

HANDWRITTEN EQUATION SOLVER USING DEEP LEARNING

|  |  |
| --- | --- |
| Mentor Name:  Dr. Sayantan Sinha | Submitted By:  Saksham Saxena  (2215500140: AE)  Vanshika Raghav (2215500053: AE) |

**1. Title of the Project**

Handwritten Equation Solver Using Deep Learning

**2. Statement About the Problem**

The problem of recognizing and solving handwritten mathematical equations is complex due to varying handwriting styles, symbol ambiguities, and the complexity of mathematical notation. Traditional OCR systems struggle with accurately interpreting handwritten equations, which include not just digits, but also operators and symbols. This project aims to develop a deep learning-based system that can accurately recognize and solve handwritten mathematical equations in real time.

**3. Describe the Problem Statement**

The core problem lies in interpreting handwritten mathematical equations, which requires accurate symbol recognition, parsing the recognized symbols into a structured mathematical format, and solving the equation. Handwritten inputs are often inconsistent, with individual variations in writing style, symbol formation, and alignment. Additionally, complex mathematical expressions (involving fractions, powers, etc.) further complicate recognition and computation.

**4. Objective and Scope of the Project**

Objective:

* Develop a deep learning-based system capable of recognizing handwritten digits, operators, and mathematical symbols from images of handwritten equations.
* Parse these symbols into structured mathematical expressions.
* Solve the parsed mathematical expressions in real time, providing accurate solutions.

Scope:

* Recognize and solve basic arithmetic operations and algebraic expressions.
* Support common mathematical symbols (+, -, /, \*, =, etc.).
* Work with multiple handwriting styles using a custom or existing dataset.
* Provide real-time solutions through a user interface for inputting handwritten equations.
* Evaluate the system using symbol recognition accuracy, equation parsing, and solution correctness.

**5. Methodology**

The methodology for this project will follow these steps:

Preprocessing:

* Normalize and preprocess input images, including binarization, resizing, and noise removal, to improve symbol recognition accuracy.

Symbol Recognition:

* Implement a Convolutional Neural Network (CNN), such as ResNet or LeNet, to classify handwritten digits, operators, and other mathematical symbols.
* Train the model using a dataset that includes diverse handwriting styles and mathematical symbols.

Sequence Prediction:

* Use Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, or Transformer models to predict the sequence of symbols and operators in the equation.
* These models will handle multi-line and multi-symbol equations, ensuring proper order of the recognized elements.

Equation Parsing:

* Parse the recognized sequence into structured formats like LaTeX or MathML, which can be easily interpreted by mathematical solvers.

Equation Solving:

* Use symbolic computation tools (e.g., SymPy) to interpret the parsed equations and compute the solutions.
* Develop an algorithm that processes the parsed expressions and performs the required mathematical operations.

Evaluation:

* Evaluate the system based on symbol recognition accuracy, correct parsing of equations, and solution accuracy. Compare performance against baseline models and state-of-the-art methods.

**6. Hardware & Software to be Used**

Hardware:

* NVIDIA GPU for deep learning model training.
* A standard laptop for development and running the final system.

Software:

* Programming Language: Python.
* Libraries:
* TensorFlow or PyTorch for building and training deep learning models.
* OpenCV for image preprocessing and manipulation.
* SymPy for symbolic mathematics and equation solving.
* NumPy and Pandas for data handling and manipulation.

**7. Future Work of this Project**

* Extend the system to recognize and solve more complex mathematical notations, such as integrals, matrices, and differential equations.
* Integrate the solution into a mobile app or web application, allowing users to capture handwritten equations via camera and receive real-time solutions.
* Implement real-time feedback, improving the system by learning from user inputs and corrections.
* Expand the system to support multiple languages and accommodate different writing styles or symbols used in various regions.

**8. References/Bibliography**

* SnapSolve – A Novel Mathematics Equation Solver using Deep Learning

By:- Priyank Shah ,Ashwini Save,Nitiket Shinde,Deep Limbad

* SolveIt : An Application for Automated Recognition and Processing of Handwritten Mathematical Equations

By;-Sagar Bharadwaj KS\*

Mentor’s Sign. (Dr. Sayantan Sinha)