

Build your own Photomath

A user should be able to pass a photo of a math expression and get the result.

Requirements

- a solution should be able to read a pretty expression (i.e. `2 + 2`) handwritten on a blank paper (pretty == very very very pretty :))
- support all possible digits, and operators for addition, subtraction, multiplication, division, and brackets:
`0123456789+-×/()`
- don't overdo it
- instructions for using your service should be described in `README.md`
- make your solution open-sourced, available through one of the publicly available Git repo hosting services

Don't stress too much about fulfilling all the requirements. If you haven't yet worked on a similar task, we hope that it will be a fun learning experience too :)

1. Implement a handwritten character detector

The implementation should be able to receive an image in a form of a numpy array and for each detected character it should return its coordinates, and a cropped image (cropped up to its bounding box). Use `OpenCV`. Don't overdo it.

2. Implement a handwritten character classifier

The implementation should be able to receive a cropped image of the character and return the classification result (the correct label).

Use convolutional neural networks. Use `tensorflow` or `keras` with `tensorflow` backend.

Do not overdo it. The goal is not to train the best model, just to have something that works reasonably for a very very pretty input.

Report the metrics you find useful and comment on them.

3. Implement a solver

Implementation should be able to receive the expression in a form of a string, parse it, and calculate the result. Parsing and solving should be done without calls to `eval` or external libraries.

Combine it all together

Implement a method which will, for a given image of a math expression return its result.

Measure and discuss the real-world problems and compare them to metrics and comments in 2.

Propose improvements.

Bonus points for

- Use `flask` or similar simple web microframework, create the simplest possible UI with an input button that opens the camera on your mobile phone and sends the image to the backend. Show the response from the backend on the screen.
- dockerize your solution