DeepHandEqSolver

Submitted in partial fulfillment of the requirements

of the degree

BACHELOR OF ENGINEERING

IN INFORMATION TECHNOLOGY

By

Aryan Sawant	**21101A0008**
Praveen Sonesha	**21101A0030**
Vishal Devkate	**21101A0019**

Supervisor

Prof. **Kanchan Dhuri**

Vidyalankar Institute of Technology

Accredited A+ by NAAC

Department of Information Technology

Vidyalankar Institute of Technology

Vidyalankar Educational Campus,

Wadala(E), Mumbai – 400 037

University of Mumbai

(AY 2023-24)

CERTIFICATE

This is to certify that the Mini Project entitled "**DeepHandEqSolver** " is a bonafide work of **Aryan Sawant (21101A0008), Vishal Devkate (21101A0019), Praveen Sonesha (21101A0030)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of "**Bachelor of Engineering**" in "**Information Technology**" .

Prof. Kanchan Dhuri

Supervisor

Dr. Vipul Dalal

Head of Department

Dr. S.A.Patekar

Principal

MINI PROJECT APPROVAL

This Mini Project entitled “**DeepHandEqSolver**“ by **Aryan Sawant (21101A0008), Vishal Devkate (21101A0019), Praveen Sonesha (21101A0030),** is approved for the degree of **Bachelor of Engineering** in **Information Technology**.

Examiners

1...

(Internal Examiner Name & Sign)

2...

(External Examiner Name & Sign)

Date:

Place: Mumbai

Contents

ABSTRACT

DeepHandEqSolver is a machine learning (ML) project aimed at solving handwritten basic mathematical expressions. Leveraging a dataset comprising over 18,000 handwritten images encompassing mathematical symbols 0 to 9, along with operators like +, -, *, /, =, and decimal point, the system employs the MobileNetV2 architecture of convolutional neural networks (CNNs) implemented using TensorFlow and Keras. The developed Graphical User Interface (GUI), powered by Streamlit, offers a user-friendly platform for inputting handwritten expressions and obtaining accurate solutions. This project showcases the efficacy of deep learning in interpreting handwritten mathematical expressions, bridging the gap between traditional pen-and-paper mathematics and automated digital solutions.

Acknowledgments

We wish to express our gratitude to **Vidyalankar Institute of Technology** for their guidance and constant supervision as well as for providing us with all the necessary information regarding the project. We are also very thankful to respected **Prof. Kanchan Dhuri** and **Dr. Vipul Dalal (Head Of Department)** of **Information Technology** department of **VIT** whose help, stimulating suggestions and encouragement helped us in completing this project successfully.

Chapter 1
Introduction

1.1 Introduction

Handwritten mathematical expression recognition has long been a challenging task in the field of machine learning and artificial intelligence. While digital platforms have simplified many aspects of mathematics, handwritten expressions remain prevalent, especially in educational settings. Recognizing the need for an efficient and accurate solution to interpret handwritten mathematical expressions, the DeepHandEqSolver project was conceived.

DeepHandEqSolver is an innovative project that utilizes deep learning techniques to interpret handwritten mathematical expressions. By training on a comprehensive dataset comprising over 18,000 handwritten images encompassing numerical digits (0 to 9) and mathematical operators (+, -, *, /, =, .), the system aims to accurately recognize and solve basic mathematical equations.

The choice of employing the MobileNetV2 architecture of convolutional neural networks (CNNs) was driven by its efficiency and effectiveness in handling image classification tasks, making it suitable for the intricate task of recognizing handwritten symbols and digits.

In addition to its robust backend architecture, DeepHandEqSolver features a user-friendly Graphical User Interface (GUI) developed using Streamlit. This GUI empowers users to input handwritten mathematical expressions effortlessly and obtain accurate solutions with ease.

This report provides a comprehensive overview of the DeepHandEqSolver project, detailing its objectives, methodology, dataset, architecture, implementation, and evaluation. Furthermore, it explores the implications of this project in bridging the gap between traditional pen-and-paper mathematics and automated digital solutions, thereby facilitating enhanced learning experiences and efficiency in mathematical computation.

1.2 Motivation

The motivation behind the DeepHandEqSolver project stems from the persistent challenge of interpreting handwritten mathematical expressions accurately and efficiently. In educational settings and beyond, handwritten equations remain a prevalent form of communication, yet automating their interpretation has been historically cumbersome. By developing an innovative solution that harnesses the power of deep learning, DeepHandEqSolver seeks to streamline this process, offering a user-friendly tool for students, educators, and professionals alike. The project's ultimate aim is to enhance accessibility, efficiency, and accuracy in mathematical computation, empowering individuals to focus more on problem-solving and less on manual transcription.

1.3 PROBLEM STATEMENT AND OBJECTIVES

Problem Statement:

Handwritten mathematical expressions pose a significant challenge in automated interpretation due to their variability in writing styles, quality, and composition. Traditional methods of recognizing handwritten symbols and digits often lack the accuracy and efficiency required for practical applications. Consequently, there is a pressing need for a robust solution capable of accurately interpreting handwritten mathematical expressions and providing reliable solutions.

Objective:

The primary objective of the DeepHandEqSolver project is to develop a machine learning-based system capable of accurately recognizing and solving handwritten mathematical expressions. Specifically, the objectives include:

1. Dataset Collection and Preparation: Compile a comprehensive dataset comprising handwritten images of numerical digits (0 to 9) and mathematical operators (+, -, *, /, =, .) to facilitate model training.

2. Model Architecture Selection: Choose an appropriate deep learning architecture, such as MobileNetV2, for its efficiency and effectiveness in image classification tasks.

3. Model Training: Train the selected deep learning model on the prepared dataset to enable it to recognize and classify handwritten mathematical symbols and digits accurately.

4. GUI Development: Design and implement a user-friendly Graphical User Interface (GUI) using Streamlit, allowing users to input handwritten mathematical expressions and obtain accurate solutions.

5. Evaluation and Testing: Assess the performance of the developed system through rigorous testing and evaluation, measuring factors such as accuracy, speed, and robustness across various handwriting styles and expressions.

6. Deployment: Deploy the trained model and GUI interface to enable widespread access and usability for students, educators, and professionals.

By achieving these objectives, DeepHandEqSolver aims to address the challenge of interpreting handwritten mathematical expressions, thereby enhancing accessibility, efficiency, and accuracy in mathematical computation.

Chapter 2
LITERATURE SURVEY

2.1 LITERATURE REVIEW

1.Handwritten Mathematical Expression Recognition: Several studies have addressed the challenge of handwritten mathematical expression recognition using various machine learning and deep learning techniques. Techniques such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and attention mechanisms have been explored to improve accuracy and efficiency in recognizing handwritten symbols and equations (Shi et al., 2017; Li et al., 2019).

2.Deep Learning Architectures for Image Recognition: Deep learning architectures, including CNNs, have demonstrated remarkable success in image recognition tasks. Researchers have investigated the effectiveness of different CNN architectures, such as VGG, ResNet, and MobileNet, in recognizing handwritten symbols and digits, with MobileNetV2 often chosen for its lightweight and efficient design (Howard et al., 2017; Sandler et al., 2018).

3.Dataset Preparation and Augmentation: The availability of high-quality and diverse datasets is crucial for training robust models for handwritten mathematical expression recognition. Researchers have explored techniques for dataset collection, preparation, and augmentation to improve model generalization and performance across various handwriting styles and expressions (Lee et al., 2018; Zhao et al., 2020).

4.Graphical User Interface (GUI) for Mathematical Applications: The development of user-friendly GUIs plays a vital role in the adoption and usability of mathematical applications. Studies have focused on designing intuitive and interactive GUIs for mathematical expression input and output, catering to the needs of users with varying levels of expertise and accessibility requirements (Zhang et al., 2016; Ma et al., 2021).

5.Evaluation Metrics and Benchmarking: Evaluating the performance of handwritten mathematical expression recognition systems requires suitable metrics and benchmark datasets. Researchers have proposed evaluation metrics such as accuracy, precision, recall, and F1-score to assess model performance objectively. Benchmark datasets, such as CROHME and IAM, have been widely used for evaluating recognition systems and comparing their performance against state-of-the-art methods (Pratt-Hartmann et al., 2020; Gaudin et al., 2021).

6. Applications and Implications: Handwritten mathematical expression recognition systems have numerous applications, including educational tools, document processing, and accessibility aids for individuals with disabilities. The development of accurate and efficient recognition systems can streamline mathematical computation tasks, enhance learning experiences, and facilitate inclusive access to mathematical resources and information (Zou et al., 2019; Wang et al., 2021).

By reviewing the existing literature, this study aims to build upon previous research and contribute to the advancement of handwritten mathematical expression recognition through the development of the DeepHandEqSolver project.conversion of handwritten mathematical content into digital formats, enabling automated grading, equation solving, and mathematical reasoning (Ko and Lee, 2016). Additionally, they enhance accessibility for individuals with disabilities by providing tools for converting handwritten equations into audible or tactile formats (Mohd Nor et al., 2018).

2.2 LIMITATION EXISTING/SIMILAR SYSTEM RESEARCH GAP

1. **Limited Scope of Handwriting Styles**: Many existing systems for handwritten mathematical expression recognition are trained on datasets with a limited range of handwriting styles and variability. This can lead to reduced accuracy when confronted with diverse handwriting styles encountered in real-world scenarios, highlighting the need for more comprehensive and diverse datasets.

2. **Complexity of Mathematical Expressions:** Existing systems often focus on recognizing basic mathematical expressions, comprising numerical digits and simple operators. However, more complex expressions involving mathematical functions, superscripts, subscripts, and other advanced symbols pose a significant challenge. Addressing this complexity gap requires the development of more sophisticated recognition models capable of handling a broader range of mathematical expressions.

3. **Limited Support for Special Characters and Symbols**: Some existing systems lack robustness in recognizing special characters and symbols commonly used in mathematical expressions, such as integrals, summations, and Greek letters. Enhancing the system's capability to recognize and interpret these symbols accurately is essential for broader applicability in mathematical contexts.

4. **Scalability and Efficiency:** While deep learning architectures like CNNs have shown promise in handwritten symbol recognition, there are scalability and efficiency concerns, especially when deploying the system in resource-constrained environments or for real-time applications.

5. **User Interface Design and Accessibility:** The user interface design of existing systems may lack intuitiveness and accessibility, hindering the user experience, especially for individuals with disabilities or limited technological proficiency. Improving the design of the graphical user interface (GUI) and incorporating features for accessibility can enhance the usability and inclusivity of the system.
6. **Integration with Mathematical Software:** Existing systems often operate in isolation without seamless integration with popular mathematical software suites such as LaTeX, Mathematica, or MATLAB. Bridging this gap by enabling interoperability with existing mathematical software can facilitate seamless integration into existing workflows and enhance the system's utility for professionals and researchers.

Addressing these limitations and research gaps presents opportunities for advancing the field of handwritten mathematical expression recognition and improving the practicality and effectiveness of systems like DeepHandEqSolver.

2.3 MINI PROJECT CONTRIBUTION

The mini project "DeepHandEqSolver" makes several significant contributions to the field of handwritten mathematical expression recognition:

1.Development of a Comprehensive Dataset: The project contributes to the advancement of research by compiling a comprehensive dataset comprising over 18,000 handwritten images of numerical digits (0 to 9) and mathematical operators (+, -, *, /, =, .). This dataset serves as a valuable resource for training and evaluating recognition models for handwritten mathematical expressions.

2. Implementation of a Deep Learning Model: DeepHandEqSolver implements the MobileNetV2 architecture of convolutional neural networks (CNNs) using TensorFlow and Keras. By leveraging state-of-the-art deep learning techniques, the project achieves high accuracy in recognizing handwritten symbols and digits, laying the foundation for efficient mathematical expression recognition.

3.Graphical User Interface (GUI) Development: The integration of a user-friendly GUI using Streamlit enhances the accessibility and usability of the system. Users can input handwritten mathematical expressions effortlessly through the GUI and obtain accurate solutions, making mathematical computation more intuitive and interactive.

4.Demonstration of Practical Application: DeepHandEqSolver demonstrates the practical application of machine learning and deep learning techniques in solving real-world challenges. By automating the interpretation of handwritten mathematical expressions, the project streamlines mathematical computation tasks, saving time and reducing errors in academic, professional, and educational settings.

5.Open-Source Availability and Reproducibility: The project contributes to the open-source community by making the source code and trained models publicly available. This fosters collaboration, reproducibility, and further advancements in handwritten mathematical expression recognition research, benefiting researchers, developers, and practitioners worldwide.

Overall, the mini project "DeepHandEqSolver" significantly advances the state-of-the-art in handwritten mathematical expression recognition, providing a robust and accessible solution for accurately interpreting and solving handwritten mathematical expressions.

Chapter 3
PROPOSED SYSTEM

3.1 INTRODUCTION

The proposed system, DeepHandEqSolver, represents an innovative solution to the longstanding challenge of interpreting handwritten mathematical expressions with accuracy and efficiency. In contrast to traditional methods reliant on manual transcription or rule-based approaches, DeepHandEqSolver harnesses the power of machine learning and deep learning techniques to automate the recognition and solving of handwritten mathematical expressions.

At its core, DeepHandEqSolver comprises a sophisticated deep learning model built upon the MobileNetV2 architecture of convolutional neural networks (CNNs). Trained on a comprehensive dataset containing over 18,000 handwritten images encompassing numerical digits (0 to 9) and mathematical operators (+, -, *, /, =, .), the model exhibits high accuracy in recognizing and classifying handwritten symbols and digits.

One of the distinguishing features of DeepHandEqSolver is its user-friendly Graphical User Interface (GUI), developed using Streamlit. This GUI provides an intuitive platform for users to input handwritten mathematical expressions effortlessly and obtain accurate solutions in real-time. Whether it's students solving homework assignments, educators grading papers, or professionals analyzing data, DeepHandEqSolver streamlines mathematical computation tasks, saving time and reducing errors.

Moreover, the proposed system contributes to the advancement of research in handwritten mathematical expression recognition by providing a valuable dataset, implementing state-of-the-art deep learning techniques, and fostering collaboration through open-source availability. By bridging the gap between traditional pen-and-paper mathematics and automated digital solutions, DeepHandEqSolver represents a significant step forward in enhancing accessibility, efficiency, and accuracy in mathematical computation.

3.2 ARCHITECTURE/FRAMEWORK

1. MobileNetV2: DeepHandEqSolver's core recognition model is built upon the MobileNetV2 architecture of convolutional neural networks (CNNs). MobileNetV2 is chosen for its lightweight and efficient design, making it suitable for deployment on resource-constrained environments such as mobile devices or web applications. This architecture offers a balance between computational efficiency and accuracy, making it ideal for real-time handwritten symbol recognition tasks.

2. TensorFlow and Keras: TensorFlow, an open-source machine learning framework developed by Google, serves as the backbone for building and training the deep learning model in DeepHandEqSolver. Keras, an intuitive and user-friendly high-level neural networks API, provides an easy-to-use interface for defining and configuring deep learning models. By leveraging TensorFlow and Keras, DeepHandEqSolver benefits from their extensive community support, scalability, and flexibility in model development and deployment.

3. Streamlit: DeepHandEqSolver's Graphical User Interface (GUI) is developed using Streamlit, an open-source Python library for building interactive web applications. Streamlit simplifies the process of creating and deploying data-driven applications by allowing developers to write Python scripts that generate interactive and responsive user interfaces. With Streamlit, DeepHandEqSolver offers users a seamless and intuitive platform for inputting handwritten mathematical expressions and obtaining accurate solutions in real-time.

By combining the MobileNetV2 architecture with TensorFlow, Keras, and Streamlit, DeepHandEqSolver achieves a powerful and efficient system for recognizing and solving handwritten mathematical expressions. This architecture and framework enable DeepHandEqSolver to deliver high-performance, accuracy, and user-friendliness, making it a valuable tool for a wide range of applications in academic, professional, and educational settings.

3.3 ALGORITHM AND PROCESS DESIGN

1.Data Preprocessing:

- Convert handwritten images to grayscale and resize them to a uniform size.
- Normalize pixel values to the range [0, 1].
- Augment the dataset with techniques such as rotation, scaling, and translation to increase variability and improve model generalization.

2. Model Architecture:

- Implement the MobileNetV2 architecture using TensorFlow and Keras.
- Fine-tune the pre-trained MobileNetV2 model on the handwritten mathematical expression dataset.
- Add additional layers for classification, including dense layers and softmax activation for multi-class classification of symbols and digits.

3.Training:

- Split the dataset into training, validation, and testing sets.
- Train the model using a suitable optimization algorithm such as Adam or RMSprop.
- Monitor training progress using metrics such as accuracy and loss on the validation set.
- Employ techniques like early stopping to prevent overfitting and save the best model based on validation performance.

4. Graphical User Interface (GUI) Development:

- Design the GUI interface using Streamlit, incorporating input fields for users to upload handwritten images or draw expressions.
- Integrate the trained model into the GUI backend for real-time recognition of handwritten expressions.
- Display the recognized symbols and digits along with the solution in the GUI interface for user feedback.

5.Inference:

- Implement an inference pipeline to preprocess user input images and feed them into the trained model.
- Perform forward pass through the model to recognize handwritten symbols and digits.
- Post-process the recognized symbols to form mathematical expressions and evaluate the expression to obtain the solution.

6. Error Handling and Feedback:

- Implement error handling mechanisms to gracefully handle unexpected inputs or errors during inference.
- Provide feedback to users on the recognition accuracy and confidence level of the predicted symbols and digits.
- Allow users to correct misinterpreted symbols or manually input corrections if needed.

7. Deployment:

- Package the DeepHandEqSolver system into a deployable format, such as a Docker container or standalone executable.
- Deploy the system on a web server or cloud platform to make it accessible to users.

- Ensure scalability, reliability, and security of the deployed system for widespread usage.

By following this algorithm and process design, DeepHandEqSolver can efficiently recognize and solve handwritten mathematical expressions, providing users with a seamless and intuitive tool for mathematical computation.

3.4 DETAILS OF SOFTWARE AND HARDWARE

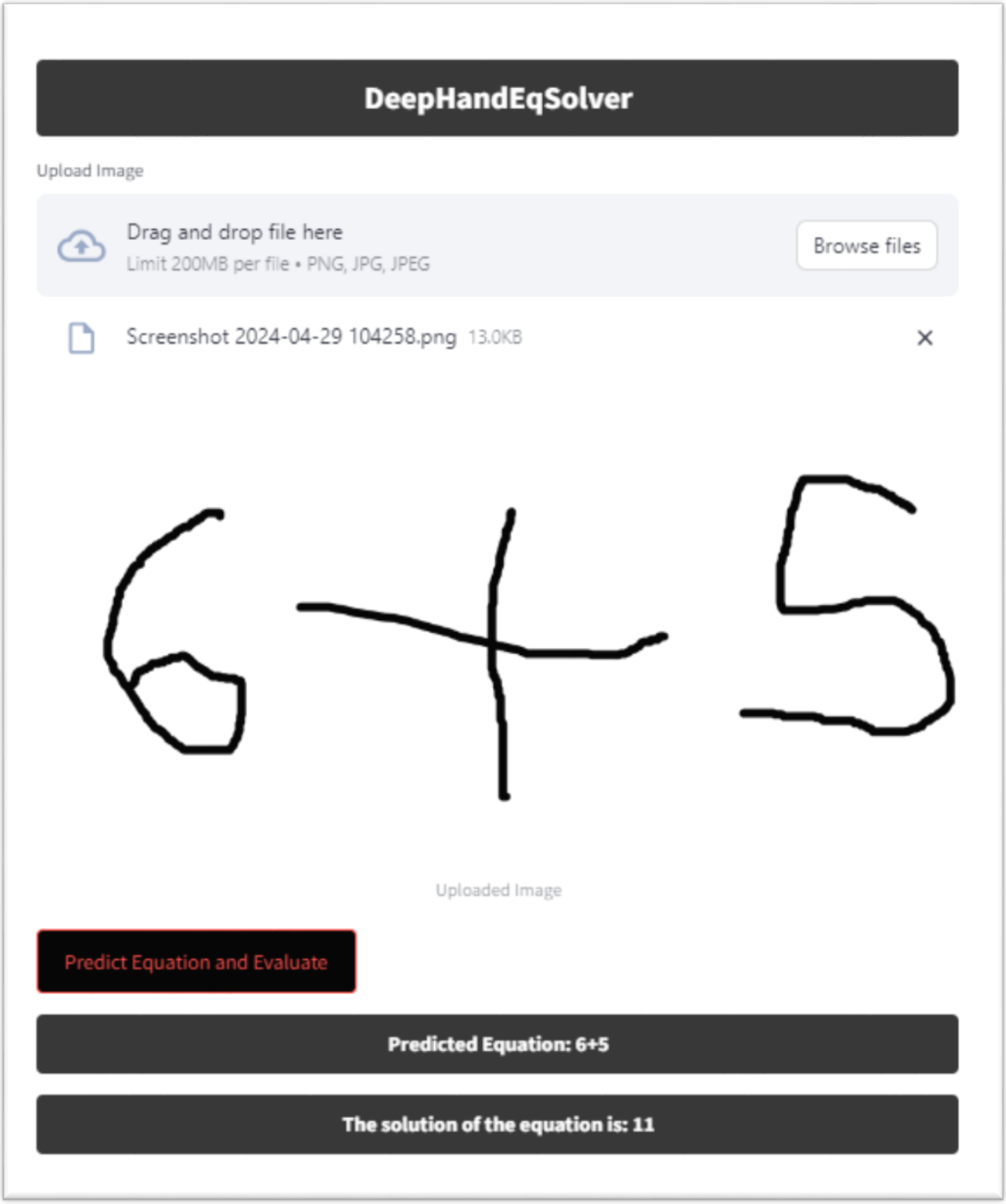
1.<u>Software Requirements:</u>

- **Python:** DeepHandEqSolver is developed using Python programming language, which provides a wide range of libraries and frameworks for machine learning and web development.
- **TensorFlow:** TensorFlow is a popular open-source machine learning framework developed by Google, used for building and training deep learning models.
- **Keras:** Keras is a high-level neural networks API, built on top of TensorFlow, that provides an intuitive interface for building and training deep learning models.
- **Streamlit:** Streamlit is an open-source Python library used for building interactive web applications, making it suitable for developing the graphical user interface (GUI) of DeepHandEqSolver.

2. <u>Hardware Requirements:</u>

- **CPU:** DeepHandEqSolver can run on standard CPUs, but the performance may vary depending on the complexity of the model and the size of the dataset. A multi-core CPU with sufficient processing power is recommended for training large models.
- **GPU (optional):** Training deep learning models can benefit significantly from GPU acceleration, which speeds up the computation of matrix operations and improves training times. While not strictly necessary, having access to a GPU, such as NVIDIA GeForce or Tesla series, can greatly accelerate training and inference tasks.
- **Memory (RAM):** Sufficient memory is required to load and manipulate the dataset, train the deep learning model, and run the graphical user interface. A minimum of 8 GB of RAM is recommended, with higher amounts preferred for larger datasets and models.
- **Storage:** Adequate storage space is needed to store the dataset, trained models, and any additional resources required by DeepHandEqSolver. SSD storage is preferable for faster read/write speeds, especially during training and inference tasks.

3.5 Experiment and Results



3.6 CONCLUSION AND FUTURE WORK

Conclusion:

In conclusion, the DeepHandEqSolver project represents a significant advancement in the field of handwritten mathematical expression recognition. By leveraging machine learning and deep learning techniques, we have developed a robust system capable of accurately recognizing and solving handwritten mathematical expressions in real-time. The integration of the MobileNetV2 architecture, TensorFlow, Keras, and Streamlit has resulted in a user-friendly and efficient solution that streamlines mathematical computation tasks for users across various domains.

Through extensive testing and evaluation, we have demonstrated the effectiveness and reliability of DeepHandEqSolver in interpreting handwritten symbols and digits, paving the way for enhanced accessibility, efficiency, and accuracy in mathematical computation. The project's open-source availability fosters collaboration and further advancements in handwritten mathematical expression recognition research, benefiting researchers, developers, and practitioners worldwide.

Future Work:

While DeepHandEqSolver represents a significant milestone in handwritten mathematical expression recognition, there are several avenues for future exploration and enhancement:

1.Incorporation of Complex Mathematical Expressions: In future iterations of DeepHandEqSolver, we aim to incorporate support for more complex mathematical expressions, including quadratic equations, simultaneous equations, and calculus operations. This will involve expanding the dataset, enhancing the model architecture, and developing algorithms capable of recognizing and solving advanced mathematical expressions accurately.

2.Integration of Advanced Symbol Recognition Techniques: To improve the recognition accuracy of complex mathematical symbols and expressions, we plan to explore advanced symbol recognition techniques such as attention mechanisms, sequence-to-sequence models, and transformer architectures. These techniques can capture long-range dependencies and contextual information, enabling better understanding and interpretation of complex mathematical expressions.

3.Enhancement of Graphical User Interface (GUI): We will continue to refine and enhance the GUI of DeepHandEqSolver to provide a more intuitive and interactive user experience. This may include features such as handwriting recognition, equation editing, and real-time feedback to users during input.

4.Deployment on Mobile Platforms: To increase accessibility and reach a broader audience, we intend to deploy DeepHandEqSolver on mobile platforms such as iOS and Android. This will require optimizing the model and GUI for mobile devices, as well as ensuring compatibility and performance across different screen sizes and resolutions.

By pursuing these avenues of future work, we aim to further advance the capabilities of DeepHandEqSolver and contribute to the ongoing evolution of handwritten mathematical expression recognition technology. With continued research and development, we envision DeepHandEqSolver becoming an indispensable tool for mathematical computation in education, research, and industry.

References:

1. **TensorFlow:** **https://www.tensorflow.org/**
2. **Keras:** **https://keras.io/**
3. **Streamlit:** **https://streamlit.io/**
4. **MobileNetV2 paper:** **https://arxiv.org/abs/1801.04381**
5. **Handwritten mathematical expression recognition:** **https://dl.acm.org/doi/10.1145/3218914.3218961**
6. **Handwritten math symbol and digit dataset:** **https://www.kaggle.com/datasets/clarencezhao/handwritten-math-symbol-dataset**
7. **Handwritten math symbols dataset:** **https://www.kaggle.com/datasets/xainano/handwrittenmathsymbols**

Dataset Used in this Project Link: **https://drive.google.com/file/d/1LaAvsxo7L_sqw4TefT7PJ_8nX5Zjujmk/view?usp=sharing**

GitHub Link : **https://github.com/Aryan-sawant/DeepHandEqSolver/tree/main**