UNIVERSIDADE DA REGIÃO DA CAMPANHA
CENTRO DE CIÊNCIAS EXATAS E AMBIENTAIS
CURSO DE SISTEMAS DE INFORMAÇÃO
FLISOL/BAGÉ

Inteligência Artificial aplicada: reconhecendo caracteres escritos à mão

Por: Alex Dias Camargo

alex@apus.digital

Abril/2021

I. The author

Details..

- **APUS Digital / Combate à Fraude**
- **AI Enginner**
- **@alexcamargoweb**

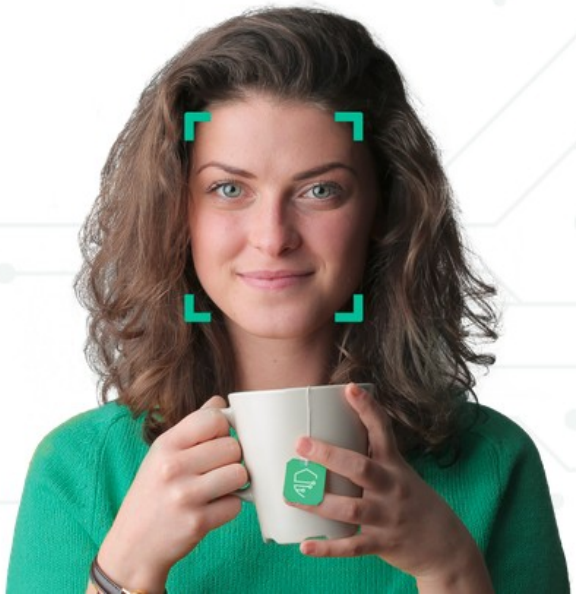
COMBATE A FRAUDE

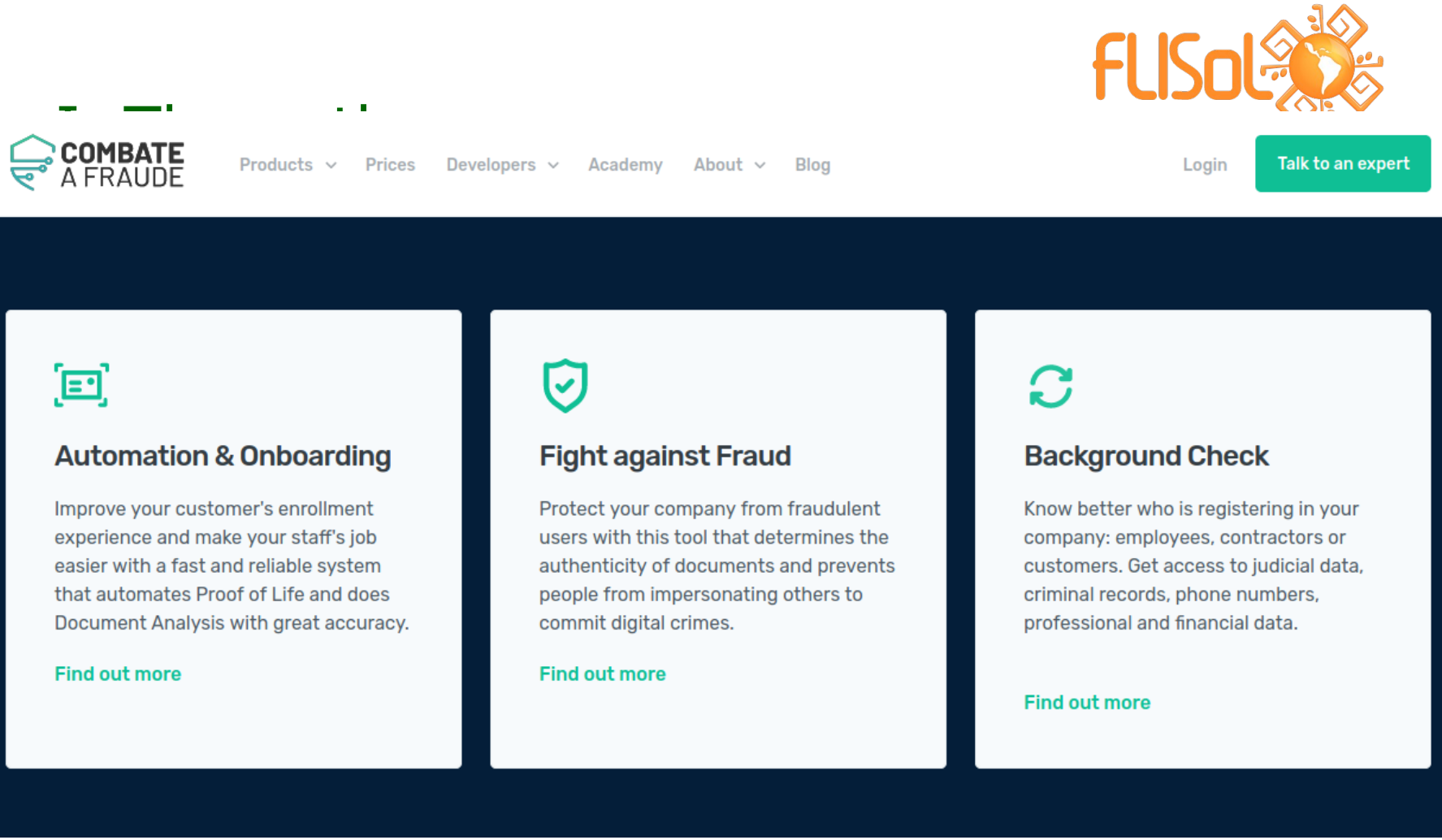
Products Prices Developers Academy About Blog Login Talk to an expert

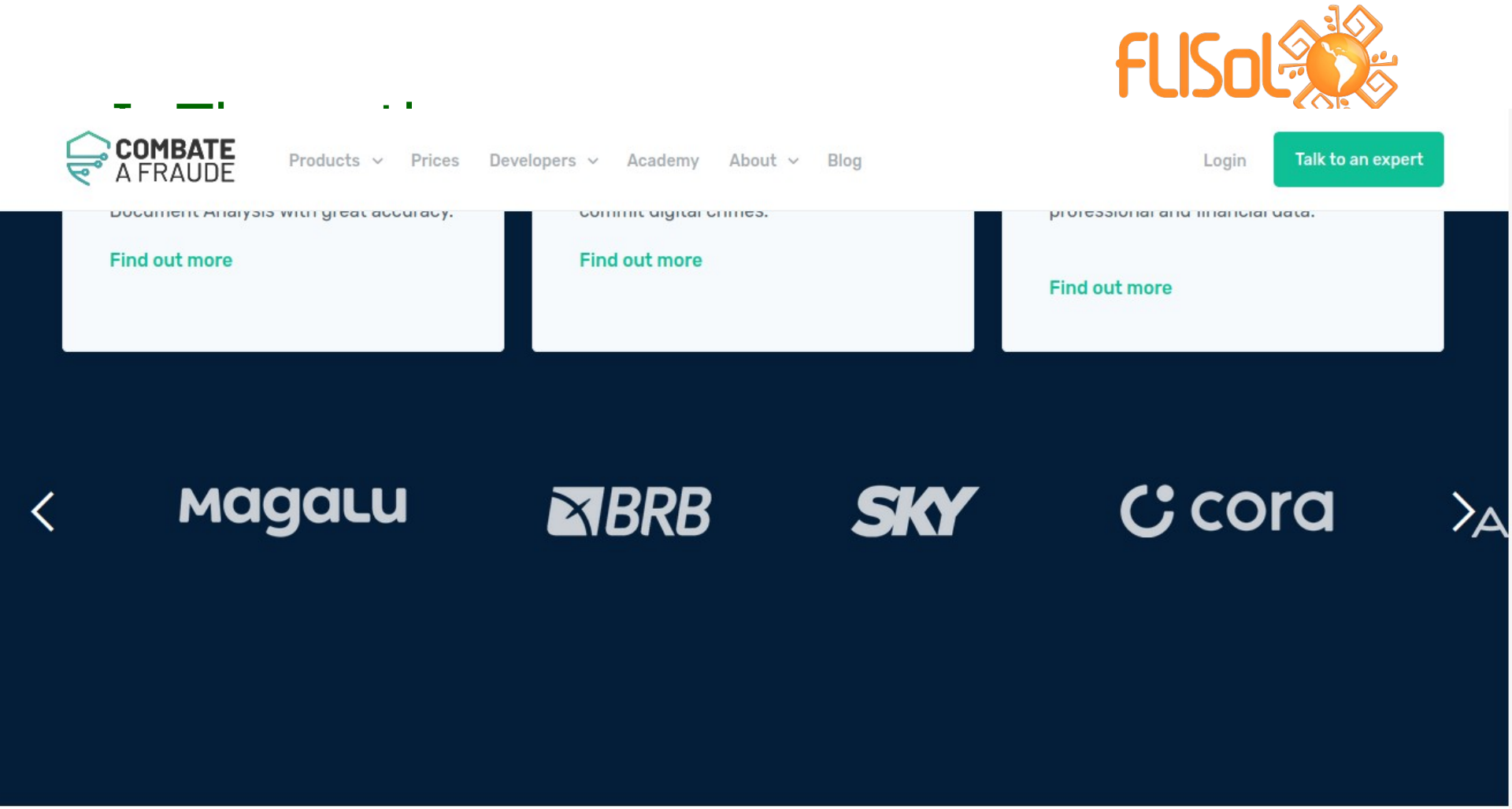
Your Counter Fraud team in the cloud

Protect your company from identity fraudsters right from Onboarding, with solutions designed for your business needs.

Platform Tour Solutions




COMBATE A FRAUDE
Products
Prices
Developers
Academy
About
Blog
Login
Talk to an expert
Automation & Onboarding
Improve your customer's enrollment experience and make your staff's job easier with a fast and reliable system that automates Proof of Life and does Document Analysis with great accuracy.
Find out more
Fight against Fraud
Protect your company from fraudulent users with this tool that determines the authenticity of documents and prevents people from impersonating others to commit digital crimes.
Find out more
Background Check
Know better who is registering in your company: employees, contractors or customers. Get access to judicial data, criminal records, phone numbers, professional and financial data.
Find out more


COMBATE A FRAUDE
Products
Prices
Developers
Academy
About
Blog
Login
Talk to an expert
Find out more
Find out more
Find out more
magalu
BRB
SKY
cora

COMBATE A FRAUDE

Products Prices Developers Academy About Blog Login Talk to an expert

DIGITAL ONBOARDING

Facilitate internal processes and your user experience

From a selfie, a photo of the front and back of the document, extract all the data from the ID/CNH, and ensure the compatibility of information. Proof of life and CPF validation are also queries of this process, as well as face match of user selfie and document photo.

Find out more

COMBATE A FRAUDE

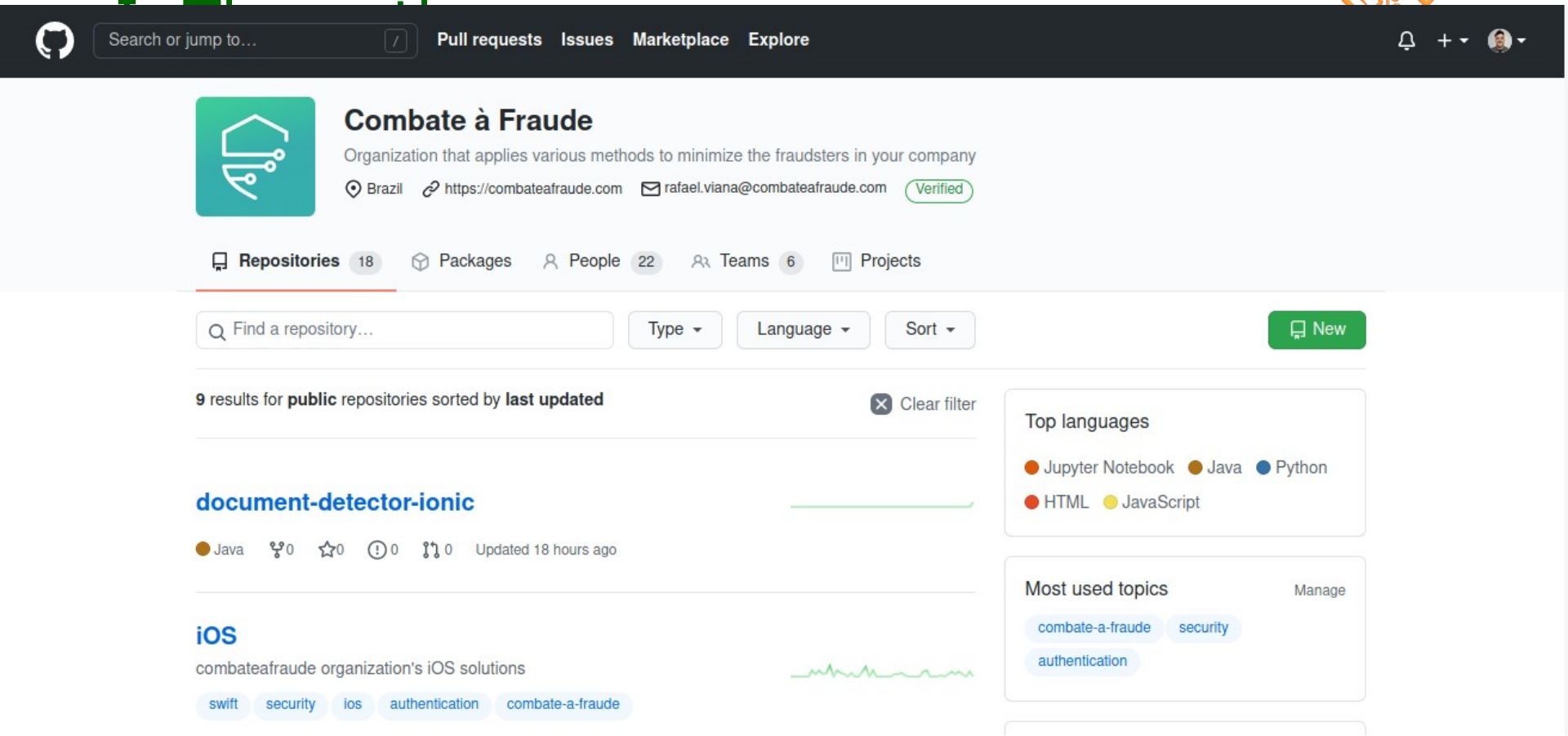
Products Prices Developers Academy About Blog Login Talk to an expert

BACKGROUND CHECK

Know your users well

Advanced analysis on the user profile focusing on specific items for your market. It is an automatic, fast and secure analysis that reduces costs and reduces the time of manual consultation on each new registration.

Get to know


Search or jump to…
Pull requests
Issues
Marketplace
Explore
Combate à Fraude
Organization that applies various methods to minimize the fraudsters in your company
Brazil
https://combateafraude.com
rafael.viana@combateafraude.com
Verified
Repositories 18
Packages
People 22
Teams 6
Projects
Find a repository…
Type
Language
Sort
New
9 results for public repositories sorted by last updated
Clear filter
document-detector-ionic
Java 0 0 0 Updated 18 hours ago
iOS
combateafraude organization's iOS solutions
swift security ios authentication combate-a-fraude
Top languages
Jupyter Notebook Java Python
HTML JavaScript
Most used topics
Manage
combate-a-fraude security
authentication

II. Briefing

Let's go!

- **OCR with Keras, TensorFlow, and Deep Learning**
- **OCR: Handwriting recognition with OpenCV, Keras, and TensorFlow**

III.

III. Concepts

In this tutorial, you will learn how to train an Optical Character Recognition (OCR) model using Keras, TensorFlow, and Deep Learning. **This post is the first in a two-part series on OCR with Keras and TensorFlow:**

- **Part 1:** *Training an OCR model with Keras and TensorFlow* (today's post)
- **Part 2:** *Basic handwriting recognition with Keras and TensorFlow* (next week's post)

For now, we'll primarily be focusing on how to train a custom Keras/TensorFlow model to recognize alphanumeric characters (i.e., the digits *0-9* and the letters *A-Z*).

III. Concepts

Figure 1: We are using two datasets for our OCR training with Keras and TensorFlow. On the *left*, we have the standard MNIST 0-9 dataset. On the *right*, we have the Kaggle A-Z dataset from Sachin Patel, which is based on the NIST Special Database 19.

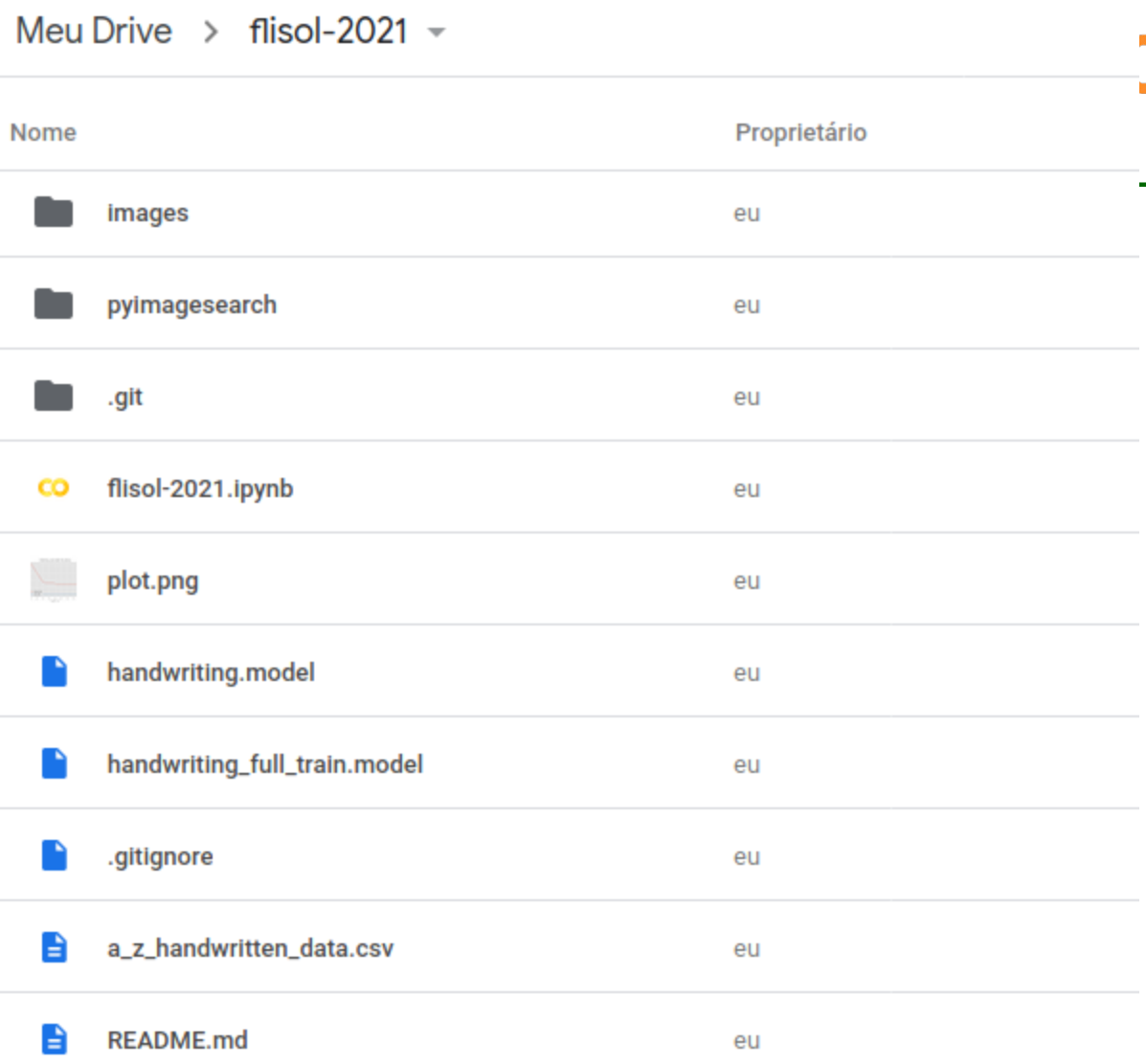
In order to train our custom Keras and TensorFlow model, we'll be utilizing **two datasets:**

- **The standard MNIST *0-9* dataset by LeCun et al.**
- **The Kaggle A-Z dataset by Sachin Patel, based on the NIST Special Database 19**

The **standard MNIST dataset** is built into popular deep learning frameworks, including Keras, TensorFlow, PyTorch, etc. A sample of the MNIST 0-9 dataset can be seen in **Figure 1 (*left*)**. The MNIST dataset will allow us to recognize the digits *0-9*. Each of these digits is contained in a *28 x 28* grayscale image. You can read more about MNIST here.

But what about the letters *A-Z*? The standard MNIST dataset doesn't include examples of the characters *A-Z — how are we going to recognize them?*

III. C


Meu Drive > flisol-2021
Nome
Proprietário
images
eu
pyimagesearch
eu
.git
eu
flisol-2021.ipynb
eu
plot.png
eu
handwriting.model
eu
handwriting_full_train.model
eu
.gitignore
eu
a_z_handwritten_data.csv
eu
README.md
eu

III. Concepts

- `pyimagesearch` module: includes the sub-modules `az_dataset` for I/O helper files and `models` for implementing the ResNet deep learning architecture
- `a_z_handwritten_data.csv` : contains the Kaggle A-Z dataset
- `handwriting.model` : where the deep learning ResNet model is saved
- `plot.png` : plots the results of the most recent run of training of ResNet
- `train_ocr_model.py` : the main driver file for training our ResNet model and displaying the results

III. Concepts

Our OCR dataset helper functions

In order to train our custom Keras and TensorFlow OCR model, we first need to implement two helper utilities that will allow us to load *both* the Kaggle A-Z datasets and the MNIST 0-9 digits from disk.

These I/O helper functions are appropriately named:

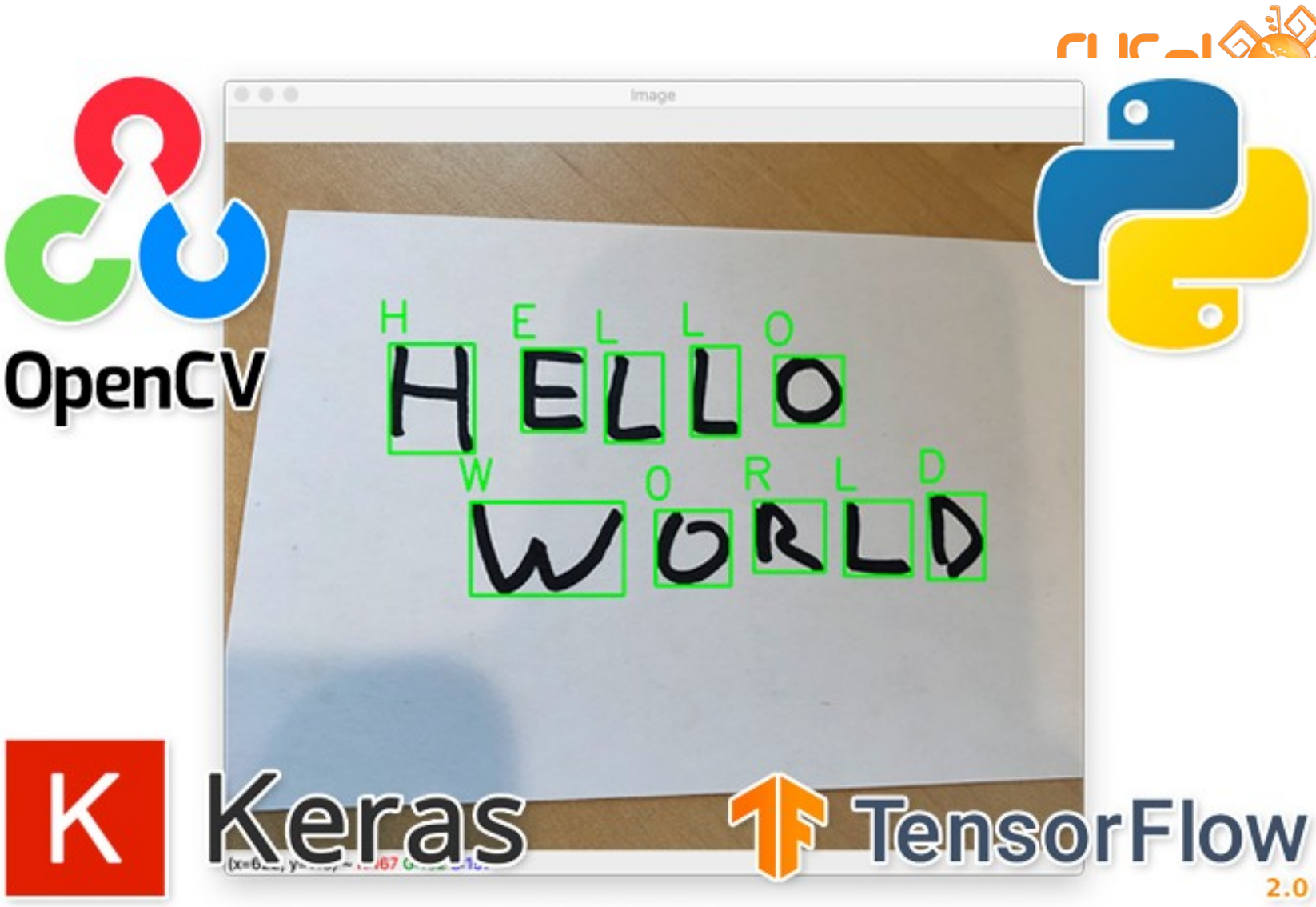
- `load_az_dataset` : for the Kaggle A-Z letters
- `load_mnist_dataset` : for the MNIST 0-9 digits

They can be found in the `helpers.py` file of `az_dataset` submodules of `pyimagesearch` .

III

Figure 3: We can see from our sample output that our Keras and TensorFlow OCR model is performing quite well in identifying our character set.

III. Concepts


OpenCV
Image
H E L L O
HELLO
W O R L D
WORLD
Keras
TensorFlow
2.0

In this tutorial, you will learn how to perform OCR handwriting recognition using OpenCV, Keras, and TensorFlow.

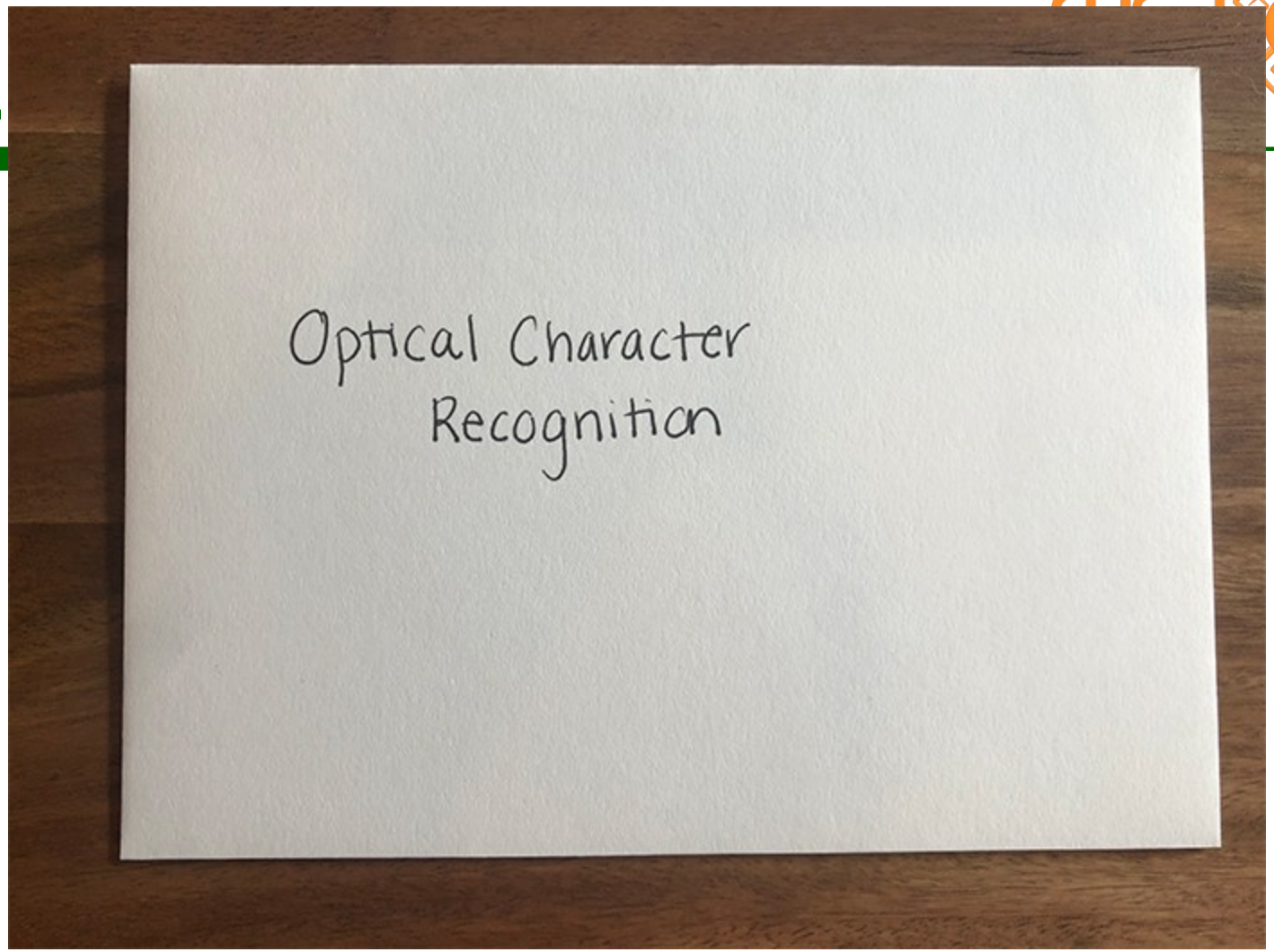
This post is **Part 2** in our two-part series on Optical Character Recognition with Keras and TensorFlow:

- **Part 1:** ***Training an OCR model with Keras and TensorFlow*** (last week's post)
- **Part 2:** *Basic handwriting recognition with Keras and TensorFlow* (today's post)

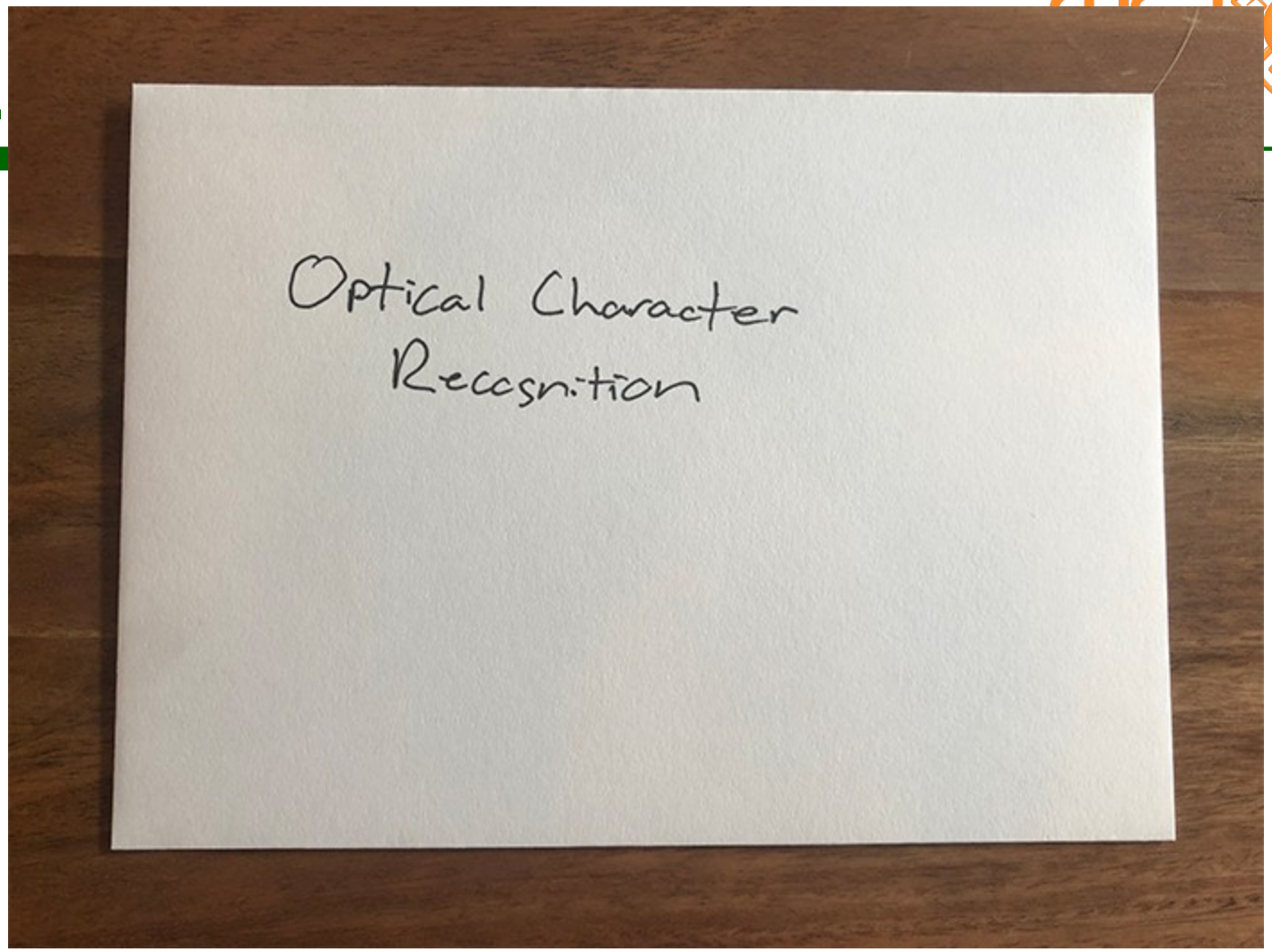
As you'll see further below, **handwriting recognition tends to be *significantly harder* than traditional OCR that uses specific fonts/characters.**

The reason this concept is so challenging is that unlike computer fonts, there are nearly *infinite variations* of handwriting styles. Every one of us has a personal style that is *specific and unique*.

III.



III.



What is handwriting recognition? And how is handwriting recognition *different* from traditional OCR?

Figure 3: OCR is more difficult for handwriting than for typed text. But why is it so difficult?

Handwriting recognition – what we've done so far

Figure 4: Here we have our two datasets from last week's post for OCR training with Keras and TensorFlow. On the *left*, we have the standard **MNIST 0-9 dataset**. On the *right*, we have the Kaggle A-Z dataset from **Sachin Patel**, which is based on the NIST Special Database 19.

III. Concepts

In last week's tutorial, we used Keras and TensorFlow to train a deep neural network to recognize both ***digits*** (*0-9*) and ***alphabetic characters*** (*A-Z*).

To train our network to recognize these sets of characters, we utilized the MNIST digits dataset as well as the **NIST Special Database 19** (for the *A-Z* characters).

Our model obtained **96% accuracy** on the testing set for handwriting recognition.

Today, we will learn how to use this model for handwriting recognition in our own custom images.

In **Figure 5**, we can see the example results from our image pre-processing steps:

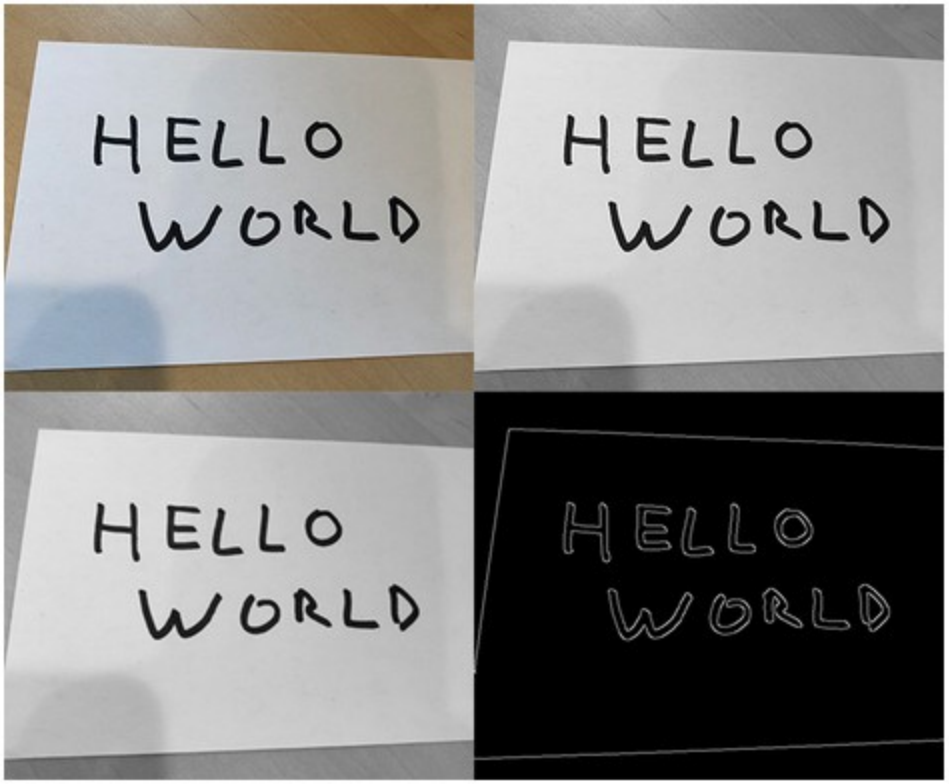


Figure 5: Here are the sample results of our OCR handwriting pre-processing pipeline, implemented with OpenCV and Python. We have our original color image (*upper-left*), our grayscale image (*upper-right*), our blurred image with reduced noise (*lower-left*), and our edge-detection map (*lower-right*).

Figure 6: True to form for programmers, we start with a *"HELLO WORLD"* example to examine the results of our deep learning OCR model. But as you can see, there are already some character recognition issues we will need to address.

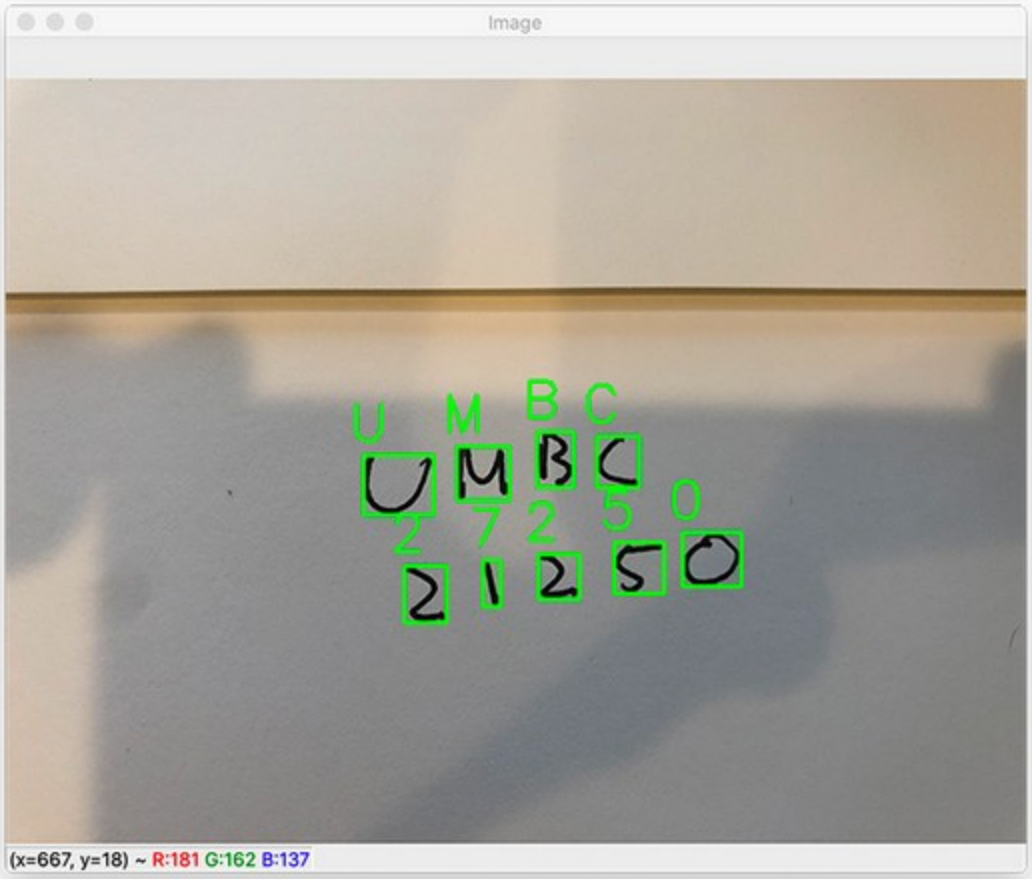


Figure 7: We use an envelope with a name and ZIP code. Not bad — nearly perfect results. The digit "1" was confused for the digit "7". If we were to apply some computer vision pre-processing, we might be able to improve our results.

III. Concepts

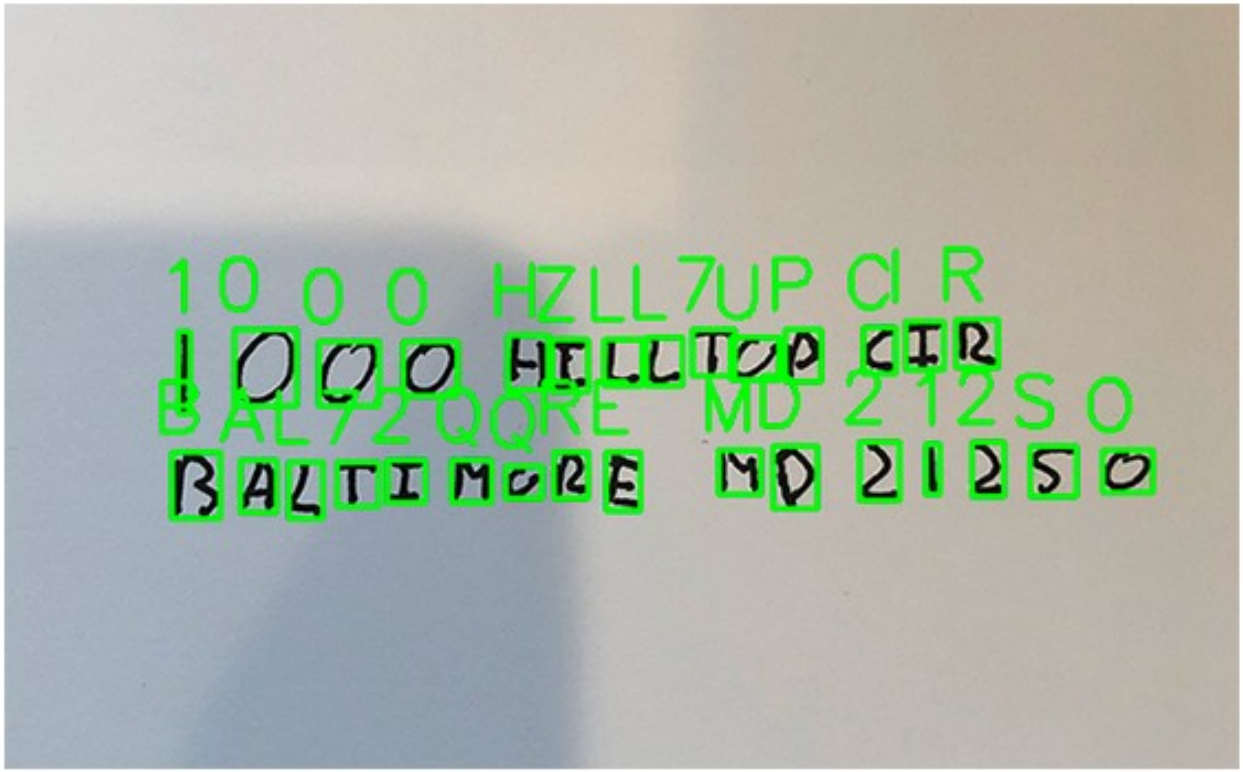


Figure 8: This time, our example is an address with two lines written on an envelope. We can see there are still multiple mistakes, and thus there are limitations to our Keras and TensorFlow OCR model.

flisol-2021.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
1 # Flisol Live 2021 - https://github.com/alexcamargoweb/flisol-2021
2 # "Inteligência Artificial aplicada: reconhecendo caracteres escritos à mão".
3 # Author: Alex Dias Camargo - https://apus.digital
4 # Adapted from: Adrian Rosebrock - https://pyimagesearch.com
5 # File: flisol-2021.ipynb
```

```
[ ]  1 # load data from google drive
     2 from google.colab import drive, output
     3 drive.mount('/content/drive')
     4 # clear terminal outputs
     5 output.clear()
     6 %cd /content/drive/MyDrive/flisol-2021/
```

```
/content/drive/MyDrive/flisol-2021
```

OCR with Keras, TensorFlow, and Deep Learning (part 1)

by PyImageSearch.com

```
[ ]  1 # tensorFlow framework installation
     2 !pip install tensorflow==2.2.0
```

Referências básicas

ROSEBROCK, Adrian. OCR with Keras, TensorFlow, and Deep Learning. Pyimagesearch. Disponível em: <https://www.pyimagesearch.com/2020/08/17/ocr-with-keras-tensorflow-and-deep-learning/>. Acesso em: 24/04/2021.

ROSEBROCK, Adrian. OCR: Handwriting recognition with OpenCV, Keras, and TensorFlow. Pyimagesearch. Disponível em: <https://www.pyimagesearch.com/2020/08/24/ocr-handwriting-recognition-with-opencv-keras-and-tensorflow/>. Acesso em: 24/04/2021.