Computer Vision for automate the process of Elections

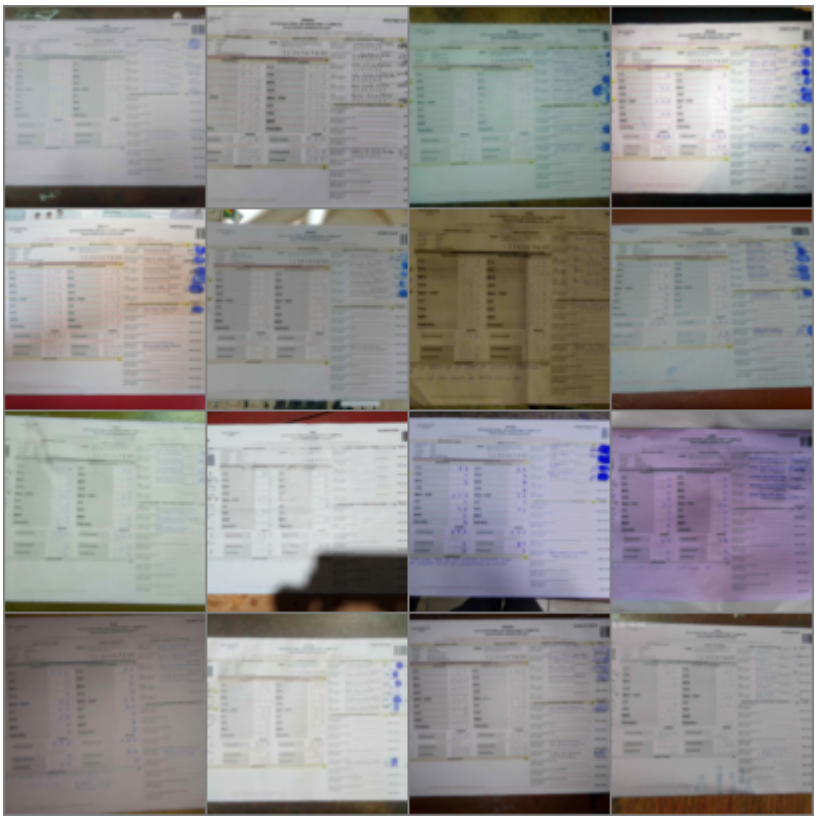
We are part of the #sg_spanish group @ Íñigo Lejarza @Cesar S. @Stanley Salvatierra @ ketcx @susyjam @Fernando T . The intention of this work is to automate the counting of votes in the electoral records. Computer vision techniques are used to attempt to read the regions of interest and subsequently read the handwritten numbers within these regions of interest. The dataset that is accessed comes from the Bolivian elections 2019. Within this dataset whose size is 60 GB. We only worked on the images that were scanned since they are easier to process due to the computer vision agorithm that was developed. A work that is left as a potential research area is to use the labels generated by the computer vision algorithm, create a VOC like dataset and train an Object Detection model, the latter was completed but with poor results in a first iteration.

Download the dataset.

To download the dataset of the images, the jupyter notebook called `DOWNLOAD_DATASET/Download_images_azure.ipynb` is used. This notebook downloads the 60 GB of information, whose categories are the following.

uploadedimages

This images comes from a direct photo from a smartphone taked the moment where the operator in charge of the elections table must to send this report for fast counts called "TREP"

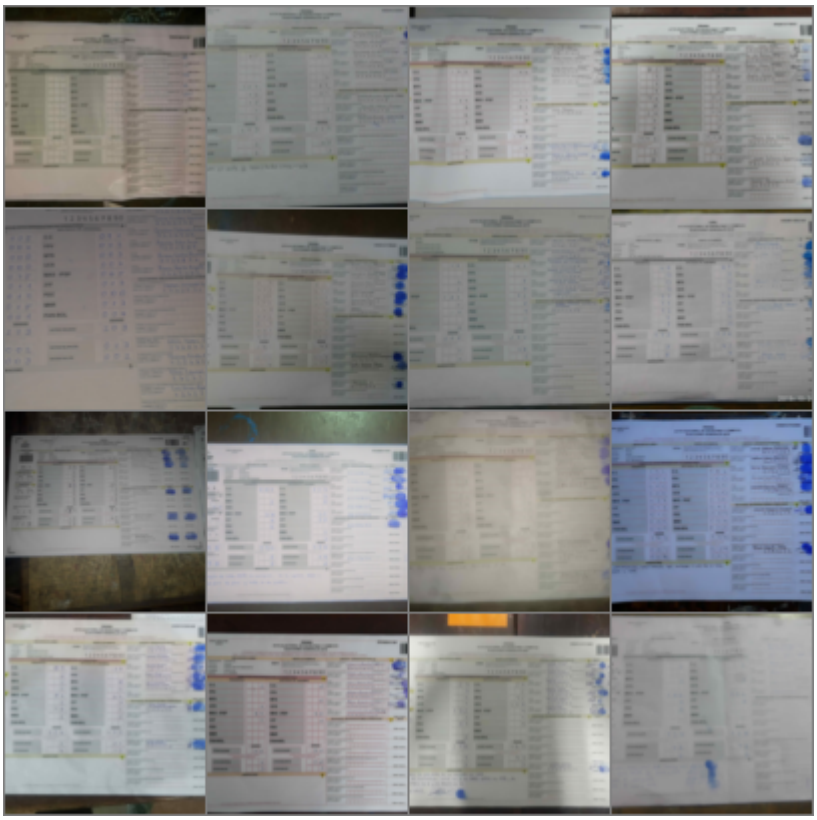


uploadedimagescomputo

This images are the same from `uploadedimages`(TREP) but are been scanned. We take this set of images (~30GB) for our Computer Vision work pipeline.

imgactastrep

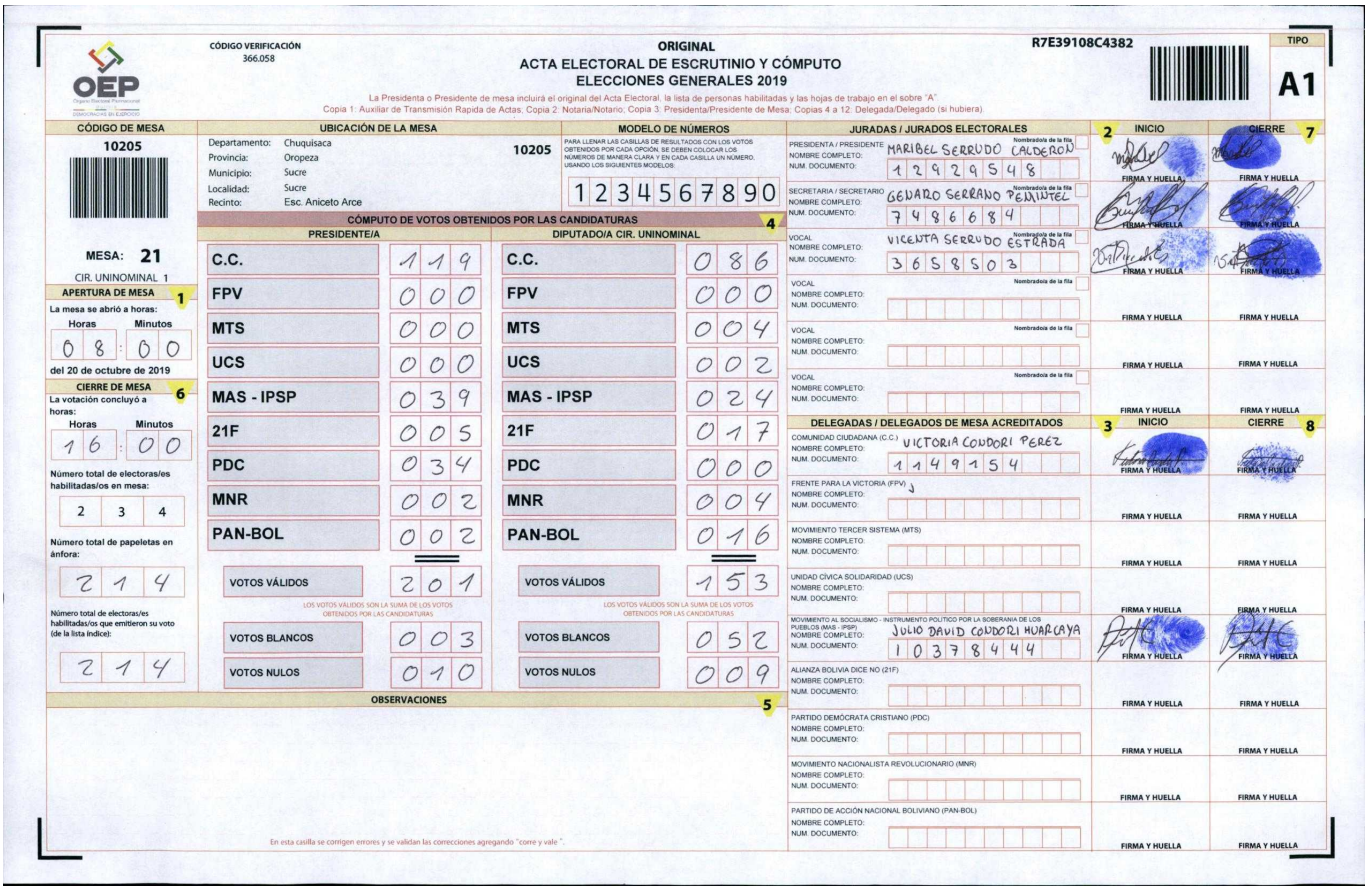
This set of images are been flaged as **fraudulent** by the original team that work with this images the last year, this set of images comes from the `uploadedimages` set.



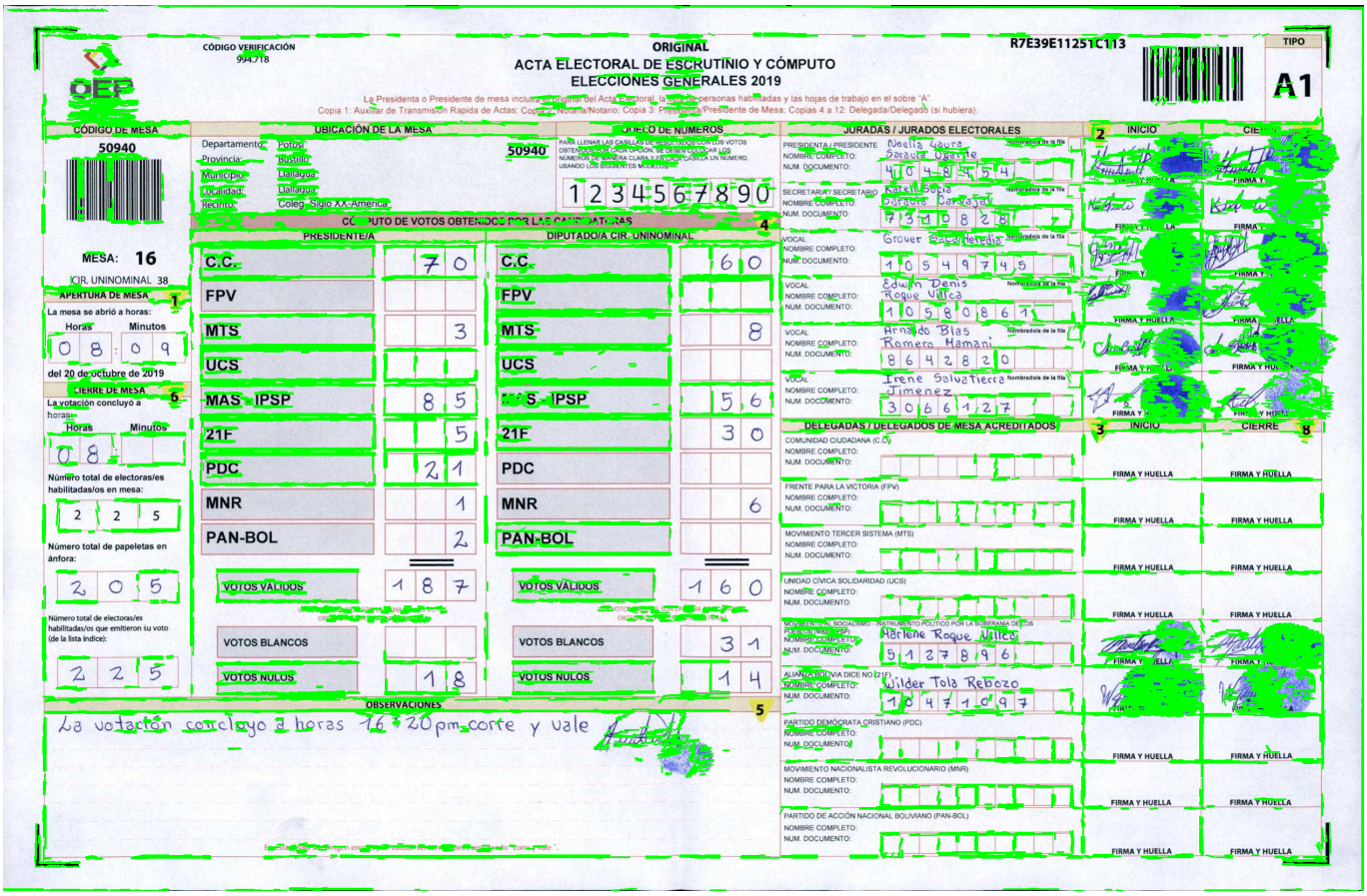
Create VOC like dataset.

Target Dataset

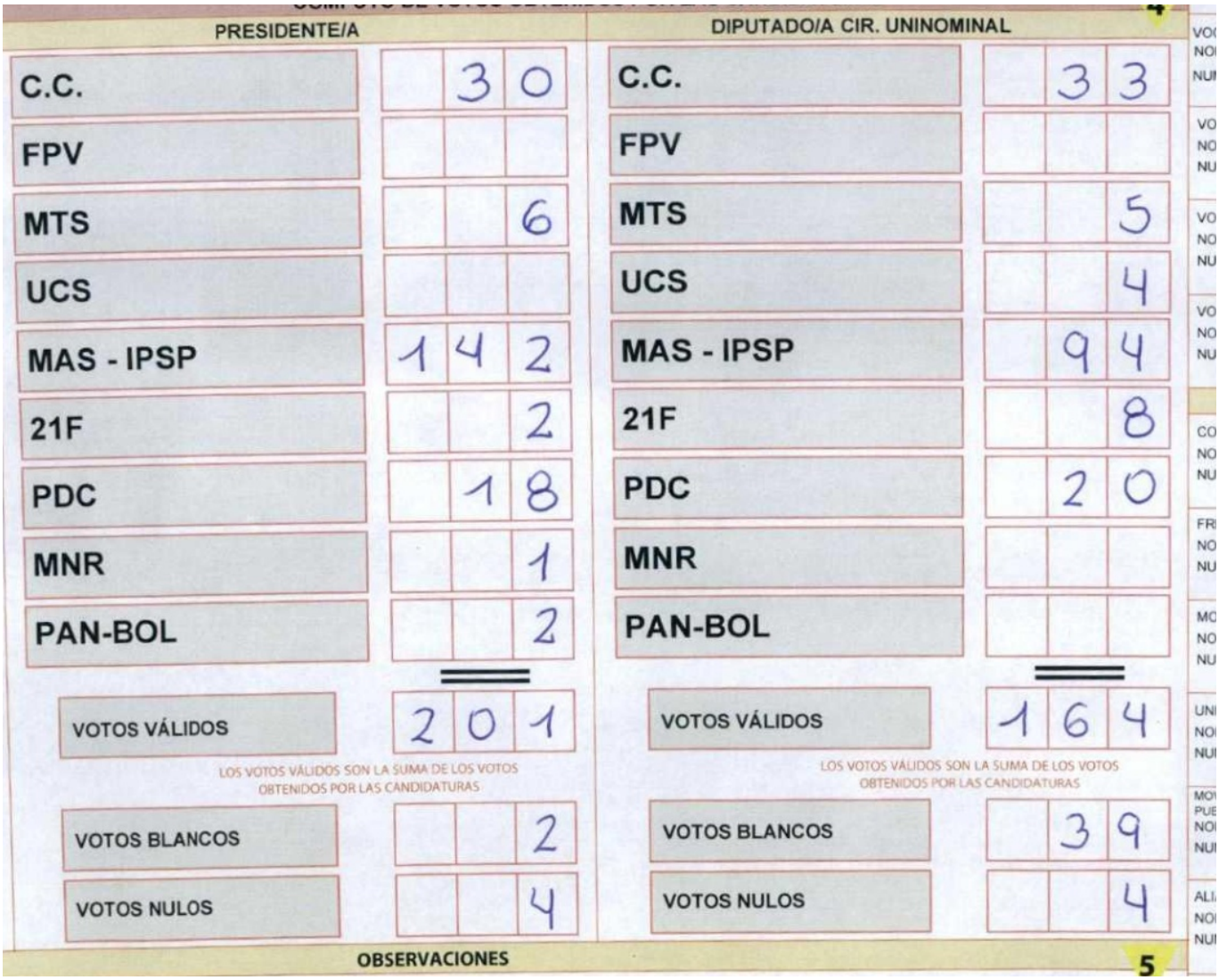
We take the `uploadedimagescomputo` set of images since this are more easy to work with Computer Vision techniques. One example of this dataset set is this image.



Using Canny edge detection and other simple computer vision techinques we can start to get some detection of the areas of interest.



Into this image we are only interested in detecting this region of the image.



Other parts of the image also can be used for another kind of applications, like "fingerprint" maching across the whole dataset or read and detect duplicate names in the region of people in charge of the election table.

This time we are only interested in counting the votes in this selected Region of Interest.

Creation of VOC Like dataset

For create this dataset we create all the steps for this into the file `VOC_CREATION/bounding_boxes_creation.py`.

With a simple.

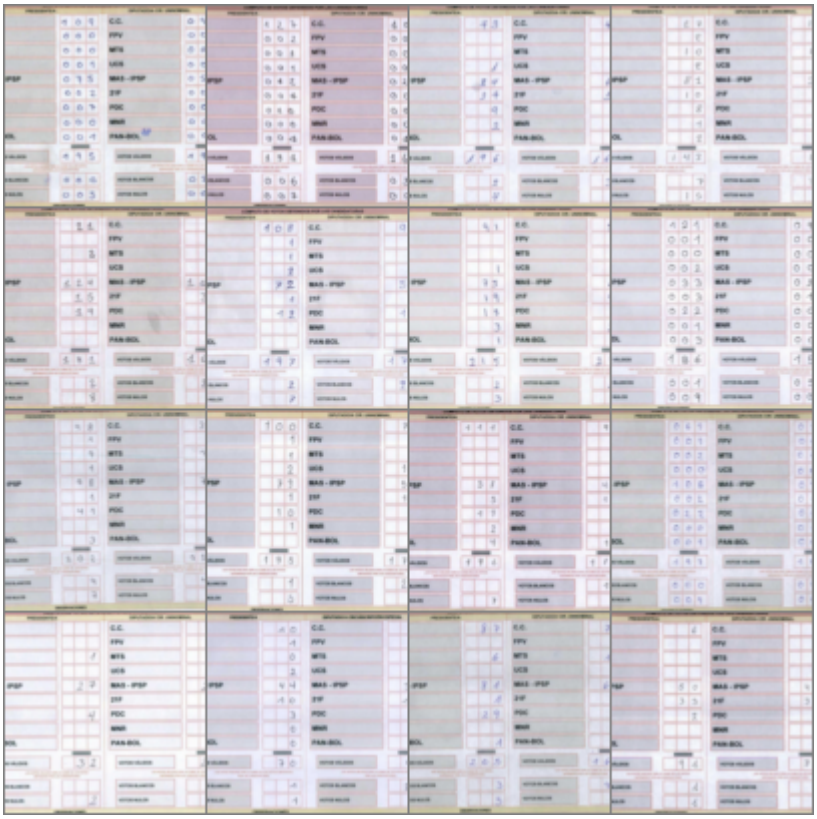
```
cd VOC_CREATION

python bounding_boxes_creation.py --data_path=$UPLOADED_IMAGES_COMPUTO
```

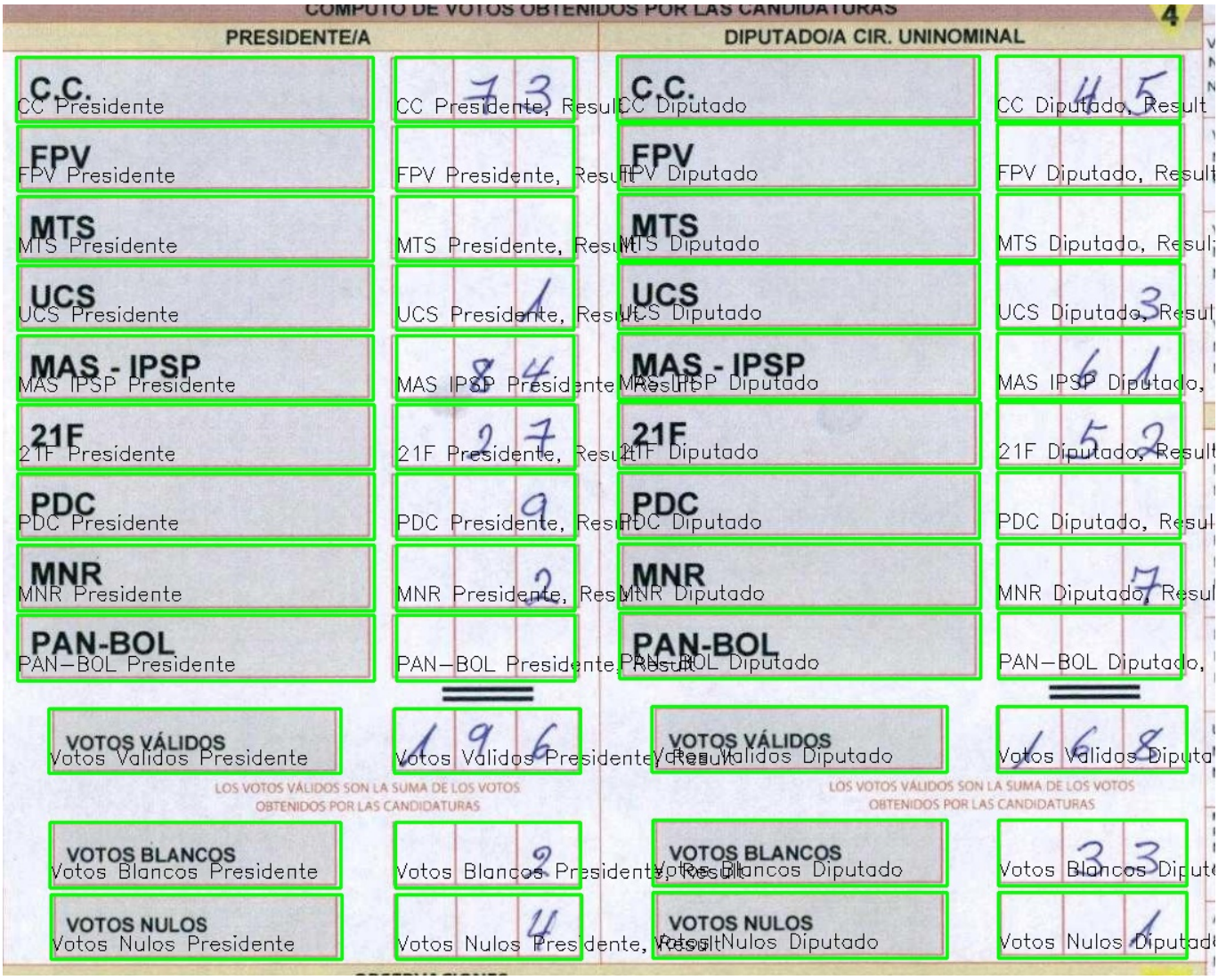
Will start to create the dataset, into a default folder called `results/`.

We made use of an env variable `$UPLOADED_IMAGES_COMPUTO` for set the directory of the target images.

The result is a set if croped images in `VOC_CREATION/results/Train/images` and his XML labels `VOC_CREATION/results/Train/labels`. ~21000 in total.



A sample with bounding boxes drawing this this.



The total success of detected boxes with the simple computer vision algorithm is aprox. `~(21000/31000)*100 ~= 68 %`. We made a try to fine tune a object detection model for increase this number and read more electoral papers.

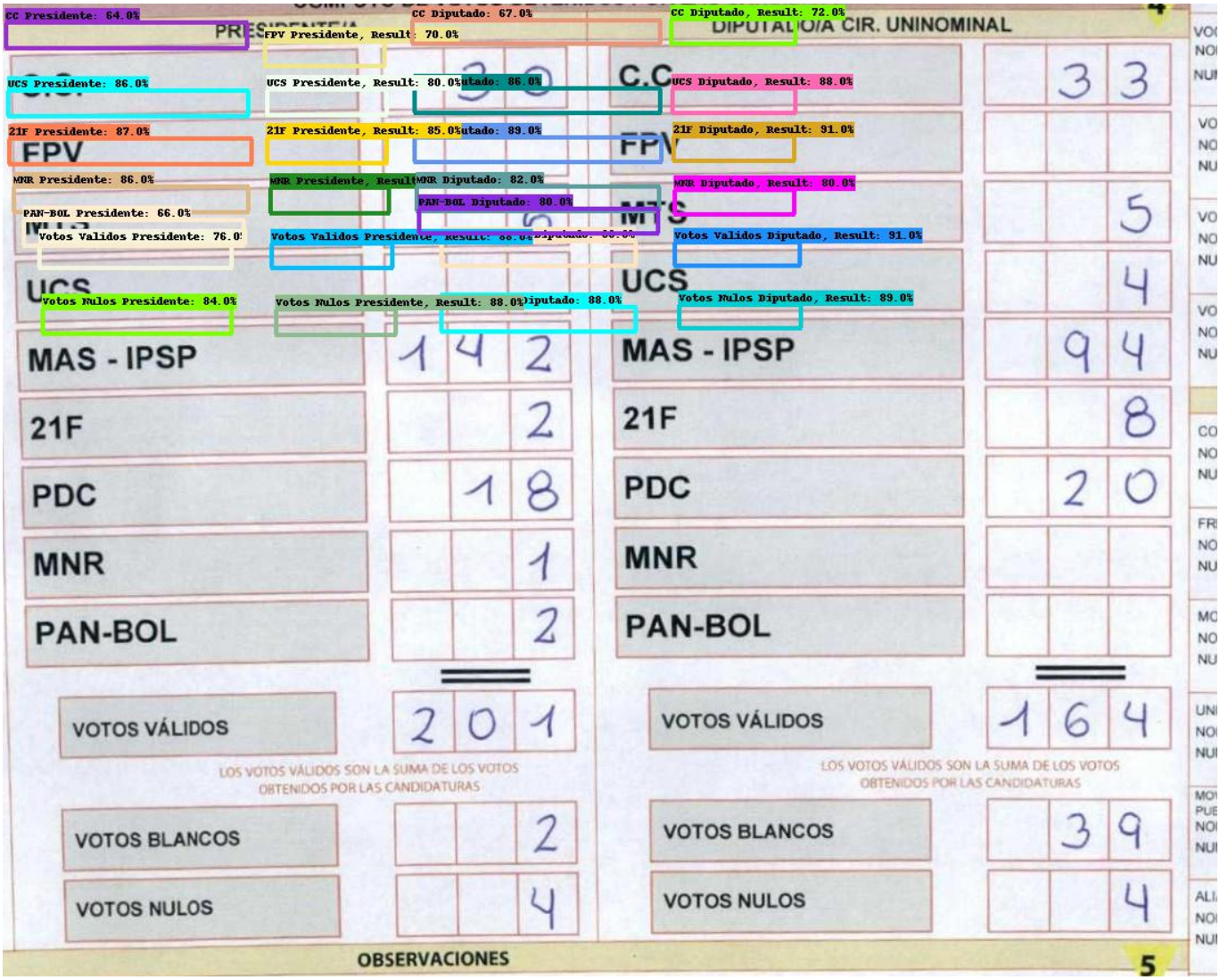
Attempt to finetune SSD MobilenetV2 with this VOC dataset.

Create TF-RECORDS FILE

We convert to tf-records format our VOC dataset using the script in `VOC_CREATION/create_tf_records.py`

Retrain the model

For fast iteration around this problem we use the MONK `https://github.com/Tessellate-Imaging/Monk_Object_Detection` library for retrain object detection models. The fork of his notebook called `Train Without Validation Dataset.ipynb` is in our folder called `OBJECT_DETECTION/Train_Without_Validation_Dataset.ipynb`. We had a problem with our credit card and we are unnable to open an azure account and train the model inside azure compute instances. For this reason we made use of google colab, we use a lot of GPU VRAM for finetune the mobilenetv2 model. The result of this finetuning is .



We think that the MONK library is not resizing correctly our image in this high level interface

We have not enought time for test the native tensorflow object detection library or another models, by this reason we leave this approach for get to work the remaining 32% of the images.

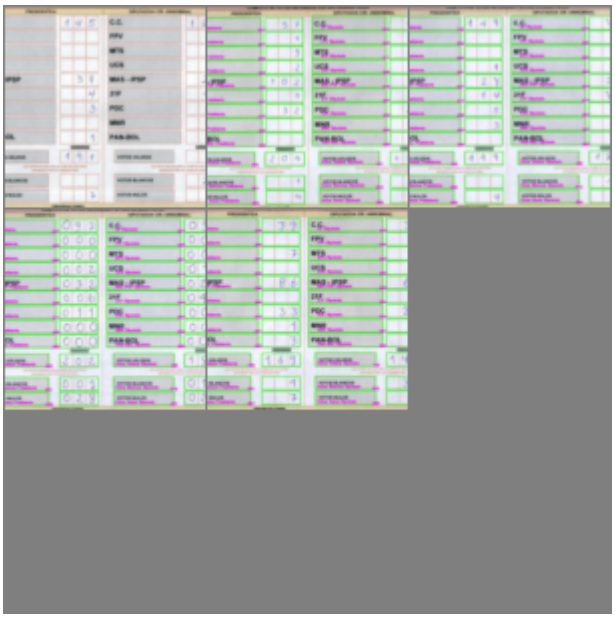
Count of votes.

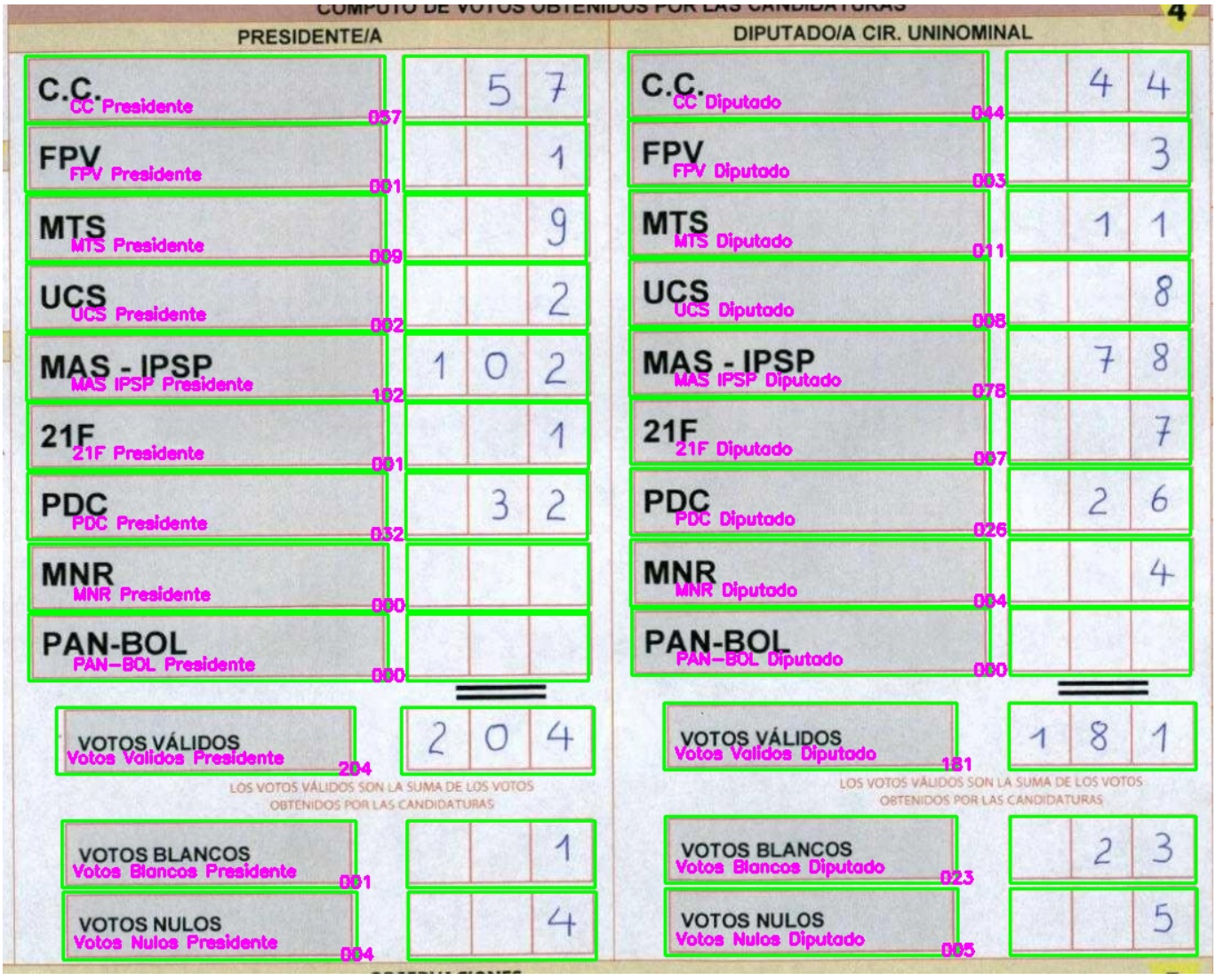
Since we labeled our Regions of Interest, we can start to count the numbers inside the boxes with another bit of work of computer vision.

We only take a sample of 5 images for proof the concept.

We create the instance `VotesCounter(ImageHanlder)` inside of the `VOTES_COUNTER/votes_counter.py` script, this instance expects some MNIST model for read the digits. We deploy a custom MNIST in our local server, we also can use any other model for read digits inside this images.

We explore the possibility to create other datasets by the side only of the digis inside this images. We will explore this alternative later. For now we obtain the followinig results with this simple MNIST.





And the log for the count into the `VOTES_COUNTER/results_votes/results_log.txt`

```
image,count_value, user_id
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,004,1
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,005,2
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,001,3
```

```
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,023,4
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,204,5
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,181,6
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,000,7
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,000,8
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,000,9
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,004,10
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,032,11
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,026,12
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,001,13
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,007,14
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,102,15
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,078,16
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,002,17
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,008,18
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,009,19
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,011,20
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,001,21
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,003,22
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,057,23
00c354f2-f7ce-11e9-95e8-c8ff28027534.jpg,044,24
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,004,1
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,004,2
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,000,3
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,013,4
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,199,5
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,186,6
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,000,7
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,000,8
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,003,9
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,007,10
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,005,11
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,004,12
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,014,13
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,073,14
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,027,15
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,024,16
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,001,17
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,004,18
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,000,19
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,002,20
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,000,21
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,000,22
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,149,23
00151e8d-f7dd-11e9-b71f-c8ff28027534.jpg,072,24
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,028,1
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,027,2
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,001,3
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,019,4
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,202,5
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,185,6
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,000,7
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,000,8
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,000,9
```

```
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,000,10
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,011,11
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,004,12
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,006,13
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,044,14
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,032,15
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,025,16
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,002,17
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,014,18
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,000,19
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,001,20
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,000,21
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,000,22
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,097,23
011e2371-f7e3-11e9-8f53-c8ff28027534.jpg,097,24
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,007,1
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,005,2
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,001,3
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,032,4
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,169,5
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,140,6
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,003,7
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,000,8
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,001,9
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,001,10
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,033,11
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,024,12
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,000,13
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,004,14
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,086,15
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,065,16
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,000,17
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,009,18
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,007,19
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,003,20
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,000,21
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,004,22
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,039,23
012c5403-f7ce-11e9-b2ea-c8ff28027534.jpg,030,24
```

which later can be processed with pandas for further analysis.

Conclusions

With this work we wanted to demonstrate the proof of concept that it is possible to automatically read the digits written by hand within physical electoral records. The procedure ranges from using simple computer vision techniques to exploring the possibility of creating an object detector using deep learning. The latter needs more work and it is hoped that this can be achieved. We also leave open the possibility of using this dataset for other types of work that can be done on top of the images that are available.