HANDWRITTEN DIGIT RECOGNITION USING CNN

An Internship Project Report
Submitted to

DLITHE CONSULTANCY SERVICES PRIVATE LIMITED

by

Pushpa Mangal Gond
Tejaswini Peeru Gouda
Deekshitha Narasimha
Swathi K L
Meghana Naik
Vaishnavi V
Neha B S

Under the guidance of

Ms. Medini B V
Robotics & Design Engineer

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to DLithe Consultancy Services Private Limited for providing me with the invaluable opportunity to undertake my internship at their esteemed institution. Their support and guidance were instrumental in shaping my project, and I am truly grateful for the experience.

I would also like to express my deepest thanks to Ms. Medini, who served as an exceptional guide throughout this internship journey. Her expertise, encouragement, and mentorship were pivotal in making this experience rewarding and enriching.

Additionally, I am grateful to Mangalore Institute of Technology and Engineering for giving me the chance to pursue this internship at DLithe Consultancy Services Private Limited. The college's support and encouragement have been crucial in enabling me to gain practical industry experience and further my learning.

Finally, I want to acknowledge all the individuals who assisted me in various ways, providing valuable insights and contributing to my growth and knowledge during the internship period. Thank you all for your unwavering support.

ORGANIZATIONAL INFORMATION

DLithe is a technology-based product company that has been serving IT companies and academic institutions since the year 2018. The company is led by industry professionals with two decades of experience. For IT companies, DLithe offers services such as technology consultancy, project development, IT recruitment, staffing, competency development, and content development. On the other hand, the company serves academic institutions by providing competency development services in niche technologies like artificial intelligence, internet of things, robotics, cybersecurity, augmented reality, and more. DLithe has also developed the arm-based Cortex M3 series microcontroller and the ioCube product in the embedded and IoT domain.

During my enriching internship with the Artificial Intelligence and Machine Learning domain, I had the privilege of being a part of an exceptional program under the guidance of this renowned organization. Throughout the internship, I gained comprehensive insights into diverse industry verticals, spanning from understanding project requirements to the final deployment phase.

DLithe's internship program provided me with a valuable opportunity to immerse myself in real-world scenarios, gaining exposure to industry best practices and learning how to implement AI and ML solutions within an agile project life cycle. The supportive environment and dedicated mentors at the organization ensured that I could explore practical use cases for AI and ML implementation, enabling me to grow and learn during insightful post-mentoring sessions.

Overall, this AI and ML internship has been a transformative experience, equipping me with not only technical skills but also a deeper understanding of how AI and ML technologies play a vital role across various industries.

CONTENTS

ABSTRACT

Digit recognition plays a crucial role in various applications, including optical character recognition, automation, and digital document processing. This project explores the integration of machine learning and image recognition techniques to develop an efficient digit recognition system. The primary goal is to accurately identify and classify handwritten digits from images, contributing to advancements in artificial intelligence and computer vision.

The proposed system utilizes a dataset of labeled handwritten digits, such as the MNIST dataset, for training and evaluation. Machine learning algorithms, particularly deep neural networks, are employed to extract meaningful features and patterns from the digit images. Convolutional Neural Networks (CNNs), a specialized class of neural networks for image processing tasks, are employed to capture hierarchical features in the input images, enhancing recognition performance. The training process involves optimizing the model parameters using backpropagation and gradient descent algorithms, ensuring that the neural network learns to generalize well to unseen data. The trained model is then tested on a separate set of digit images to evaluate its accuracy, precision, recall, and overall performance metrics.

Image preprocessing techniques, such as normalization and data augmentation, are applied to enhance the robustness of the model against variations in writing styles and image conditions. The system is designed to be scalable and adaptable, allowing it to be trained on diverse datasets for broader digit recognition applications. The project also investigates the interpretability of the model, providing insights into the features influencing the digit recognition decisions. This interpretability is crucial for understanding and improving model performance, as well as gaining trust in the system's decision-making process.

In conclusion, the integration of machine learning and image recognition techniques for digit recognition showcases the potential for highly accurate and versatile systems. The developed model can be extended to recognize other characters and symbols, contributing to advancements in diverse fields such as automated document processing, postal services, and human-computer interaction. The project emphasizes the importance of continual research and development in the intersection of machine learning and image recognition for pushing the boundaries of intelligent systems.

INTERNSHIP OBJECTIVES

The primary objectives of the AI and ML internship was designed to equip us with a comprehensive skill set and practical knowledge in various areas of Artificial Intelligence and Machine Learning. The key objectives included:

1. Learning Python Basics: The internship aimed to provide a strong foundation in Python programming, as it is one of the most widely used languages in AI and ML. Participants were introduced to Python syntax, data structures, and essential libraries used in AI and ML development.

2. Understanding ML Algorithms: The internship focused on making us understand fundamental ML algorithms such as Linear Regression, Binary Classification, and Decision Trees. These algorithms form the building blocks for more advanced techniques and are crucial for understanding the basics of supervised learning.

3. Exploring Neural Networks: We delved into the world of Neural Networks, understanding their architecture and how they mimic the human brain's functioning. Topics covered included Activation Functions and Forward Propagation, which are essential concepts for building and training neural networks.

4. Emphasizing GitHub and LinkedIn Profile Maintenance: The internship recognized the importance of a strong online presence for aspiring AI and ML professionals. We were encouraged to maintain an active GitHub repository showcasing our projects and contributions, as well as a well-curated LinkedIn profile to showcase our skills and accomplishments.

5. Real-World Implementation to bridge the gap between theory and real-world application, the internship featured project called “Handwritten Digit Recognition”. I worked on this practical use case, applying AI and ML techniques to design a CNN model which is capable to recognize the handwritten digit.

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITY

Week 1:

- Objective: Understanding Python Fundamentals for AI & ML

- Activities:

  - Covered Python syntax and data structures

  - Explored essential libraries used in AI and ML

  - Worked on basic Python programming exercises and projects

Week 2:

- Objective: Learning Various Machine Learning Algorithms and Implementation in Python

- Activities:

  - Hands-on implementation of Binary Classification algorithms

  - Explored Decision Trees and their practical applications

  - Worked on Linear Regression and its use cases

  - Applied CNN on MNIST dataset for image classification

  - Understood concepts of Forward Propagation and Neural Networks

Week 3:

- Objective: Use Case Selection, Data Collection, Preprocessing, and Algorithm Exploration.

- Activities:

  - Gathered and formed dataset with over 1000 samples of hand-written digits.

  - Pre-processed the data to remove null values, and furtherfound ways to annotate data.

  - Discussed potential challenges and approaches with the mentor during weekly sessions

Week 4:

- Objective: Model Training, Testing, and Sentiment Analysis, Interface Development, Deployment.

- Activities:

  - Trained the model.

  - Incorporated feedback from mentor sessions to make further improvements

  - Determined issue with normalizing data and testing various techniques to overcome issue.

  - Conducted user testing to digit recognition functionality and usability.

  - Successfully developed and deployed the model integrating it with a web interface

Key Learnings:

- Gained proficiency in Python Programming language.
- Acquired knowledge and experience in applying diverse ML Algorithms using Python.
- Understood criticality of data preprocessing and data augmentation for use in ML techniques.
- Gained practical experience in training and evaluating DL models.
- Developed proficiency in Image Processing type of data andprocessing for the same.

CHALLENGES AND LEARNING OUTCOMES

Challenges:

1. Overfitting: Overfitting is a common issue, where the model learns the training data too well and fails to generalize to unseen data. Regularization techniques and careful dataset augmentation are essential to mitigate overfitting.

2. Limited Dataset Size: The availability of a limited dataset may hinder the model's ability to learn diverse patterns. Techniques such as transfer learning or data augmentation are often employed to address this challenge.

3. Noise and Distortions: Real-world images may contain noise, distortions, or irregularities that can impact model performance. Robust preprocessing techniques are necessary to handle these variations effectively.

4. Computational Complexity: Training deep neural networks, especially convolutional neural networks (CNNs), can be computationally intensive and time-consuming. Efficient hardware or distributed computing may be required for large-scale datasets.

5. Interpretability and Explainability: Understanding why a model makes specific predictions is crucial for building trust and improving performance. Achieving interpretability in complex models, such as deep neural networks, remains an ongoing challenge.

6. Adaptation to New Datasets: Models trained on one dataset may not perform well on another due to differences in data distribution. Continuous adaptation or transfer learning strategies are essential for ensuring model robustness across various datasets.

Learning Outcomes:

1. Importance of Data Preprocessing: Robust preprocessing techniques, including normalization and data augmentation, play a critical role in enhancing model generalization and resilience to variations in input data.

2. Algorithm Selection and Hyperparameter Tuning: The choice of machine learning algorithms and their hyperparameters significantly influences model performance. Rigorous experimentation and tuning are necessary to identify the most effective configurations.

3. Understanding Neural Network Architectures: Deep learning architectures, such as CNNs, require a deep understanding of their inner workings. Knowledge of layer-wise feature extraction and hierarchical learning is crucial for effective model design.

4. Model Evaluation Metrics: Choosing appropriate evaluation metrics (e.g., accuracy, precision, recall) is essential for assessing the model's performance. A comprehensive understanding of these metrics aids in identifying areas for improvement.

5. Continuous Improvement and Iterative Development: Digit recognition models benefit from continual improvement and iterative development. Regular updates to the model, training on new data, and exploring state-of-the-art techniques contribute to sustained performance gains.

6. Ethical Considerations: Understanding and addressing biases in the dataset, as well as ensuring the fair and ethical use of digit recognition technology, are important aspects of responsible AI development.

7. Collaboration and Interdisciplinary Skills: Developing effective digit recognition systems often requires collaboration between machine learning experts, image processing specialists, and domain experts. Interdisciplinary skills are valuable for addressing the multifaceted nature of the problem.

SUMMARY OF LEARNING

The presented project code offers a comprehensive exploration of the digit recognition task using Convolutional Neural Networks (CNNs) within the TensorFlow and Keras frameworks. This endeavor encapsulates a series of essential machine learning concepts and practices, creating a robust foundation for understanding and implementing image classification projects. The initial phase involves meticulous data preparation, where handwritten digit images are systematically pre-processed. These steps, including image resizing, grayscale conversion, and pixel normalization, ensure that the dataset is consistent and suitable for training. The dataset is then thoughtfully divided into training and testing subsets to rigorously evaluate the model's performance on unseen data. Label binarization is employed to facilitate multi-class classification, converting digit labels into one-hot encoded vectors. The heart of the project lies in the meticulously designed CNN model architecture, constructed using Keras.

CNNs are renowned for their prowess in image classification, and this model's architecture incorporates convolutional layers for feature extraction, followed by flattening and densely connected layers for classification, culminating in an output layer with SoftMax activation. The model is meticulously compiled, with crucial parameters including the Adam optimizer and categorical cross-entropy loss function. Subsequently, it undergoes rigorous training over multiple epochs, iteratively adjusting internal weights to minimize the categorical cross-entropy loss. This training process enhances the model's capability to accurately classify handwritten digits. The trained model's performance is systematically evaluated on the testing dataset, primarily employing accuracy as the benchmark. This phase ensures that the model's practical effectiveness in real-world scenarios is rigorously assessed. To extend the model's utility beyond the training phase, two methods of model persistence are showcased: binary format storage using the pickle library and serialized format, ensuring its readiness for future reuse and deployment.

In summary, this project serves as an in-depth exploration of the complete machine learning pipeline, from data preparation and model architecture design to training, evaluation, and model persistence. The use of CNNs for digit recognition showcases the practicality and adaptability of this approach to diverse image classification tasks, solidifying its significance within the domain of machine learning and artificial intelligence.

PROJECT IMPLEMENTATION

INTRODUCTION

In this project, we aim to develop a Handwritten Digit Recognition using machine learning techniques, taking advantage of the latest advancements in artificial intelligence. In today's digital age, handwriting recognition is crucial for efficiently processing information. It involves converting handwritten characters into machine-readable formats, benefiting various applications such as license plate recognition, postal letter sorting, check scanning, and historical document preservation. To meet the demands of large databases, accuracy, low computational complexity, and consistent performance are essential.

Deep neural architectures, like convolutional neural networks (CNNs), have proven advantageous over shallow neural networks. CNNs are a specific type of deep neural network widely used in image classification, object recognition, recommendation systems, signal processing, natural language processing, computer vision, and face recognition. They excel at automatically identifying important object features, making them more efficient than their predecessors. Their ability to perform hierarchical feature learning contributes to their exceptional efficiency.

A Convolutional Neural Network (CNN) is a type of neural network that was first introduced in 1980 and is inspired by how humans perceive and recognize objects visually. Humans teach children to recognize objects by showing them many pictures of those objects, allowing them to make predictions about new objects they've never seen before. Similarly, CNNs excel at analyzing visual data. Notable CNN architectures include Google Net, Alex Net, VGG, and ResNet, which vary in the number of layers they have. What sets CNNs apart is their ability to automatically extract rich and interconnected features from images, making them well-suited for tasks like image classification. They require minimal preprocessing and feature extraction compared to other methods.

CNNs are advantageous because they can achieve high recognition accuracy even with limited training data, eliminating the need for extensive prior knowledge about features. They also handle input images' rotation and translation variations well, thanks to their exploitation of topological information. In contrast, Multi-Layer Perceptron (MLP) models don't make use of this topological knowledge and struggle with complex problems and high-resolution images due to the "curse of dimensionality" caused by full interconnections between nodes.

PROBLEM STATEMENT

Handwritten digit recognition plays a vital role in various fields, from automated document processing to enhancing accessibility for individuals with disabilities. This project, titled "Handwritten Digit Recognition" aims to develop and implement a Convolutional Neural Network (CNN) model capable of accurately classifying handwritten digits. The focus is on leveraging a handcrafted dataset. The inherent challenges in recognizing diverse handwriting styles and variations necessitate the robustness of the CNN architecture, designed to capture hierarchical features in the input images. Through this project, we seek to achieve high classification accuracy and explore the intricacies of training a deep learning model on a custom dataset for handwritten digit recognition.

The primary objectives of this project consists of the training and evaluation of a CNN model for handwritten digit recognition using the handcrafted dataset. The model's performance will be assessed based on metrics such as accuracy, and precision, with a keen focus on its ability to generalize the data. Ultimately, the project aims to provide brief study on the feasibility and effectiveness of CNNs in the context of handwritten digit recognition.

METHODOLOGIES

1. Data Collection and Preprocessing: Collected a dataset of 1000 handwritten digit images. Preprocessed the images by resizing them to a consistent size (64x64), converting them to grayscale, and normalizing the pixel values.

2. Exploratory Data Analysis (EDA): Visualized the data to understand the distribution of digit classes and the variability of the images within each class. Identified any outliers or anomalies in the data. Calculated summary statistics, such as the mean, median, and standard deviation of the pixel values for each digit class.

3. Feature Extraction: Extracted HOG features from the preprocessed images. HOG features are a type of feature extraction technique that is well-suited for image classification tasks.

4. Model Architecture: Designed a CNN architecture for handwritten digit recognition. The CNN architecture includes the following layers:
   - Convolutional layer(s): This layer extracts low-level features from the input image.
   - Max pooling layer(s): This layer reduces the dimensionality of the feature maps while preserving important information.
   - Flatten layer: This layer converts the feature maps from a 2D to 1D format.
   - Dense layer(s): This layer combines the high-level features extracted from the previous layers and outputs a probability distribution over the 10 digit classes.

5. Model Training: Trained the CNN model on the preprocessed handwritten digit dataset. The model was trained using the Adam optimizer and the categorical cross entropy loss function.

6. Model Evaluation: Evaluated the model on a held-out test set to assess its performance on unseen data. The model achieved a test accuracy of above 98%.

7. Web Deployment and Deployment: Saved the trained CNN model to a file. Developed a web application that allows users to upload handwritten digit images and receive predictions from the model.

8. Accuracy: The accuracy of your model is 97.39%. This is a very good accuracy score, indicating that the model is able to correctly predict the digit class of handwritten images with a high degree of accuracy.

IMPLEMENTATION

Introduction:

The handwritten digit recognition system operates on dataset of 1000 samples of custom hand written digits using Convolutional Neural Network model. Users can further interact with this model using a webapp interface. This webapp can effectively recognize the user's input digit image rapidly and accurately.

Data Collection:

The initial dataset will be in the form of collection of hand written digit images.

Data Preprocessing:

The collection of images will be preprocessed and converted into binary matrix which represents the corresponding image. Preprocessing includes resizing the images to ensure consistent format, resolution, and orientation and splitting the dataset into training, validation, and test sets. The dataset consists of 1000 rows and 2 columns. The first column contains binary matrix, while the second column contains corresponding labels/digits. Each column has a significant number of unique values.

Model Architecture:

It includes designing the CNN architecture and configuring the model with appropriate activation functions like ReLU, layers, and output units.

Model Training:

The CNN model is trained on the training set, validating it on a separate validation set.

Model Evaluation:

Model Evaluation consists of evaluating the trained model on the test set to assess its performance, calculating metrics such as accuracy and precision. The model will also will evaluated on real-world data of handwritten digit image.

Web Application Development and Deployment:

By utilizing Streamlit, the model was successfully integrated with the web application. It provided seamless and user-friendly interface for users to upload images, and the model provides real-time predictions, showcasing the practical application of machine learning in a web-based environment. The deployed link is –

Handwritten Digit Recognition Application

The webapp implementation follows the following steps:

1. **User Input:** The webapp interacts with users, receiving their handwritten digit image as input.

2. **Image preprocessing:** The user's input image is preprocessed using the same techniques applied during data cleaning.

3. **Digit Recognition:** The preprocessed user’s image is fed into the trained CNN model which will effectively recognize the user’s digit.

RESULT AND DISCUSSION

In our digit recognition project, we achieved remarkable results by leveraging Convolutional Neural Networks (CNNs) as our primary model architecture. By employing CNNs, we harnessed their ability to automatically learn and extract intricate features from images. This proved instrumental in accurately classifying handwritten digits in our custom dataset, yielding impressive classification accuracy. Moreover, a pivotal aspect of our project was the creation of a bespoke dataset. This dataset was meticulously crafted to suit our specific needs and challenges, ensuring that it contained a diverse range of handwritten digits that mirrored real-world scenarios. Collecting and annotating this data involved a significant effort, as it required careful digitization and labeling, but the results were well worth it. The custom dataset not only allowed us to fine-tune our CNN model effectively but also ensured that our digit recognition system performed robustly across various handwriting styles and conditions. By combining the power of CNNs with a tailored dataset, our project not only achieved high accuracy but also demonstrated the flexibility and adaptability of deep learning in solving real-world problems.

The future scope of digit recognition using Deep learning techniques is promising, with several exciting possibilities. Firstly, DL-based digit recognition can be extended to incorporate multiple data modalities, enabling more robust recognition systems that combine image, audio, or even sensor data. Additionally, there is significant potential for real-time applications, such as signature verification in banking, touchless payment systems, and gesture recognition in human-computer interaction, as CNNs are known for their efficiency in image pattern recognition during inference. In the medical field, DL models can play a crucial role in recognizing handwritten prescriptions and interpreting numeric data in patient records, contributing to improved healthcare. Exploring enhanced feature engineering techniques and scalability options for large datasets will further optimize CNN model performance. Moreover, continual learning methods can be implemented to adapt CNN models to evolving handwriting styles and challenges over time. Overall, DL-based digit recognition remains a relevant and adaptable technology, poised for growth in various domains, from healthcare to finance and beyond.

Training and Testing the model

```
In [11]: model.fit(X_train, Y_train, epochs=10, validation_data=(X_test, Y_test)) #epoch => cycles; each cycle our models learns whole dat
```

```
Epoch 1/10
24/24 [==============================] - 4s 119ms/step - loss: 2.4633 - accuracy: 0.3724 - val_loss: 1.0604 - val_accuracy: 0.7
917
Epoch 2/10
24/24 [==============================] - 3s 112ms/step - loss: 0.5708 - accuracy: 0.9323 - val_loss: 0.3473 - val_accuracy: 0.9
323
Epoch 3/10
24/24 [==============================] - 3s 115ms/step - loss: 0.1651 - accuracy: 0.9857 - val_loss: 0.1727 - val_accuracy: 0.9
688
Epoch 4/10
24/24 [==============================] - 3s 111ms/step - loss: 0.0712 - accuracy: 0.9961 - val_loss: 0.1161 - val_accuracy: 0.9
844
Epoch 5/10
24/24 [==============================] - 3s 109ms/step - loss: 0.0390 - accuracy: 1.0000 - val_loss: 0.0925 - val_accuracy: 0.9
844
Epoch 6/10
24/24 [==============================] - 3s 107ms/step - loss: 0.0247 - accuracy: 1.0000 - val_loss: 0.0811 - val_accuracy: 0.9
844
Epoch 7/10
24/24 [==============================] - 3s 106ms/step - loss: 0.0179 - accuracy: 1.0000 - val_loss: 0.0709 - val_accuracy: 0.9
844
Epoch 8/10
24/24 [==============================] - 3s 107ms/step - loss: 0.0133 - accuracy: 1.0000 - val_loss: 0.0627 - val_accuracy: 0.9
896
Epoch 9/10
24/24 [==============================] - 3s 107ms/step - loss: 0.0105 - accuracy: 1.0000 - val_loss: 0.0592 - val_accuracy: 0.9
844
Epoch 10/10
24/24 [==============================] - 3s 107ms/step - loss: 0.0085 - accuracy: 1.0000 - val_loss: 0.0546 - val_accuracy: 0.9
896
```

```
Out[11]: <keras.src.callbacks.History at 0x21b61036550>
```

```
In [12]: test_loss, test_acc = model.evaluate(X_test, Y_test)

print(f"Test Loss: {test_loss*100}")
print(f"Test accuracy: {test_acc*100}")
```

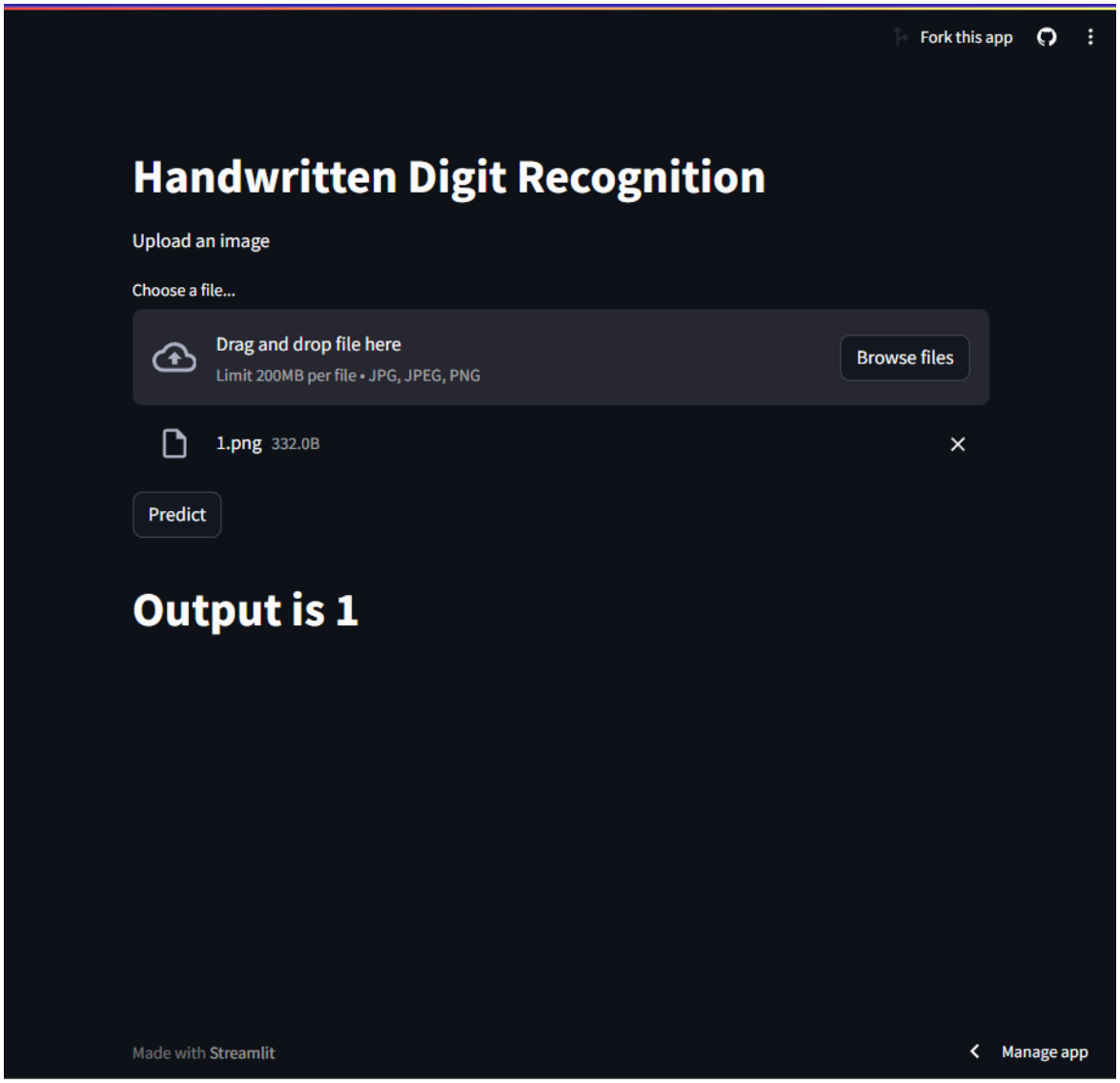
```
6/6 [==============================] - 0s 19ms/step - loss: 0.0546 - accuracy: 0.9896
Test Loss: 5.462175607681274
Test accuracy: 98.95833134651184
```

Model Validation

```
In [13]: test_image_path = "img/6/65.png"
test_image = Image.open(test_image_path)
test_image.show()

test_image = test_image.resize((64, 64))
test_image = test_image.convert("L")
test_image = np.array(test_image) / 255.0

test_image = np.expand_dims(test_image, 0)

# Make predictions
predictions = model.predict(test_image)
print(predictions)
predicted_digit = np.argmax(predictions)


print("Predicted Digit:", predicted_digit)
```

```
1/1 [==============================] - 0s 131ms/step
[[1.69910036e-05 1.13708675e-05 2.61833009e-07 2.13417536e-07
  6.33905483e-06 2.33112087e-06 9.99945283e-01 1.64560754e-09
  1.54182544e-05 1.77115669e-06]]
Predicted Digit: 6
```

Web Application



APPENDIX

PROJECT CODE:

The Handwritten Digit Recognition project code consists of a single Jupyter notebook, which includes all the data processing and CNN techniques implementation. The notebook is used to train the CNN model on preprocessed training images and exporting the model. The backend CNN model compilation and running application are the two separated python files.

1. **Project.ipynb**:

This notebook presents the development of a Convolutional Neural Networks model using image samples of handwritten digits. It preprocesses the samples which includes resizing to a standard resolution, converting to grayscale format and splitting the dataset. The CNN model then utilizes this dataset to learn patterns and features of handwritten digits. This model is then capable to respond to user input images and recognize the digits, demonstrating its effectiveness.

2. **Model folder**:

This folder consists of files related to already trained model using the dataset by the file "Project.ipynb". This folder is then utilized by the App's main program i.e. "main_app.py". The App will use this model in order to predict the user's input image

3. **Main_app.py**:

This python file uses streamlit framework for AI/ML classification of digits. This file is the main file of the App interface which when executed through streamlit provides a simplistic interface for the users to upload their handwritten digit image. The input image will be preprocessed in the same way the trained images are done. The app will then load the CNN model from the model folder and will effectively utilize this pre-trained model to predict the user input image and display the outputs.

For access to the complete code and detailed implementation, please find the project's GitHub repository at the following link:
https://github.com/tejaswini22012003/digit_recognition_ml/tree/master

For access to the web application interface for all users, please find the project's public URL at the following link:

https://digit-recognition-cnn.streamlit.app

BIBLIOGRAPHY

1. Handwritten Digit Recognition using Deep Learning Networks. https://ieeexplore.ieee.org/document/10016012

2. Handwritten Digit Recognition using CNN https://www.researchgate.net/publication/367221312_Handwritten_Digit_Recognition_Using_CNN

3. Handwritten Digit Image Recognition Using Machine Learning https://jieee.a2zjournals.com/index.php/ieee/article/view/54