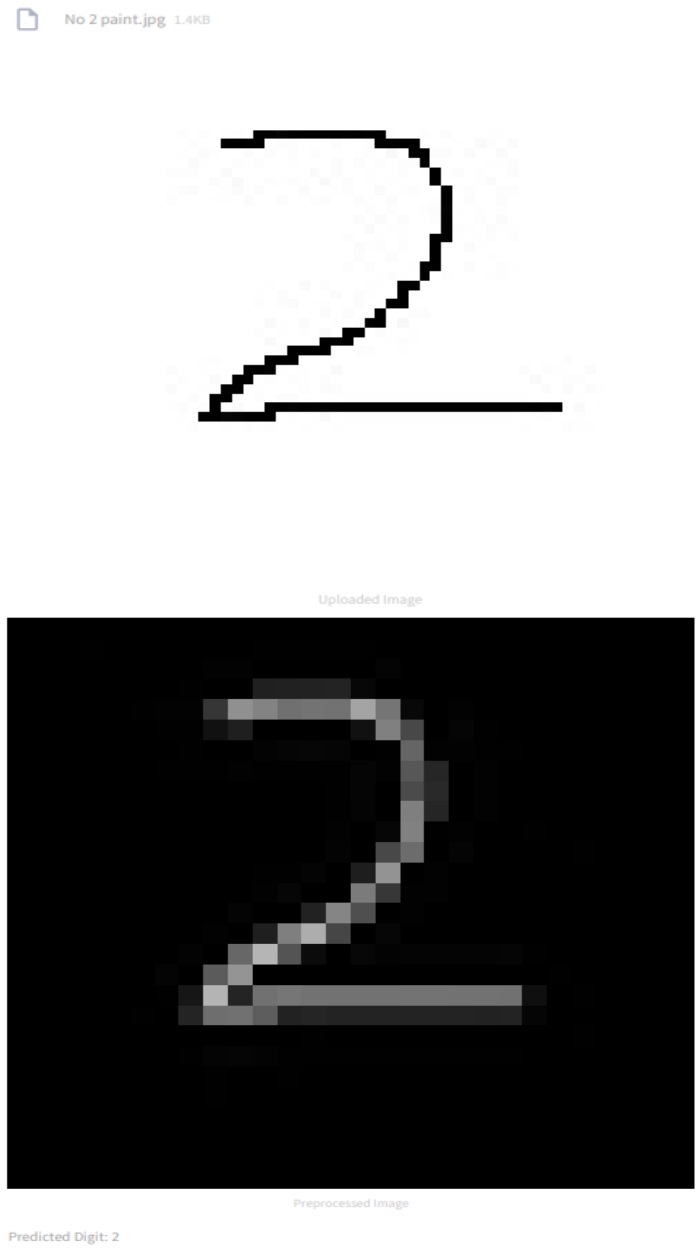
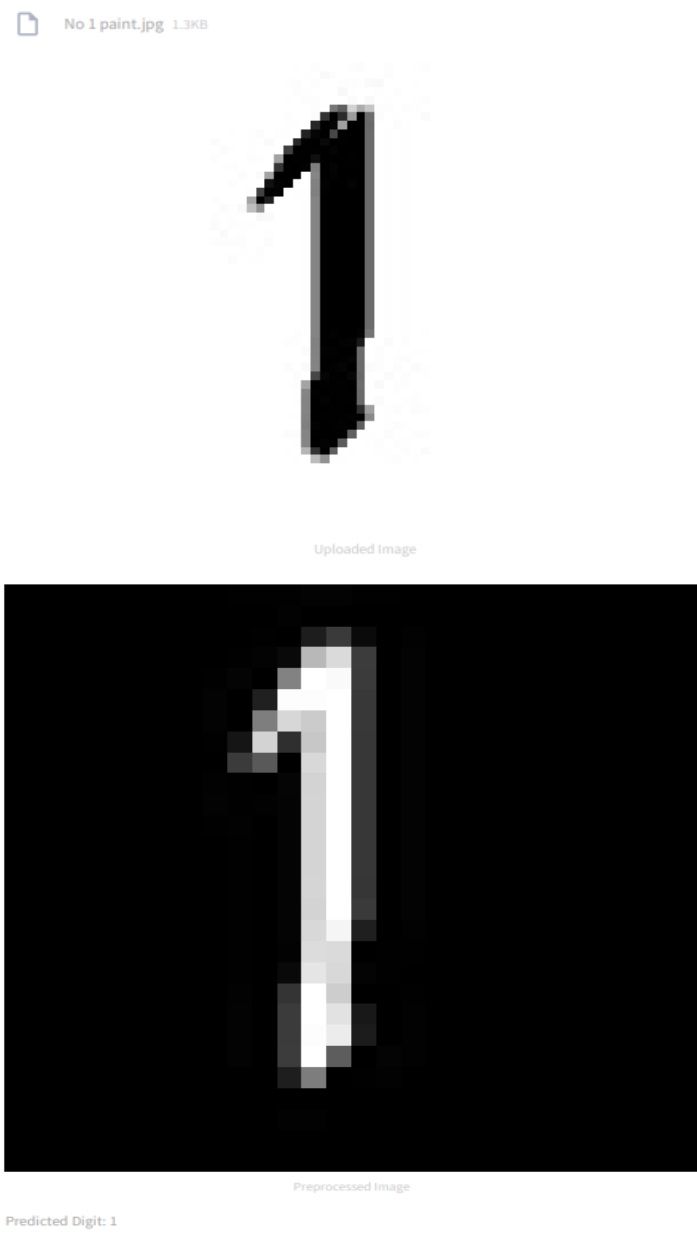


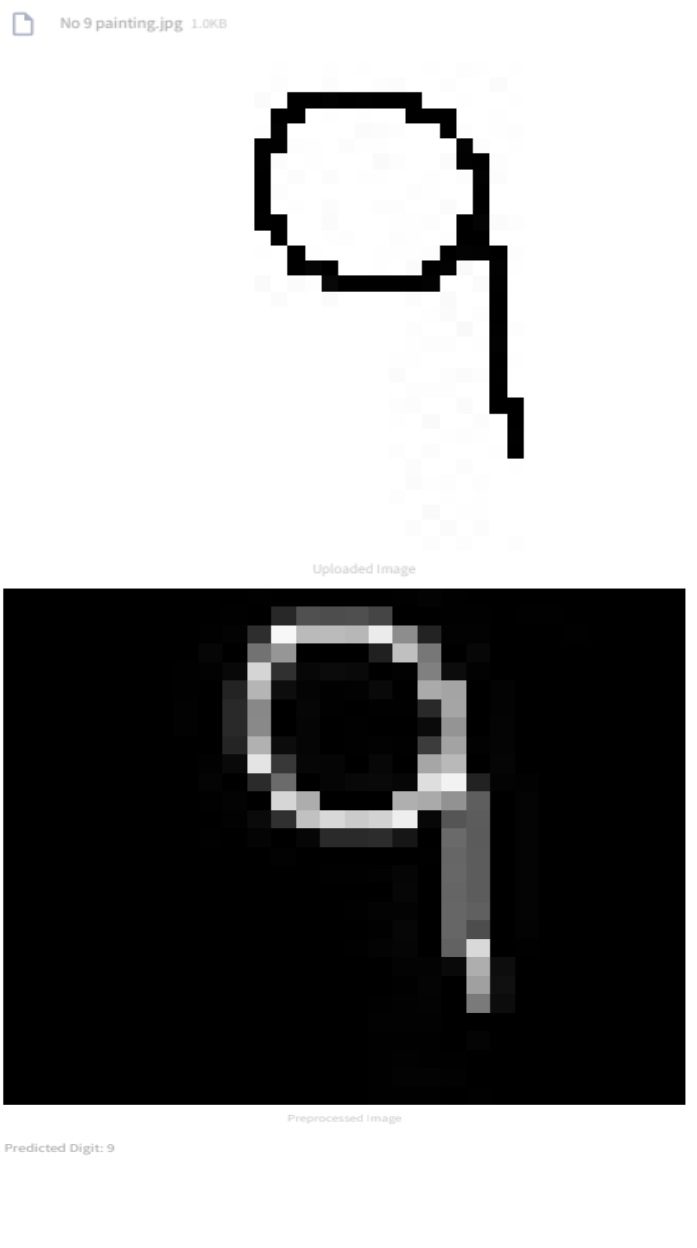
Appendix F-2: MNIST Digit Recognizer application for image uploading



# Appendix G 1-10: Uploaded images, preprocessed and predicted digits







# References

1. AIML.com.(2023).Advantages and Disadvantages of Random Forest. Retrieved from https://aiml.com/what-are-the-advantages-and-disadvantages-of-random-forest/

2.Appier. (2024). 5 Types of Regression Analysis And When To Use Them. Retrieved from https://www.appier.com/en/blog/5-types-of-regression-analysis-and-when-to-use-them

3.Datagen. Convolutional Neural Network: Benefits, Types, and Applications. Retrieved from https://datagen.tech/guides/computer-vision/cnn-convolutional-neural-network/

4.Datagen.Image Classification Using CNN: Introduction and Tutorial. Retrieved from https://datagen.tech/guides/image-classification/image-classification-using-cnn/#:~:text=CNNs%20can%20learn%20to%20recognize,data%20to%20detect%20specific%20patterns.

5.Ebrary.net. Statistical modeling and machine learning for molecular biology Random Forests and Ensemble Classifiers: The Wisdom of the Crowd. Retrieved from https://ebrary.net/60412/computer\_science/random\_forests\_ensemble\_classifiers\_wisdom\_crowd

6.Frost, J. (2024). What is K Means Clustering? With an Example. Statistics By Jim. Retrieved from https://statisticsbyjim.com/basics/k-means-clustering/

7.Géron, A. (2019). Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.). O'Reilly Media.

8.GeekforGeeks.(2023).Image classification using Support Vector Machine (SVM) in Python. Retrieved from https://www.geeksforgeeks.org/image-classification-using-support-vector-machine-svm-in-python/

9.Kanade, V. (2022). What Is a Support Vector Machine? Working, Types, and Examples. Spiceworks. Retrieved from https://www.spiceworks.com/tech/big-data/articles/what-is-support-vector machine/#:~:text=A%20support%20vector%20machine%20(SVM)%20is%20a%20machine%20learning%20algorithm,classes%2C%20labels%2C%20or%20outputs. (2022, September 29)

10.Kavlakoglu, E. (2024). Apply lasso regression to automate feature selection. IBM Developer. Updated March 6, 2024. Retrieved from https://developer.ibm.com/tutorials/awb-lasso-regression-automatic-feature-selection/ . (2024, January 29)

11.Kumar, A. N. (2021). Why Linear Regression is not suitable for classification? Medium. Retrieved from https://medium.com/analytics-vidhya/why-linear-regression-is-not-suitable-for-classification-cd724dd61cb8 . (2021, January 27)

12.Kumar, A. (2023). Classification Problems in Machine Learning: Examples. Analytics Yogi. Reimagining Data-driven Society with Data Science & AI. Retrieved from https://vitalflux.com/classification-problems-real-world-examples/ . (2023, November 28).

13.Makhijani, C. (2023). Machine Learning Model Deployment as a Web App using Streamlit. Medium. Retrieved from https://charumakhijani.medium.com/machine-learning-model-deployment-as-a-web-app-using-streamlit-4e542d0adf15 .(2023, November 7)

14.Makariev, D. (2023). Exploring the Classic MNIST: A Benchmark for Machine Learning Models. Retrieved from https://www.makariev.com/blog/exploring-the-classic-MNIST-Benchmark-for-machine-learning-models/

15.Mirz. (2023). Handwritten Digit Recognition using Python. Medium. Retrieved from https://medium.com/@farbodmirzaei1/handwritten-digit-recognition-using-python-93155d751ed7 .(2023, July 6)

16.Młynarek, P. (2024). Data Visualization Features with Streamlit, Dash, and Panel. Part 2. Sunscrapers. Retrieved from https://sunscrapers.com/blog/streamlit-dash-panel-features-part-2/ . (2024, February 6).

17. Morin, L. (2023). Why is it difficult to use a linear regression model for the classification problems? Data Science Stack Exchange. Retrieved from https://datascience.stackexchange.com/questions/124230/why-is-it-difficult-to-use-a-linear-regression-model-for-the-classification-probability .(2023, October 24).

18. Naga Sai Nischal, K. S., Nithin Sai, G., Mathew, C., Gowda, C. S., & Chandrakala, B. M. (2020). A survey on Recognition of Handwritten ZIP Codes in a Postal Sorting System. International Research Journal of Engineering and Technology (IRJET), 7(3). Retrieved from https://www.irjet.net/archives/V7/i3/IRJET-V7I3842.pdf

19. Narkhede, S. (2018). Understanding Confusion Matrix. Towards Data Science. Retrieved from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62. (2018, May 9).

20. Olumide, S. (2023). Root Mean Square Error (RMSE): What You Need To Know. Arize. Retrieved from https://arize.com/blog-course/root-mean-square-error-rmse-what-you-need-to-know/ . (2023, August 8).

21. Prgomet, A. (2024). I8 [File in GitHub repository: machine\_learning\_ds23]. Retrieved from https://github.com/AntonioPrgomet/machine\_learning\_ds23/tree/main/l8

22. Quora.Pros and Cons of Lasso Regression. Retrieved from https://www.quora.com/What-are-the-pros-and-cons-of-lasso-regression

23. Sharma, P. (2022). Deploying machine learning models using Streamlit – An introductory guide to Model Deployment. Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2020/12/deploying-machine-learning-models-using-streamlit-an-introductory-guide-to-model-deployment/ . (2022, August 1).

24. Wikipedia. (2024). Cross-validation (statistics). In Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Cross-validation\_(statistics) . (2024, March 8).

25. Wikipedia. (2024). MNIST database. In Wikipedia. Retrieved from https://en.wikipedia.org/wiki/MNIST\_database . (2024, February 24)

26. Wikipedia. (2024). Training, validation, and test data sets. In Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Training,\_validation,\_and\_test\_data\_sets . (2024, January 26).