

# Computer Vision for automate the process of Elections

---

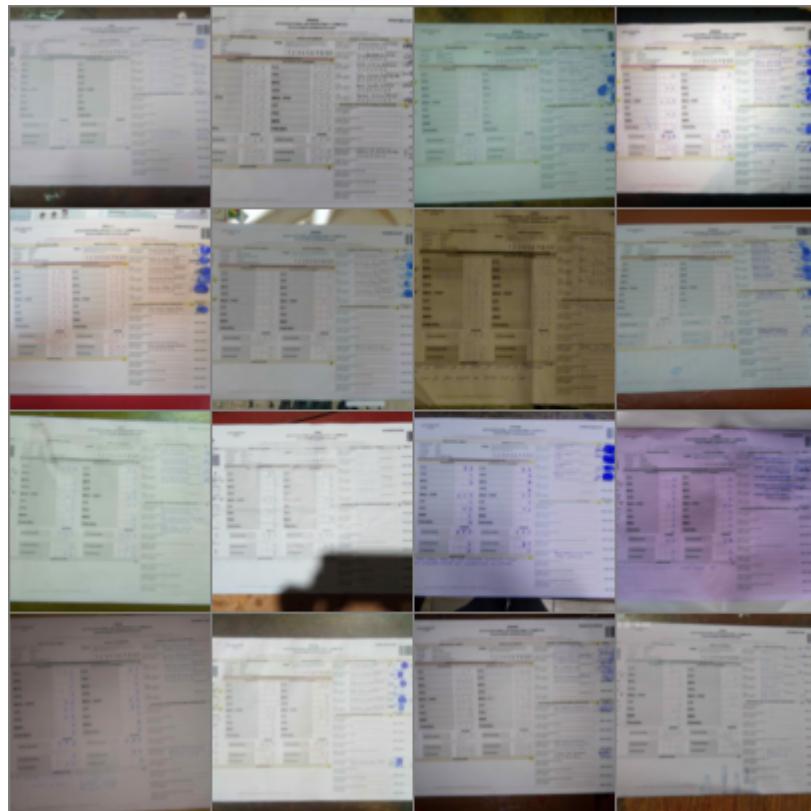
We are part of the #sg\_spanish group @ Íñigo Lejarza @Cesar S. @Stanley Salvatierra @ ketcx @susyjam @Fernando T. The intention of this work is to automate the counting of votes in the electoral records. Computer vision techniques are used to attempt to read the regions of interest and subsequently 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 algorithm 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"



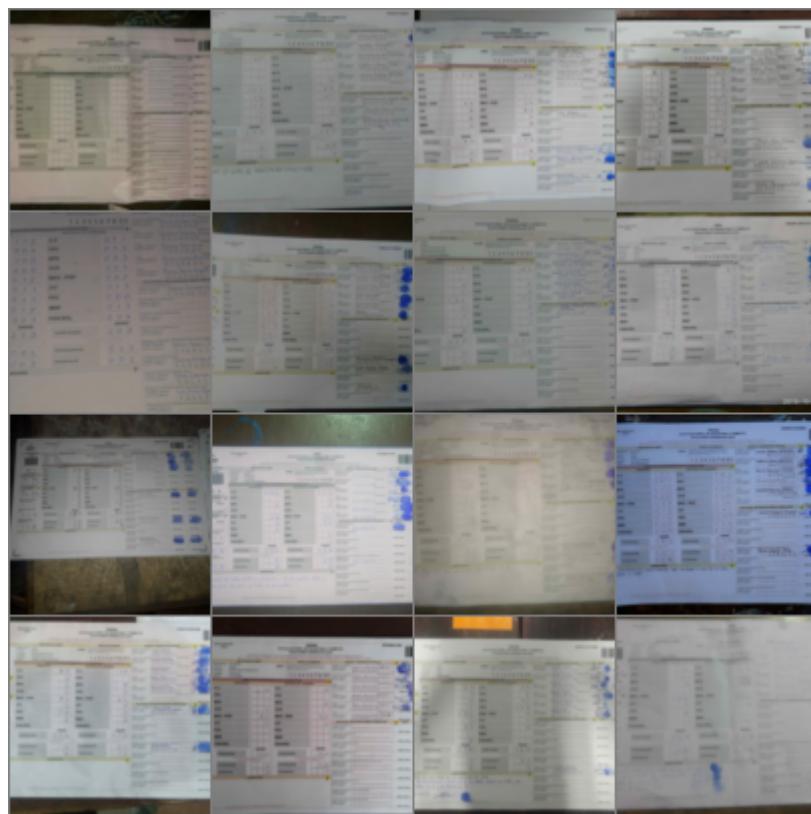
### uploadedimagescomputo

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 [uploadedimagescompute](#) 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 techniques we can start to get some detection of the areas of interest.

**ORIGINAL**

**ACTA ELECTORAL DE ESCRUTINIO Y CÓMPUTO**  
**ELECCIONES GENERALES 2019**

CÓDIGO VERIFICACIÓN  
594718

**JURADAS / JURADOS ELECTORALES**

R7E39E1125T113

**TIPO**  
**A1**

La Presidenta o Presidente de mesa incluye el número del Acta original, las personas habilitadas y las hojas de trabajo en el sobre "A".  
Copia 1: Auxiliar de Transmisión Rápida de Actas; Copia 2: Notario; Copia 3: Presidente de Mesa; Copias 4 a 12: Delegada/Delegado (si hubiera).

<b>CÓDIGO DE MESA</b>		<b>UBICACIÓN DE LA MESA</b>		<b>CÓDIGO DE NÚMEROS</b>		<b>JURADAS / JURADOS ELECTORALES</b>	
Departamento:	Bolívar	Provincia:	Zarzal	50940	PARA LLLENAR LAS CASILLAS CON LOS VOTOS DIFERENTES. AL VOTACIÓN SE DEBERÁ HACER UNA MARCA EN LA CASILLA CORRESPONDIENTE AL NÚMERO USANDO LOS DIFERENTES MATERIALES.	Presidenta / Presidente Dra. Luisa Gómez Documentos: 10475754	INICIO
Municipio:	Zarzal	Distrito:	Zarzal		NOMBRE COMPLETO: NUM. DOCUMENTO:	2do Jurado: Don José Documentos: 131824	CIERRE
Mesidad:	Colombia	Localidad:	Zarzal		NOMBRE COMPLETO: NUM. DOCUMENTO:	3er Jurado: Don Juan Documentos: 131824	
Rección:	Coleg. Sagio XX-Amerca				NOMBRE COMPLETO: NUM. DOCUMENTO:	4to Jurado: Don Juan Documentos: 131824	
<b>MESA: 16</b>	<b>CIR. UNINOMINAL 38</b>	<b>APERTURA DE MESA</b>	<b>1</b>	<b>CÓDIGO DE VOTOS OBTENIDOS POR LAS CANDIDATURAS</b>	<b>4</b>	<b>GROVER</b>	<b>FIRMA</b>
La mesa se abrió a horas: Horas	08	Minutos	09	PRESIDENTE/CIR. UNINOMINAL		Edwin Dennis Roque Ulloca	FIRMA
del 20 de octubre de 2019				C.C.	160	Hans de Bias Romero Ramón	FIRMA
<b>CIERRE DE MESA</b>	<b>6</b>	La votación concluyó a horas: Horas	08	FPV		Irene Solache Sierra	FIRMA
		Minutos	08	MTS	8	Nombres de la fila	FIRMA
				UCS		Jiménez	FIRMA
				MAS-IPSP	56	Documentos: 30661217	FIRMA
				21F	30		FIRMA
				PDC			FIRMA
				MNR	6		FIRMA
				PAN-BOL			FIRMA
				VOTOS VÁLIDOS	160		FIRMA
				VOTOS BLANCOS	31		FIRMA
				VOTOS NULOS	114		FIRMA
				OBSERVACIONES			
La votación concluyó 2 horas 16:20 pm, corte y vale							

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

PRESIDENTE/A		DIPUTADO/A CIR. UNINOMINAL	
C.C.	30	C.C.	33
FPV		FPV	
MTS		MTS	5
UCS		UCS	4
MAS - IPSP	142	MAS - IPSP	94
21F	2	21F	8
PDC	18	PDC	20
MNR	1	MNR	
PAN-BOL	2	PAN-BOL	
<hr/>		<hr/>	
VOTOS VÁLIDOS	201	VOTOS VÁLIDOS	164
LOS VOTOS VÁLIDOS SON LA SUMA DE LOS VOTOS OBTENIDOS POR LAS CANDIDATURAS			
VOTOS BLANCOS	2	VOTOS BLANCOS	39
VOTOS NULOS	4	VOTOS NULOS	4
OBSERVACIONES			5

Other parts of the image also can be used for another kind of applications, like "fingerprint" matching 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 of cropped 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.

COMPUTO DE VOTOS OBTENIDOS POR LAS CANDIDATURAS		DIPUTADO/A CIR. UNINOMINAL	
PRESIDENTE/A		DIPUTADO/A CIR. UNINOMINAL	
<b>C.C.</b> CC Presidente	13	<b>C.C.</b> CC Diputado	115
<b>FPV</b> FPV Presidente		<b>FPV</b> FPV Diputado	
<b>MTS</b> MTS Presidente		<b>MTS</b> MTS Diputado	
<b>UCS</b> UCS Presidente	1	<b>UCS</b> UCS Diputado	3
<b>MAS - IPSP</b> MAS IPSP Presidente	84	<b>MAS - IPSP</b> MAS IPSP Diputado	61
<b>21F</b> 21F Presidente	27	<b>21F</b> 21F Diputado	52
<b>PDC</b> PDC Presidente	9	<b>PDC</b> PDC Diputado	
<b>MNR</b> MNR Presidente	2	<b>MNR</b> MNR Diputado	7
<b>PAN-BOL</b> PAN-BOL Presidente		<b>PAN-BOL</b> PAN-BOL Diputado	
<b>VOTOS VÁLIDOS</b> Votos Validos Presidente	196	<b>VOTOS VÁLIDOS</b> Votos Validos Diputado	168
LOS VOTOS VÁLIDOS SON LA SUMA DE LOS VOTOS OBTENIDOS POR LAS CANDIDATURAS			
<b>VOTOS BLANCOS</b> Votos Blancos Presidente	9	<b>VOTOS BLANCOS</b> Votos Blancos Diputado	33
<b>VOTOS NULOS</b> Votos Nulos Presidente	4	<b>VOTOS NULOS</b> Votos Nulos Diputado	1

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

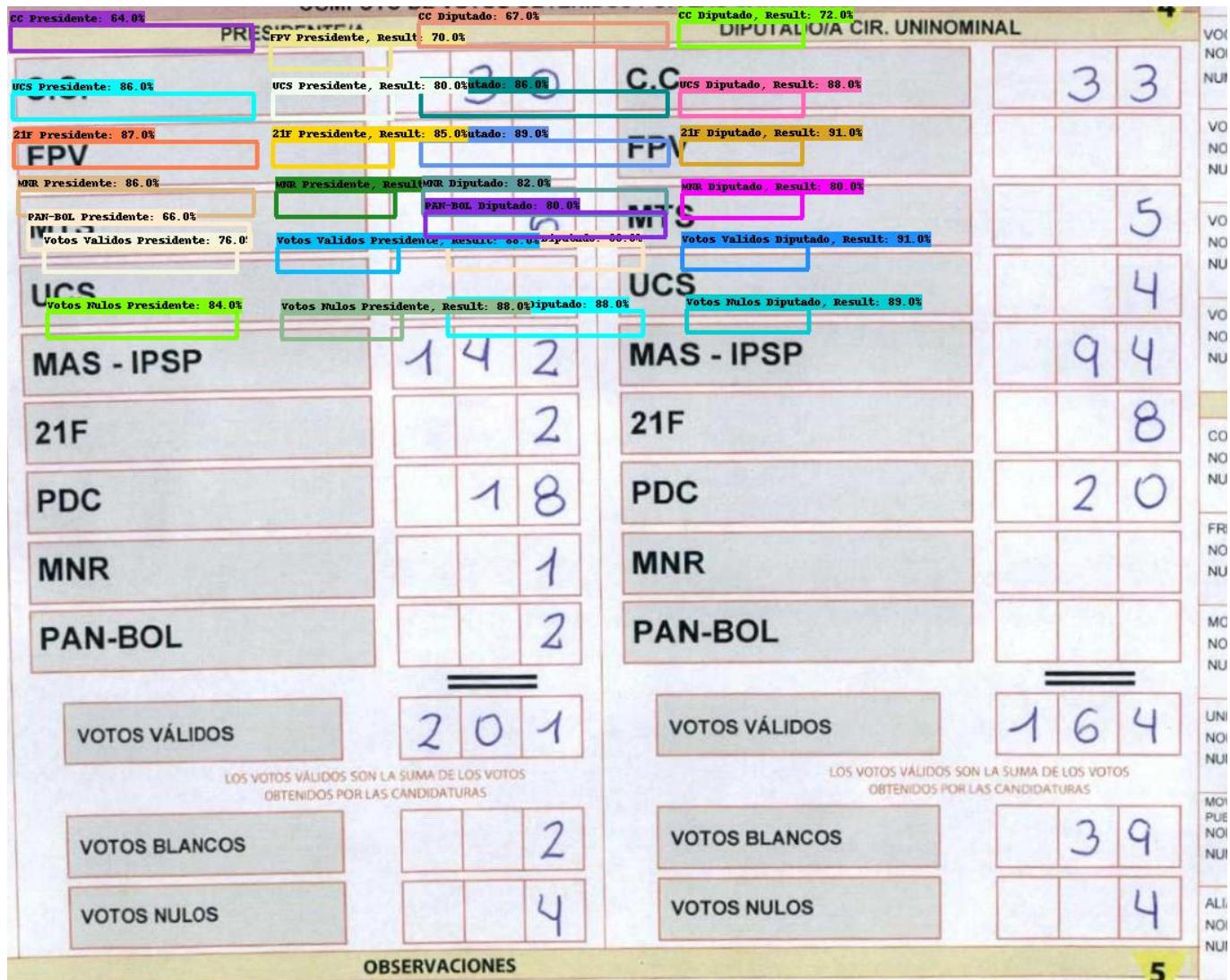
Attempt to finetune SSD MobilenetV2 with this VOC dataset.

### 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](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 unable 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 enough 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.

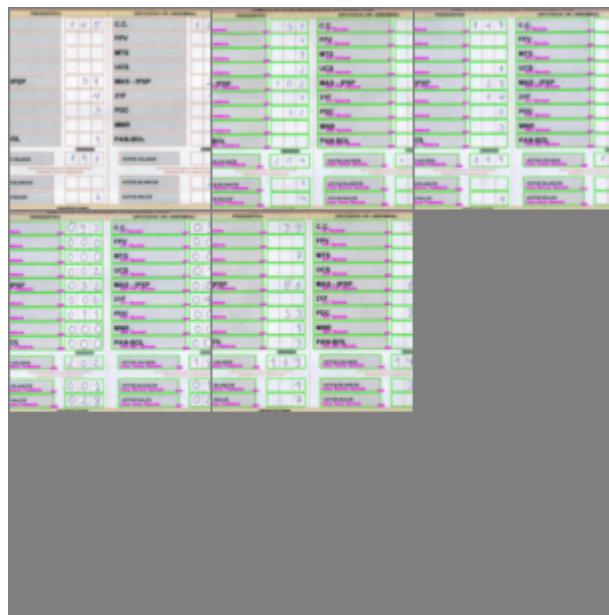
## Count of votes.

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(ImageHandler)` 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 digits inside this images. We will explore this alternative later. For now we obtain the following results with this simple MNIST.



PRESIDENTE/A		DIPUTADO/A CIR. UNINOMINAL		
C.C. CC Presidente	057	5	7	4 4
FPV FPV Presidente	001		1	3
MTS MTS Presidente	009		9	1 1
UCS UCS Presidente	002		2	8
MAS - IPSP MAS IPSP Presidente	102	1	0	2
21F 21F Presidente	001		1	7
PDC PDC Presidente	032		3	2
MNR MNR Presidente	000			4
PAN-BOL PAN-BOL Presidente	000			1 8 1
<b>VOTOS VÁLIDOS</b> Votos Validos Presidente		2	0	4
LOS VOTOS VÁLIDOS SON LA SUMA DE LOS VOTOS OBTENIDOS POR LAS CANDIDATURAS				
VOTOS BLANCOS Votos Blancos Presidente	001		1	2 3
VOTOS NULOS Votos Nulos Presidente	004		4	5
<b>VOTOS VÁLIDOS</b> Votos Validos Diputado		181	LOS VOTOS VÁLIDOS SON LA SUMA DE LOS VOTOS OBTENIDOS POR LAS CANDIDATURAS	
VOTOS BLANCOS Votos Blancos Diputado	023			
VOTOS NULOS Votos Nulos Diputado	005			

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.