

# Computer Vision for automate the process of Elections

---

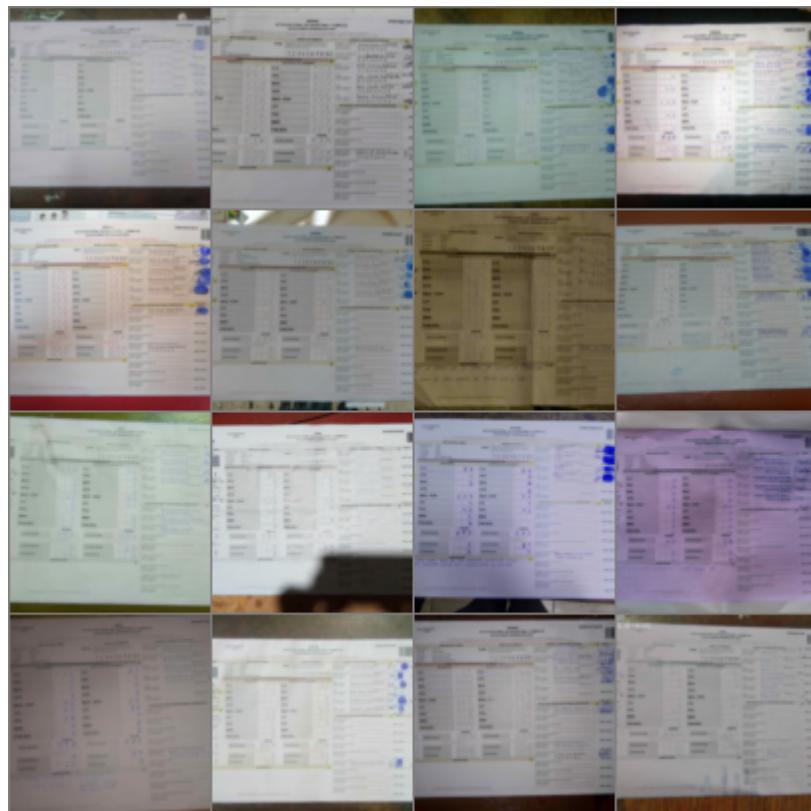
The intention of this work is to automate the counting of votes in the electoral records. Computer vision techniques are used to attempt to read the regions of interest and subsequently read the handwritten numbers within these regions of interest. The dataset that is accessed comes from the Bolivian 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"



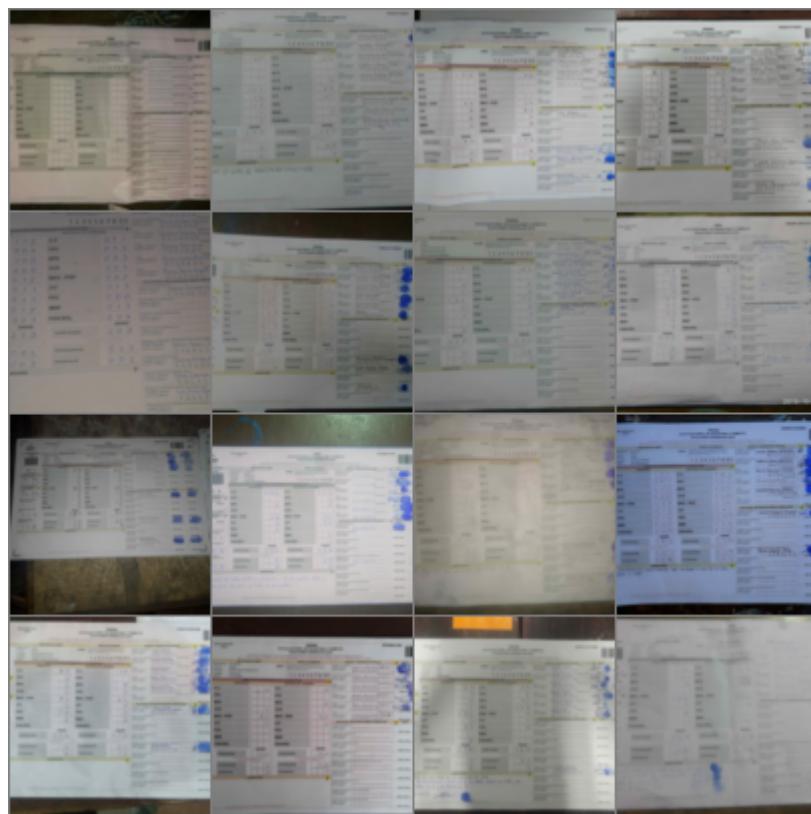
### uploadedimagescomputo

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 [uploadedimagescompto](#) 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.

CÓDIGO VERIFICACIÓN 994718		ORIGINAL		R7E39E11251C113		TIPO A1	
		ACTA ELECTORAL DE ESCRUTINIO Y CÓMPUTO ELECCIONES GENERALES 2019					
		<small>La Presidenta o Presidente de mesa incluye la copia 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: Notaria; Copia 3: Presidente de Mesa; Copias 4 a 12: Delegado/Delegada (si hubiera).</small>					
CÓDIGO DE MESA 50940		UBICACIÓN DE LA MESA		JUICIO DE NÚMEROS		JURADAS / JURADOS ELECTORALES	
Departamento: Potosí Provincia: Tarija Municipio: Tarija Distrito: Tarija Localidad: Colos, Barrio XX Lámparas		50940 <small>PARA LLLENAR LAS CASILLAS SE DEBEN USAR LOS VOTOS NUMEROS DE LA PLUMA CLAVILLO, PINTAR CON UN NÚMERO, USANDO LOS SIGUIENTES MODELOS:</small>		1 2 3 4 5 6 7 8 9 0		PRESIDENTA / PRESIDENTE: Doña Luisa Gómez SECRETARIO / SECRETARIA: Luisa Gómez NUM. DOCUMENTO: 11104-935754 VOCAL: Gómez Luisa	
MESAS: 16		C.C. 50940		DIPUTADO A CIR-UNINOMINAL		2 INICIO CIERRE	
KIR UNINOMINAL 38		C.C. 50940 FPV MTS UCS MAS-IPSP 21F PDC MNR PAN-BOL		C.C. 50940 FPV MTS UCS MAS-IPSP 21F PDC MNR PAN-BOL		1 2 3 4 5 6 7 8 9 0 10 5 4 9 7 4 5 10 5 8 0 6 6 1 1 10 6 4 2 8 2 0 10 6 6 1 2 7 1 10 4 7 1 0 9 7	
APERTURA DE MESA: 1		Horas: 08 Minutos: 09 <small>del 20 de Octubre de 2019</small>		Horas: 15 Minutos: 00		Horas: 16 Minutos: 20	
CIERRE DE MESA: 5							
La votación concluyó a horas: 16:20 pm							
Número total de electores/es habilitados/as en mesa:		VOTOS VALIDOS: 187 VOTOS BLANCOS: 31 VOTOS NULOS: 114		VOTOS VALIDOS: 160 VOTOS BLANCOS: 31 VOTOS NULOS: 114		Firma: Wilder Tola Rebozo Firma: Irene Salvatierra Jiménez	
Número total de papeletas en urna:		2 0 5		2 0 5		Firma: Wilder Tola Rebozo Firma: Irene Salvatierra Jiménez	
Número total de electores/es habilitados/as que emitieron su voto (de la lista indicada):		2 2 5		2 2 5		Firma: Wilder Tola Rebozo Firma: Irene Salvatierra Jiménez	
OBSERVACIONES:		La votación concluyó a 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](#).

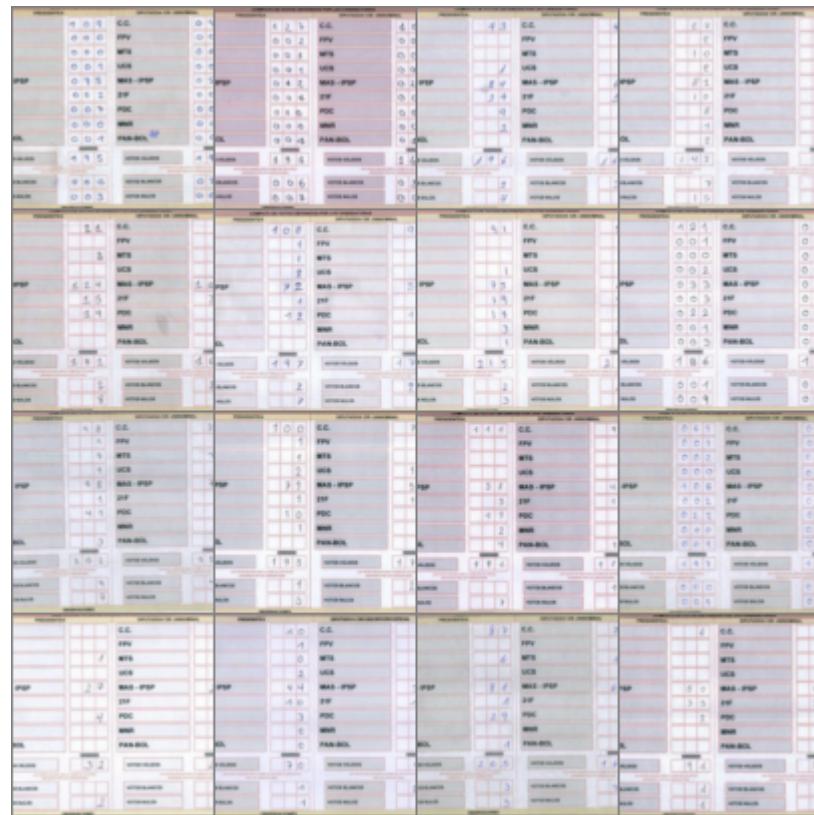
With a simple.

```
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 their 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	73	<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.

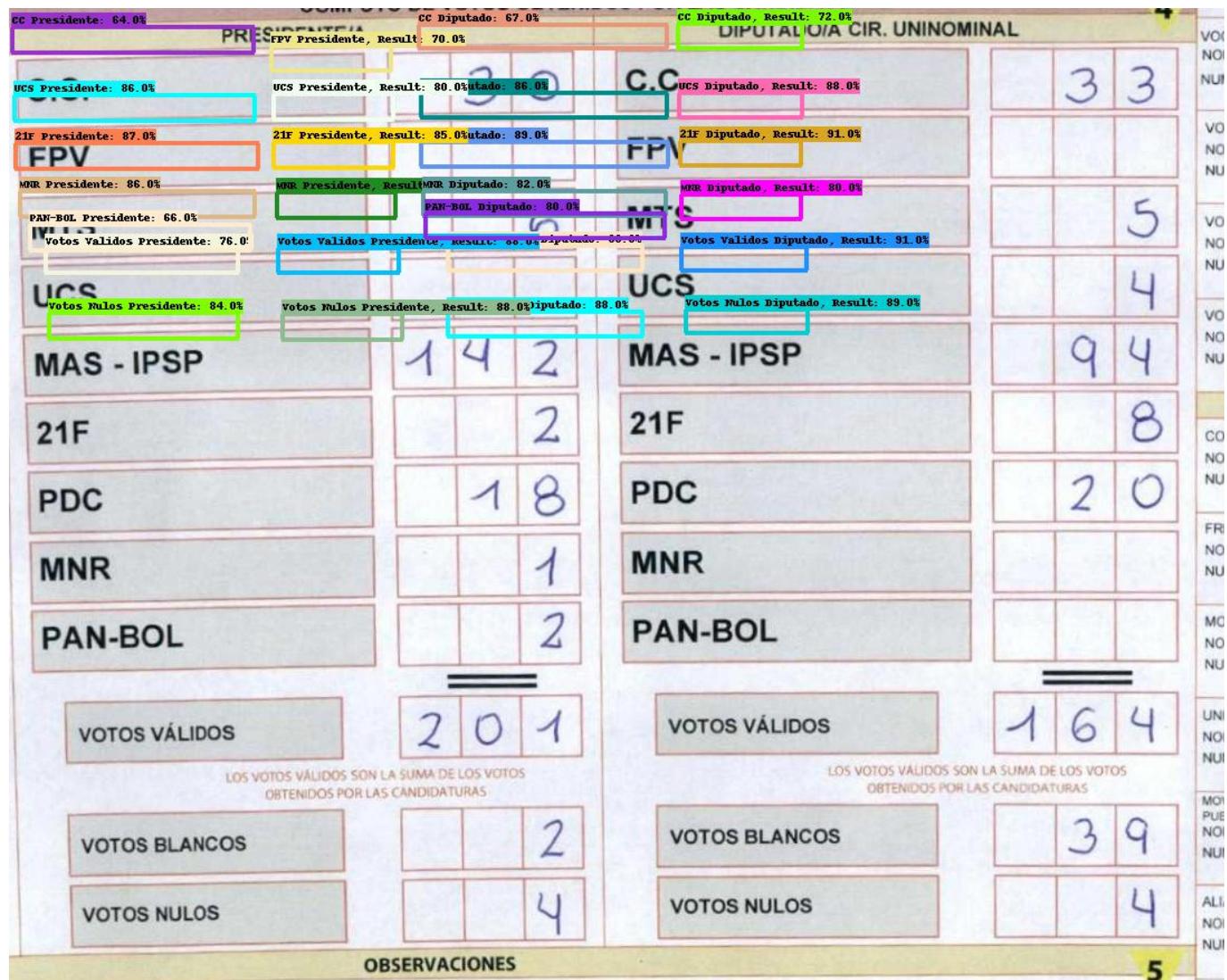
Attempt to finetune SSD MobilenetV2 with this VOC dataset.

### 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.

WORK IN PROGRES...