

**DATE: 14.06.2025 XX/XX/XXXX**

**Mahmut Cenk EFELER**

**Ender Efe ERYEGÜL – 280206062**

**Bartu ERDAL – 280206061**

**Recognition and Mathematical Operation System for Handwritten Expressions**

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**EE491 / EE492 PROJECT REPORT**

**Abstract:**

This project presents a recognition and processing system for handwritten mathematical expressions. The main pipeline consists of two subsystems: the Image Processing System and the Mathematical Processing System.

The Image Processing System takes a handwritten expression image as input and applies preprocessing steps such as blurring, thresholding, and contour detection using OpenCV. Individual characters are segmented and normalized, then passed through a convolutional neural network trained on a custom dataset to classify them into letters, digits, or mathematical symbols. The recognized symbols, along with their positional metadata, are structured into a list for further interpretation.

The Mathematical Processing System uses this structured list to reconstruct a mathematically meaningful expression. It leverages tokenization, postfix conversion, and a binary expression tree to enable symbolic computations such as differentiation, integration, and polynomial equation solving. All modules are implemented using open-source tools including Python, OpenCV, and PyTorch.

The system achieves accurate character recognition and reliable symbolic analysis on a wide range of inputs. It offers an extensible architecture that can support future features like handwriting adaptation and real-time deployment. A mobile interface has also been developed using React Native and Flask, enabling users to draw mathematical expressions on a tablet and receive real-time results through the system.

**ABSTRACT i**

**TABLE OF CONTENTS ii**

**ABBREVIATOBS iii**

**LIST OF FIGURES iv**

**LIST OF TABLES v**

1. **INTRODUCTION**
2. **PROBLEM DEFINITION**
   1. Subproblems of the System
3. **PROPOSED SOLUTION**
   1. Image Processing
      1. Character Segmentation
      2. Character Classification (CNN)
      3. Symbol Parser and Expression Construction
   2. Mathematical Processing
      1. Tokenizer
      2. Postfix Conversion
      3. Expression Tree and Math Operations
4. **RESULTS AND DISCUSSION** 
   1. Accuracy and Loss Analysis
   2. Confusion Matrix
   3. Real Input Examples
5. **CONCLUSION**

**REFERENCES**

**APPENDIX**

**LIST OF FIGURES**

**LIST OF TABLES**