Number Recognizer Documentation

Release 0.1.5

Rafael Lopes Conde dos Reis

February 27, 2017

Contents

Contents:

CHAPTER 1

Number Recognizer

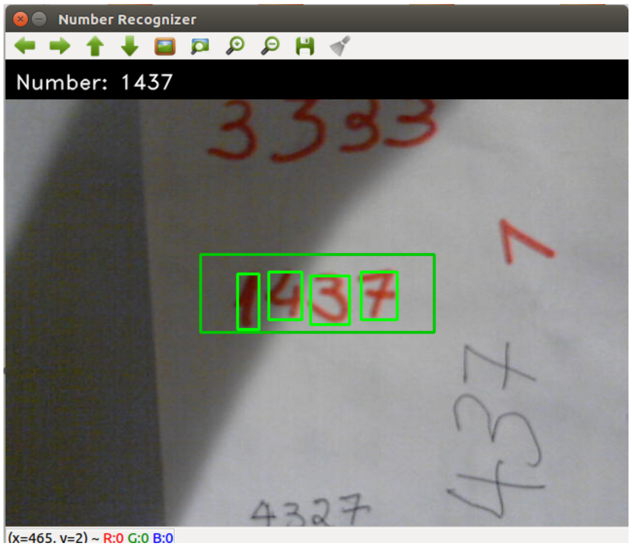
A software for the recognition of handwritten numbers.

- Free software: BSD license
- Repository: https://github.com/condereis/number-recognizer

Requirements

- OpenCV
- TensorFlow
- Numpy
- Pandas
- Click

Recognition Process



```
Number: 1437
```

```
(x=465, y=2) ~ R:0 G:0 B:0
```

- Get a grayscale version of the frame;

- Select a ROI;
- Apply a Gaussian Adaptive Threshold;
- Apply Median Blur to reduce noise;
- Apply Dilation also to reduce noise;
- Apply an AND operation on 3 consecutive frames to reduce noise;
- Find ROI Contours;
- Generate a square image that is white inside the contour and black outside, for each contour found.
- Resize image using linear interpolation to a 28x28 image, for each contour found;
- Run an Deep Convolutional Neural Network trained for MNIST dataset on each image. For more information on the model check here.
- Concatenate de digits and print the number.

Credits

Tools used in rendering this package:

- Cookiecutter
- cookiecutter-pypackage

CHAPTER 2

Installation

Stable release

To install Number Recognizer, first install OpenCV and TensorFlow then run this command in your terminal:

```
$ pip install number-recognizer
```

This is the preferred method to install Number Recognizer, as it will always install the most recent stable release.

If you don’t have pip installed, this Python installation guide can guide you through the process.

From sources

The sources for Number Recognizer can be downloaded from the Github repo.

First install OpenCV and TensorFlow.

You can either clone the public repository:

```
$ git clone git://github.com/condereis/number-recognizer
```

Or download the tarball:

```
$ curl  -OL https://github.com/condereis/number-recognizer/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 3

Usage

Run:

```
$ recognizer
```

Write a number on a clean peace of paper. Position it inside the green box in a way that the number is focused. Press **r** to recognize the number. Press **q** to quit the program.

For more information on the model used to classify the digits check here.

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at https://github.com/condereis/number-recognizer/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

Write Documentation

Number Recognizer could always use more documentation, whether as part of the official Number Recognizer docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/condereis/number-recognizer/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

Get Started!

Ready to contribute? Here's how to set up *number-recognizer* for local development.

1. Fork the *number-recognizer* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/number-recognizer.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv number-recognizer
$ cd number-recognizer/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 recognizer tests
$ python setup.py test or py.test
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.

2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/condereis/number-recognizer/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ python -m unittest tests.test_recognizer
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search