Mathmagic

**mathamatical Extractor**

This is a program which will extract farmulas from images & calculate them

Contents

[Summary 2](#_Toc181183860)

[Introduction 2](#_Toc181183861)

[Program Explanation 2](#_Toc181183862)

[1. \*\*Imports\*\*: 2](#_Toc181183863)

[2. \*\*Function fix\_multiplication(expr)\*\*: 2](#_Toc181183864)

[3. \*\*Function evaluate\_trig(expr)\*\*: 2](#_Toc181183865)

[4. \*\*Function extract\_formula(image\_path)\*\*: 2](#_Toc181183866)

[5. \*\*Function show\_trigonometric\_steps(left\_side, right\_side)\*\*: 2](#_Toc181183867)

[6. \*\*Function show\_logarithmic\_steps(left\_side, right\_side)\*\*: 3](#_Toc181183868)

[7. \*\*Function create\_equation(formula)\*\*: 3](#_Toc181183869)

[8. \*\*Function evaluate\_or\_solve(left\_side, right\_side)\*\*: 3](#_Toc181183870)

[9. \*\*Function main()\*\*: 3](#_Toc181183871)

[10. \*\*Execution Block\*\*: 3](#_Toc181183872)

[Algorithm Outline 3](#_Toc181183873)

[1. \*\*Input\*\*: 3](#_Toc181183874)

[2. \*\*Image Processing\*\*: 4](#_Toc181183875)

[3. \*\*Formula Extraction\*\*: 4](#_Toc181183876)

[4. \*\*For Each Formula\*\*: 4](#_Toc181183877)

# Summary

This code serves as a mathematical formula extractor and evaluator. It combines image processing, text extraction, and symbolic mathematics to analyze equations, providing step-by-step solutions for trigonometric and logarithmic expressions. The algorithm flows from user input through image processing to formula extraction and evaluation, handling errors and providing informative outputs along the way.

# Introduction

This document presents the functioning of math magic. This program extracts formulas from an image and computes them.

# Program Explanation

## 1. \*\*Imports\*\*:

- **cv2**: OpenCV library for image processing.

- **pytesseract**: Tesseract-OCR Engine, used for optical character recognition (OCR).

- **sympy**: Library for symbolic mathematics, used to manipulate mathematical expressions.

## 2. \*\*Function fix\_multiplication(expr)\*\*:

- This function Uses a regular expression to add a multiplication operator (\*) between a number and a variable or parenthesis,

e.g., converting 2(x) to 2\*x.

## 3. \*\*Function evaluate\_trig(expr)\*\*:

- This function takes a trigonometric expression as a string, converts it to a symbolic expression using sympy, and evaluates it.

- It handles exceptions and prints errors if evaluation fails.

## 4. \*\*Function extract\_formula(image\_path)\*\*:

- Loads an image using OpenCV.

- If the image is loaded successfully, it applies Tesseract OCR to extract text from the image.

- Splits the extracted text by new lines and filters out empty lines to obtain a list of formulas.

## 5. \*\*Function show\_trigonometric\_steps(left\_side, right\_side)\*\*:

- Takes the left and right sides of an equation as strings.

- Converts them to sympy expressions.

- Checks for variables and attempts to solve the equation using sympy. It prints steps of the solution process.

- Evaluates the expressions and provides feedback based on the existence of solutions.

## 6. \*\*Function show\_logarithmic\_steps(left\_side, right\_side)\*\*:

- Similar to show\_trigonometric\_steps, but designed for logarithmic equations.

- It follows the same logic for extracting variables, solving, and evaluating.

## 7. \*\*Function create\_equation(formula)\*\*:

- This function checks for a single = sign in the formula and splits the formula into left and right sides.

- It trims any whitespace and assigns 0 to the right side if it’s empty.

## 8. \*\*Function evaluate\_or\_solve(left\_side, right\_side)\*\*:

- Fixes multiplication issues in the provided expressions.

- Converts them into sympy expressions.

- Determines whether the equation is trigonometric or logarithmic and calls the appropriate function to show steps and solutions.

- If there are no variables, it evaluates the expression numerically.

## 9. \*\*Function main()\*\*:

- Prompts the user for an image file path.

- Extracts formulas from the image using extract\_formula.

- For each extracted formula, it creates an equation and evaluates it, printing the solutions.

## 10. \*\*Execution Block\*\*:

- Ensures that main() runs when the script is executed directly.

# Algorithm Outline

## 1. \*\*Input\*\*:

- Prompt user for an image file path.

## 2. \*\*Image Processing\*\*:

- Load the image using OpenCV.

- Extract text from the image using Tesseract OCR.

## 3. \*\*Formula Extraction\*\*:

- Split the OCR output into separate formulas.

## 4. \*\*For Each Formula\*\*:

- Create an equation from the formula (split into left and right sides).

- Fix multiplication syntax in both sides.

- Evaluate or solve the equation:

- If trigonometric, use show\_trigonometric\_steps.

- If logarithmic, use show\_logarithmic\_steps.

- If neither, check for variables and solve or evaluate numerically.

5. \*\*Output\*\*:

-Print the solutions or evaluations for each formula.