MATHAMATICAL EXTRACTOR

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

Mathmagic

Contents

Summary 2

Introduction 2

Program Explanation 2

1. **Imports**: 2

2. **Function `fix_multiplication(expr)`**: 2

3. **Function `evaluate_trig(expr)`**: 2

4. **Function `extract_formula(image_path)`**: 2

5. **Function `show_trigonometric_steps(left_side, right_side)`**: 2

6. **Function `show_logarithmic_steps(left_side, right_side)`**: 3

7. **Function `create_equation(formula)`**: 3

8. **Function `evaluate_or_solve(left_side, right_side)`**: 3

9. **Function `main()`**: 3

10. **Execution Block**: 3

Algorithm Outline 3

1. **Input**: 3

2. **Image Processing**: 4

3. **Formula Extraction**: 4

4. **For Each Formula**: 4

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 library for optical character recognition.
- `sympy`: Library for symbolic mathematics.

2. **Function `fix_multiplication(expr)`**:

- 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)`**:

- Converts a trigonometric expression to radians using `sympy` and evaluates it. It handles exceptions to manage errors during evaluation.

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)`**:

- Parses a formula string to separate the left side and right side of the equation.

- Validates that there is at most one `=` sign.

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.