READING OF PLACES OF REGULATIONS WITH RASPBERRY PI 3+. ARTIFICIAL VISION

Salvador Marín i Montero

CIFO La Violeta, Barcelona, 20 de juny del 2019.

index

SUMMARY.

The reading of vehicle license plates is essential for the security sectors and car parking counts and check for tickets and departures. It is done with artificial vision so that sophisticated software intervenes in a controlled environment in order to have a uniform and reliable reading, with enough hardware to give an effective response in the least possible time.

PREFACE.

The initial idea was to read Chinese characters for, with OCR software, to convert a character to a Western language. This, unfortunately, has not been possible as a result of the first tests done on a PC with python language and bookstores to use, due to the strong demand for space, power of calculation, time and complexity that calls for this conversion.

Therefore they have had to reduce and transform the initial demands, reading a short text and defined in a controlled industrial environment at the same time as practical. The example is extracted from another project, reading medication boxes, to finally end up reading plates of vehicle license plates in Spain, as there are impressions of a web page.

READING FOR VEHICLE REGISTRATION.

So, finally, this reading has been chosen.

The license plates have been generated with a website designed for use.

https://www.matriculasdelmundo.com/creador.php

https://www.matriculasdelmundo.com/creador.php#CE

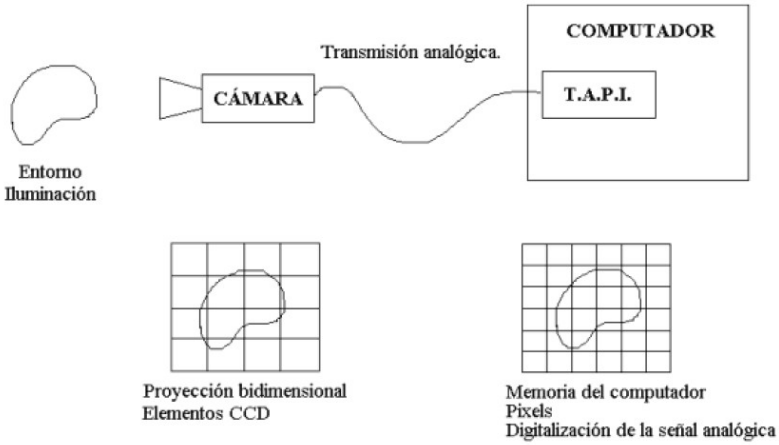
With it, a total of 10 images of enrollment in Spain have been generated, as shown in the image, which have been printed in A4 landscape while trimming the bottom of the text on the web.

The intention is to provide the camera with the same license plate image in the same position and with the same background and texture in order to minimize the input variables in the camera used in conjunction with the raspberry.

It is important to minimize the entry conditions so that, as described in the following section, the artificial vision demands readings as simple as possible.

ARTIFICIAL VISION.

In principle to read the license plates, an artificial vision assembly with OCR (optical characters reconaissance, optical character recognition) has been thought of. To achieve this, it is based on the following scheme shown in the image.



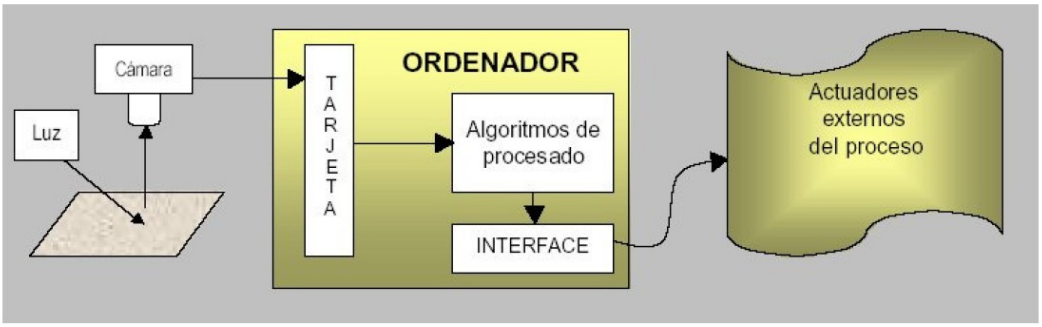


Figura 1. 17 Subsistemas físicos de un equipo de visión artificial

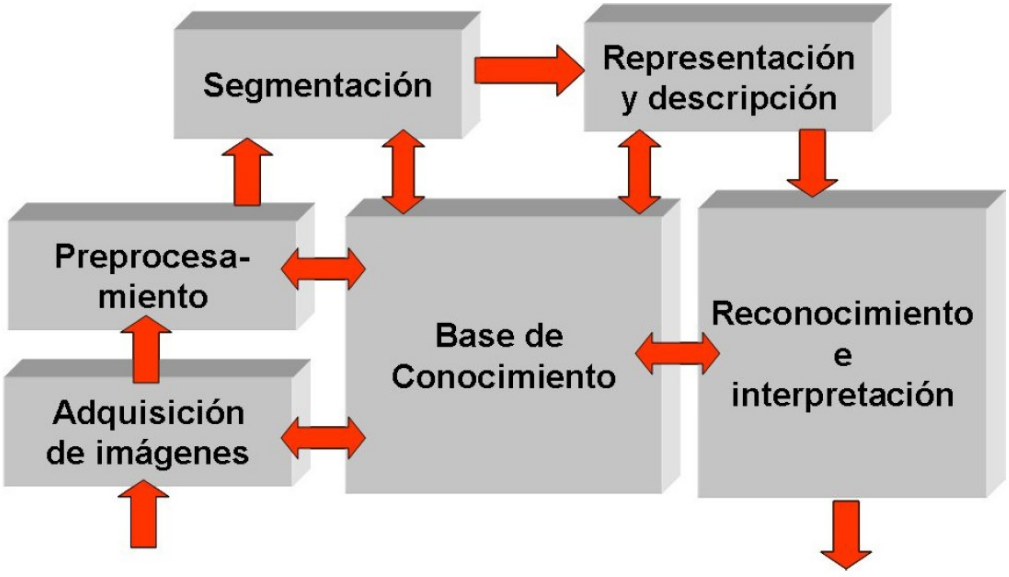


Figura 1. 19 Etapas de una aplicación de visión artificial

Below are the stages necessary to develop our artificial viscose system.

The first stage is the construction of the image formation system. Its objective is to enhance, through photographic techniques (lighting, optics, cameras, filters, screens ...), the visual characteristics of objects (shapes, textures, colors, shades, ...). The success of many applications depends on a good design in this first stage.

Once the image is acquired it will be passed to the preprocessed stage. The objective is to improve the informative quality of the acquired image. Signaling-noise (DNSS), SNR, enhancement operations are included to mitigate the acquisition imperfections due to the transfer function of the image capture system (deconvolution), to regularize the image, to improve the contrast or optimize the intensity distribution (enhancement) or enhance some features of the image, such as edges or areas.

Segmentation is the phase where the image is partitioned in areas with meaning. For example, a satellite image determines the areas of water, culture, urban, roads, ... There are several techniques: thresholds, discontinuities, growth of regions, use of color or movement, etc. These strategies must be analyzed. Once the image is split into zones with higher-level features, it will be passed to its extraction of the features. They are basically morphological types, such as area, perimeter, eccentricity, moments of inertia, skeletons, but also characteristics based on texture or color can be used.

***

An Artificial Vision project is usually of a multidisciplinary nature. The execution of the different stages, mentioned in the previous section, requires the handling of the following techniques:

1- Photography and Optics: Creating the right lighting environment in the acquisition of images, often requires the use of professional photography and video techniques. The selection of the optics and the camera, the use of filters and polarizers, the illumination techniques with

Screens and the choice of focus types are some skills that can be mentioned.

2- Digital Processing of Images (Image Processing): It refers to computer algorithms that make the digital image acquired in another of more relevance. The separation between image processing and Artificial Vision is very diffuse.

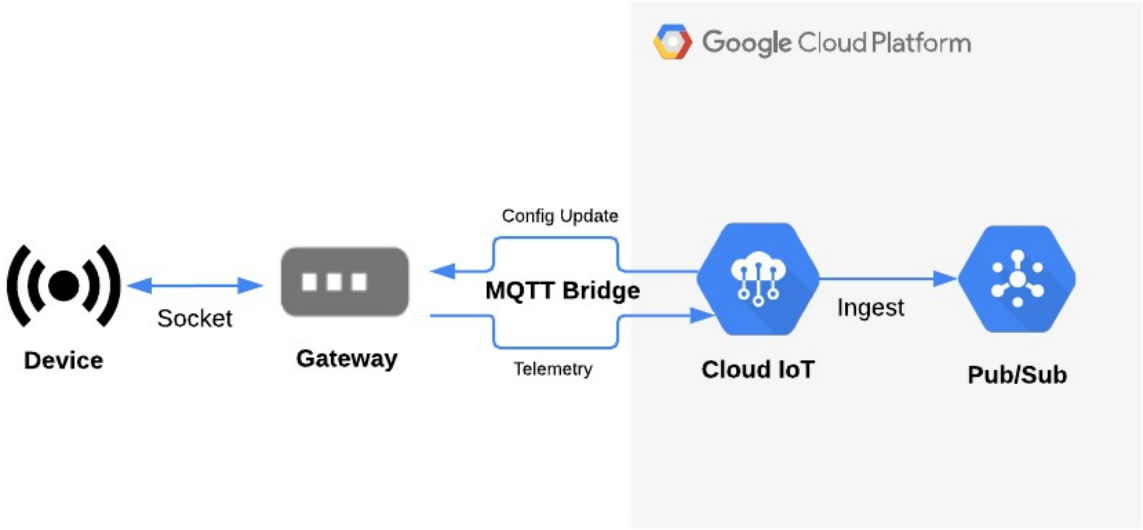
3- Pattern Recognition (Pattern Recognition): discipline, within Artificial Intelligence, dedicated to the classification of signals and the search for patterns existing within them. They include the techniques of statistical classifiers, Neural Networks, Expert Systems, Blurred Logic, ...

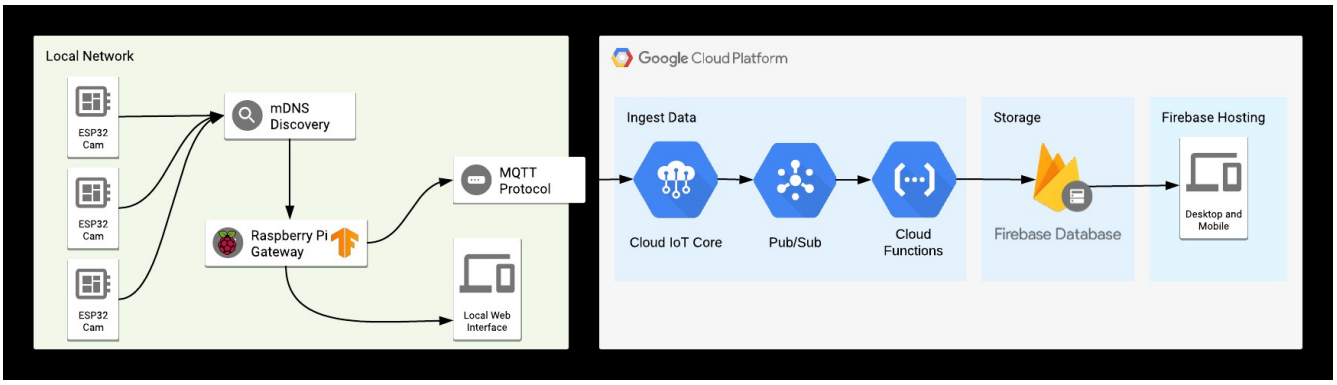
4- Computer Graphics: Present the reverse problem of Artificial Vision. If Vision wishes to extract the physical characteristics of the images, the Graphics Computation is dedicated to the visual presentation of the geometric models. Increasingly, the Artificial Vision uses

Graphic Computing to represent the conclusions extracted from the analysis of the acquired images.

GOAL.

To demonstrate that, with a Raspberry PI 3+, you can create an artificial vision system that, with a scheme of Internet of Things, could raise a series of 10 readings of vehicle license plates in file format CSV to a cloud formed in Google Drive. An example could be that shown in the images.





HARDWARE USED.

The following hardware elements are available to achieve this practice:

A Raspberry PI 3+ Model B Starter Pack with 16Gb SDK card.

A NIKON PIXMA COOLPIX P100 camera.

SOFTWARE USED.

On your computer or PC:

- Windows 10.
- Putty. To connect remotely to Raspberry.
- WinSCP. With the same purpose to transfer file between PC and Raspberry.
- Linux.
- Python version 2 and version 3 installed.

To the Raspberry PI 3+:

- Operating system Raspbian Lite Strecht.
- Python version 2 and version 3.
- Installation of the following packages and programs under Python:

Pydrive. Bookstore to upload files from Raspberry to Google Drive.

Numpy Python mathematical bookstore

Pytesseract. Python ORC character recognition library.

Gphoto2. Software in python for capturing images with reflex cameras.

OpenCV2. Sotware written in python for image processing.

in addition to those described on python imports made at the beginning of the programs.

INSTALLATION OF SOFTWARE.

It is noteworthy that the Raspbian Linux Lite Strecth operating system has been installed in the Raspberry Pi3 + operating modus to reduce the internal processing and memory capacity of Raspberry.

https://www.raspberrypi.org/downloads/raspbian/

We begin by defining in a folder of work in the Raspberry all the files, libraries and executables in python with which it must work.

It will be cd / home / pi.

From the Putty terminal, it starts to update the following.

sudo apt-get install pip

sudo apt-get update

sudo apt-get upgrade

The intention is to update the Python language manager in order to install the programs indicated above, which are those that require capturing, preprocessing and final reading of the characters printed on the license plate.

Once done, it is proceeded to install an OCR character reader that could work on python and that it is compact enough to be installed in the Raspberry SDK memory. Choose the so-called Pytesseract, which is well-known and works, according to previous tests that were carried out on the PC of the CIFO La Violeta course, quickly enough.

sudo pip install pytesseract

https://pypi.org/project/PyTesser/

Next, the Gphoto2 software, which allows, with a camera described above, a USB connection to one of the Raspberry ports, take photos, save them to the SDK I memory, and then treat them. It is a very bulky software and we will have to take the steps according to a page that describes the entire installation process.

http://www.jpereira.net/software-revisiones-y-consejos/gphoto2-control-remoto-de-camara-open-source

https://pimylifeup.com/raspberry-pi-dslr-camera-control/

sudo pip install gphoto2.

https://pypi.org/project/gphoto2/

Finally, it will be necessary to test the operation and check with the camera mentioned above.

Later, it will be necessary to put the image processing software OpenCV -4, which allows us to take the color images of the digital camera, make them black and white, remove the noise and the artifacts or small spots, and finally leave Full image to make character recognition.

https://hyfrmn.wordpress.com/2015/02/03/install-libgphoto2-and-gphoto2-from-source-on-raspberry-pi/

sudo pip install python-opencv

https://www.alatortsev.com/2018/09/05/installing-opencv-3-4-3-on-raspberry-pi-3-b/

https://www.alatortsev.com/2018/11/21/installing-opencv-4-0-on-raspberry-pi-3-b/

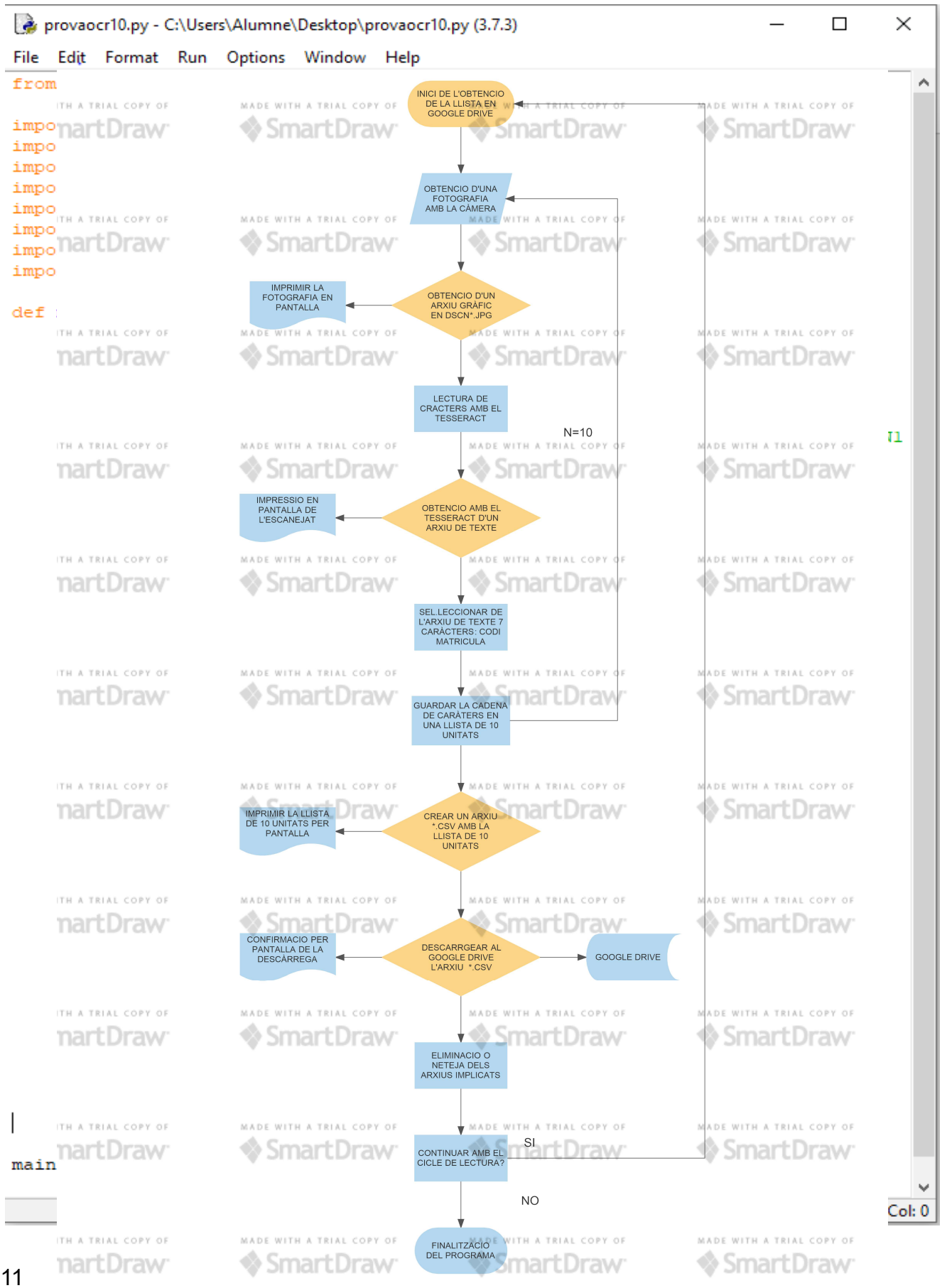
https://www.learnopencv.com/install-opencv-3-4-4-on-raspberry-pi/

https://circuitdigest.com/microcontroller-projects/how-to-install-python-opencv-on-raspberry-pi

We also installed the numpy library in order to have mathematical libraries to handle images and discriminate regions of the image that were recognized by OCR.

https://www.numpy.org/

https://pypi.org/project/numpy/


provaocr10.py - C:\Users\Alumne\Desktop\provaocr10.py (3.7.3)
File Edit Format Run Options Window Help
from
impo
impo
impo
impo
impo
impo
impo
impo
def
main
Col: 0
INICI DE L'OBTENCIO DE LA LLISTA EN GOOGLE DRIVE
OBTENCIO D'UNA FOTOGRAFIA AMB LA CÀMERA
OBTENCIO D'UN ARXIU GRÀFIC EN DSCN*.JPG
IMPRIMIR LA FOTOGRAFIA EN PANTALLA
LECTURA DE CRACTERS AMB EL TESSERACT
N=10
OBTENCIO AMB EL TESSERACT D'UN ARXIU DE TEXTE
IMPRESSIO EN PANTALLA DE L'ESCANEJAT
SEL.LECCIONAR DE L'ARXIU DE TEXTE 7 CARÀCTERS: CODI MATRICULA
GUARDAR LA CADENA DE CARÀTERS EN UNA LLISTA DE 10 UNITATS
CREAR UN ARXIU *.CSV AMB LA LLISTA DE 10 UNITATS
IMPRIMIR LA LLISTA DE 10 UNITATS PER PANTALLA
DESCARRGEAR AL GOOGLE DRIVE L'ARXIU *.CSV
GOOGLE DRIVE
CONFIRMACIO PER PANTALLA DE LA DESCÀRREGA
ELIMINACIO O NETEJA DELS ARXIUS IMPLICATS
CONTINUAR AMB EL CICLE DE LECTURA?
SI
NO
FINALITZACIO DEL PROGRAMA

Next, the two programs involved in the software are placed. The codes are placed so that they can be reproduced a posteriori. A flowchart is also presented that describes the work modes and operation of the programs involved.

provaocr11-bis.py - C:\Users\Alumne\Desktop\PROJECTEOCR\provaocr11-bis.py (3.7.3)

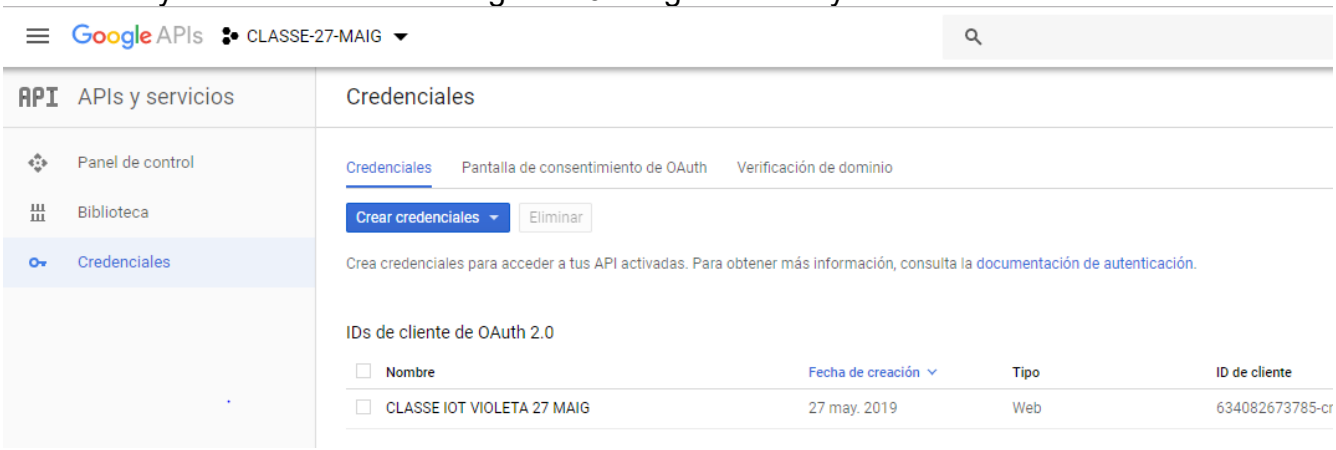
File Edit Format Run Options Window Help

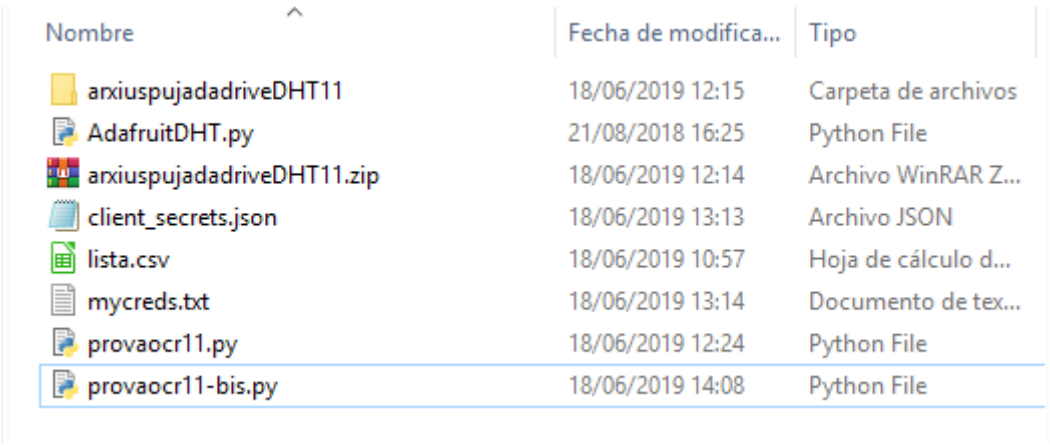
```
from PIL import Image

from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
import sys
import time
import datetime
import pydrive

import numpy as np
import pytesseract
import gphoto2
import os
import csv
import imutils
import cv2

data='lista.csv'
g_login = GoogleAuth()
g_login.LoadCredentialsFile("mycreds.txt")
if g_login.credentials is None:
    # Authenticate if they're not there
    g_login.LocalWebserverAuth()
elif g_login.access_token_expired:
    # Refresh them if expired
    g_login.Refresh()
else:
    # Initialize the saved creds
    g_login.Authorize()
# Save the current credentials to a file
g_login.SaveCredentialsFile("mycreds.txt")
drive = GoogleDrive(g_login)
file_drive = drive.CreateFile({'title': data})
file_drive.SetContentFile(data)
file_drive.Upload()
print("L'arxiu ha pujat al drive")
sys.exit()
```

Ln: 1 Col: 0

These two files need to be first executed on the PC, in order to request, from a Google API, a credential file.

You must first have a Google API enabled as shown in the attached image. Once the credential file is requested, it is executed on the PC, generating a file called mycreds.txt, which will copy the information contained in the files client_secrets.json.
Then the mycreds.txt file can be retrieved, and the two programs are python to raspberry and from there you can work on reading the 10 images taken by the camera.





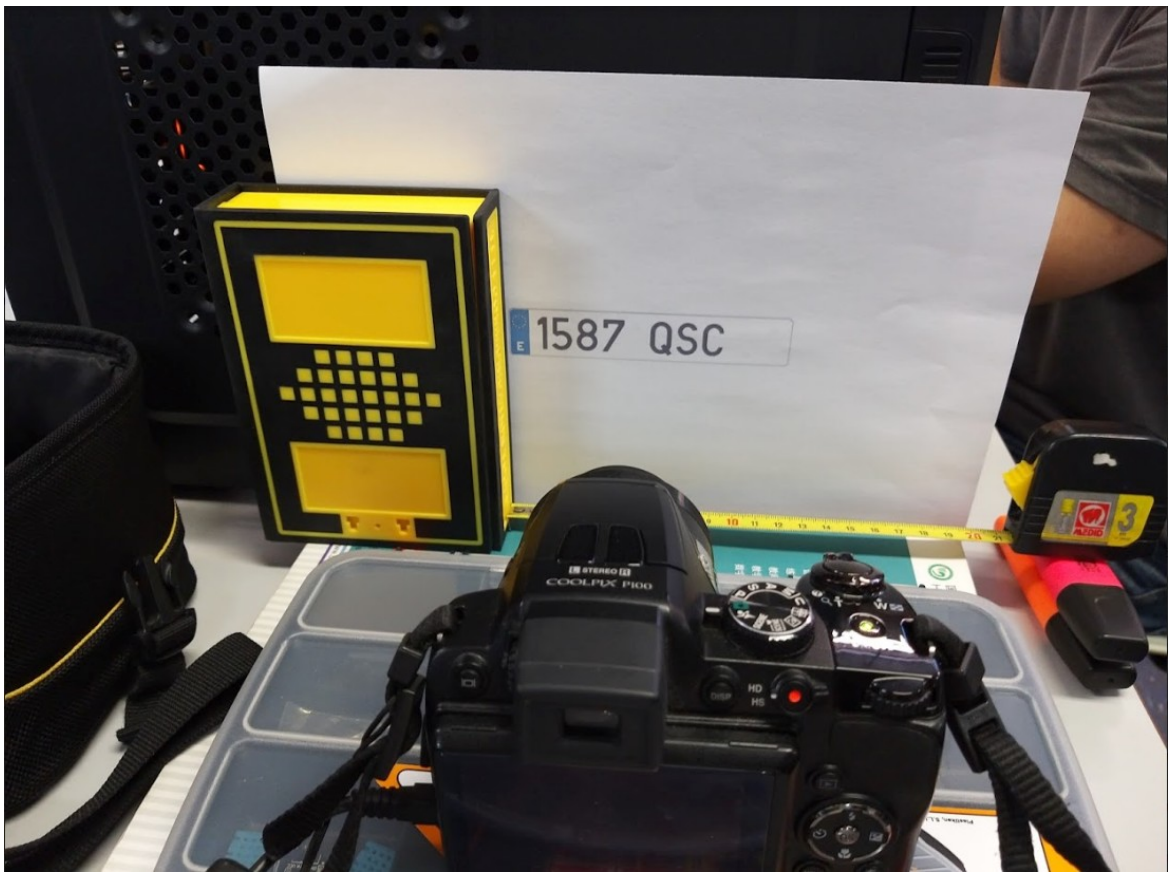
READING ASSEMBLY

To read the sheets with printed license plates, it will be necessary to implement a camera assembly that is suitable for taking photographs that, with raspberry, we can preprocess and try to do the OCR.

Below are a series of images where modification is carried out in class. It should be taken into account that it is based on very precarious means but not less effective.

Finally, a screenshot is made showing the reading of the photograph with the read license.

And, last but not least, as an example, an example of a list file. Csv where you can see the charts of the license plates read.


1587 QSC
COOLPIX P100


1587 QSC
Photobag

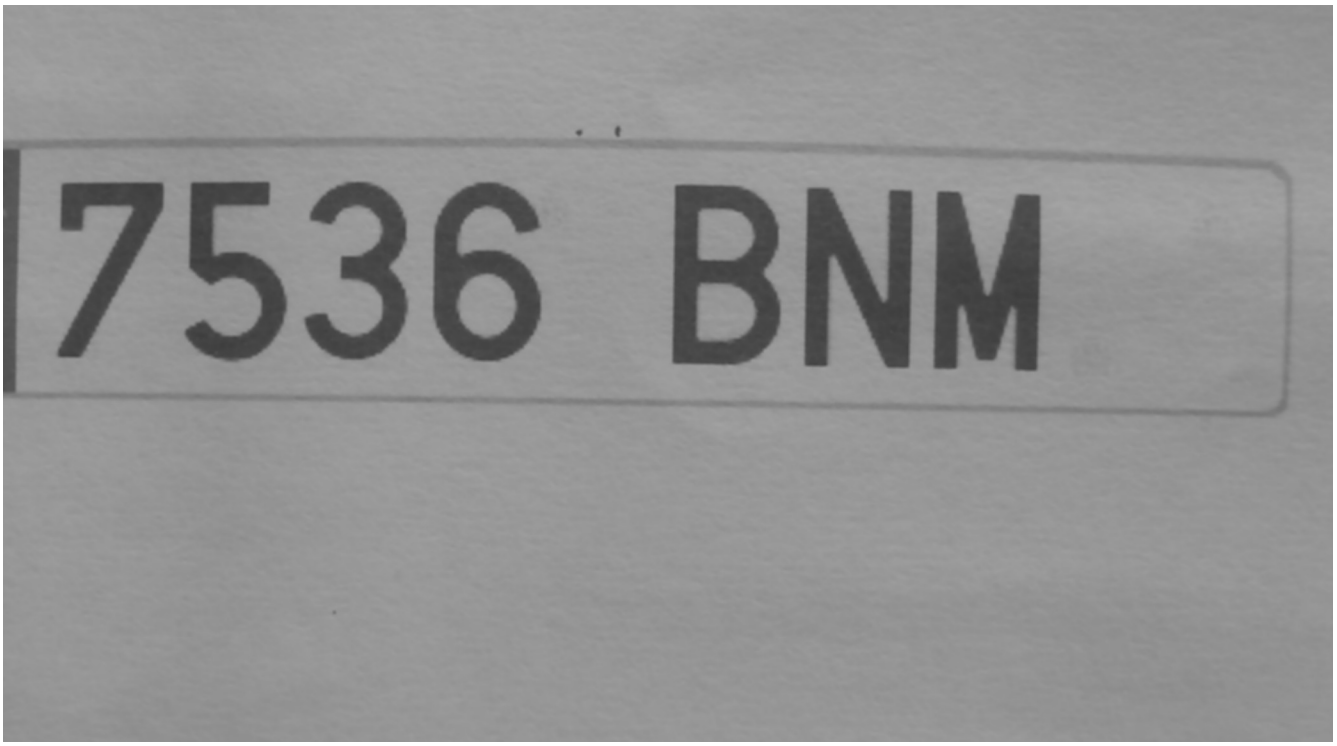

TELECOMMUNICATION TOOLS
1587 QSC
26x
10x
CMOS
10M
Full HD
Movie


```
pi@raspberrypi: ~
[u'1587 QSC']
root@raspberrypi:/home/pi# python provaocr10.py
El nuevo fichero está en la ubicación /store_00010001/DCIM/100NIKON/DSCN0088.JPG en la cá
El fichero DSCN1.JPG existe. ¿Sobreescribir? [y|n] y
Guardando el fichero como DSCN1.JPG
Deleting file /store_00010001/DCIM/100NIKON/DSCN0088.JPG on the camera

*** Error ***
Manejador PTP de objeto no válido
('Detected Number is:', u'')

[u'']
root@raspberrypi:/home/pi# python provaocr10.py
El nuevo fichero está en la ubicación /store_00010001/DCIM/100NIKON/DSCN0089.JPG en la cám
El fichero DSCN1.JPG existe. ¿Sobreescribir? [y|n] y
Guardando el fichero como DSCN1.JPG
Deleting file /store_00010001/DCIM/100NIKON/DSCN0089.JPG on the camera

*** Error ***
Manejador PTP de objeto no válido
('Detected Number is:', u'1587 QSC')
1587 QSC
[u'1587 QSC']
root@raspberrypi:/home/pi#
```

lista.csv

	A	B	C	D	E	F	G	H
1	Matricula							
2								
3	5	3	6			B	N	M
4	5	1	6		A	E	F	
5	3	6	9		C	V	B	
6								
7								
8	2	8	4		Z	D	T	

DEVELOPMENT OF READING.



This would be the reading of the resulting plate of the NIKON camera already trimmed, grayscale and all the noise and artifacts (or small stains) present in the image. Compare with previous images how the A4 landscape format was conveniently printed.

DISCUSSION OF THE MOUNTING AND DEVELOPMENT GENERATED TO PRACTICE.

The following phenomena must be taken into account to implement this practice. These phenomena are the result of about 40 hours of practice with OCR and python programming, assembly, realization and subsequent analysis and review of this practice.

- The camera must be with full battery to operate, it should not be fed with the 5V USB of the Raspberry, so that it does not remove the operating force from the latter.

- The SDK card of the camera must be properly formatted.

- The photographs are loaded in the directory where the python code and its software are executed, for our case cd / home / pi

- The photographs must have a white background and not a color, but the OCR tesseract does not work well, in fact it does not read.

- Characters captures basically depend on the surrounding lighting, so it is recommended to do a few tests before making the series of 10 photographs.

- The readings are sometimes incomplete, some characters may be left in some cases.

- Handwriting can not be recognized.

- A unique reading format is necessary in order to simplify the reading and that they are all uniform with a single height and format of letter.

- The foacl distance is very important. It is determined in this assembly in 130mm (see photographs).

- The lighting is also great. It is necessary to avoid the illumination with fluorescentes and to go to a diffused light, being the best indirect sunlight.

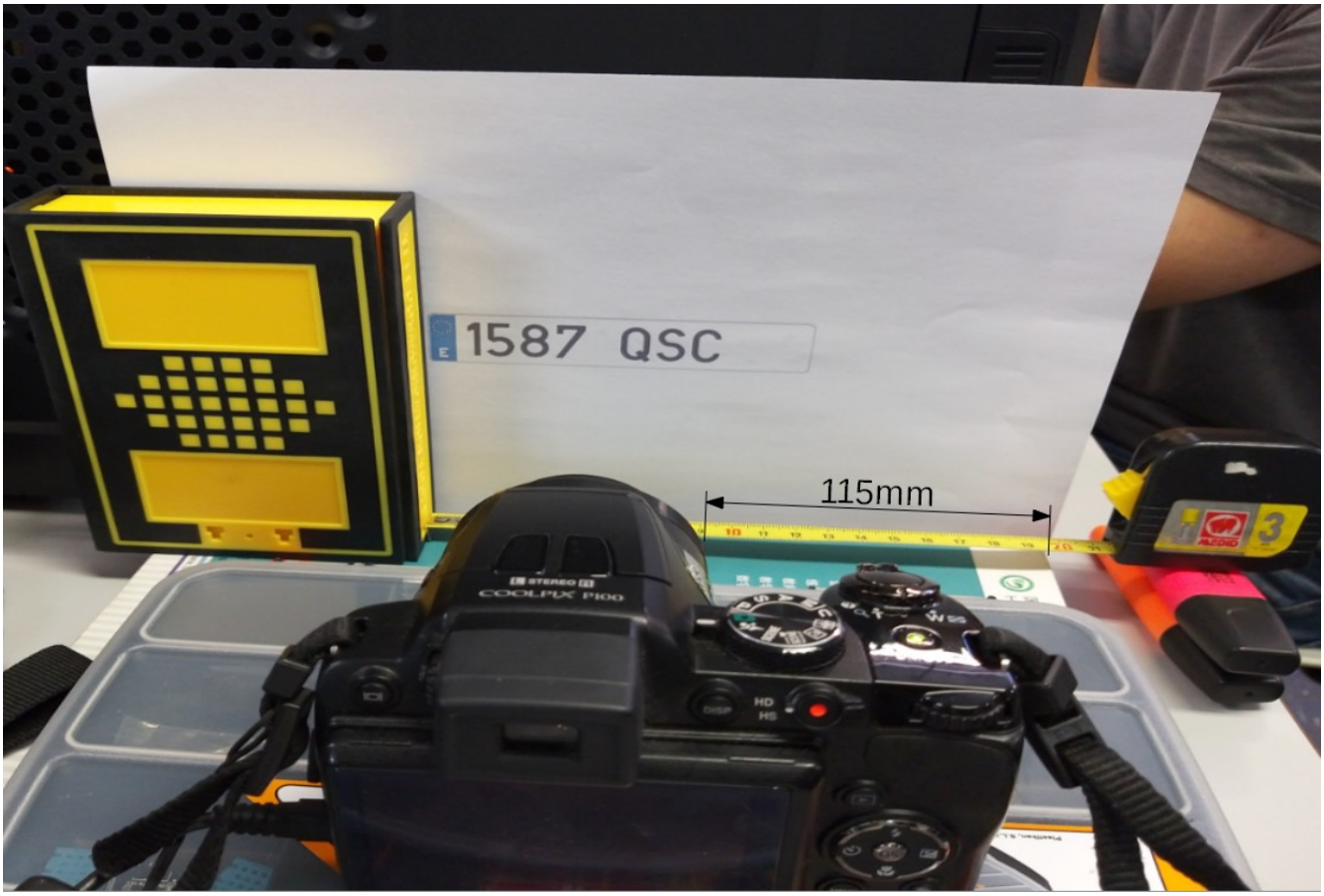
- The position of the blade is also very important, so that any blinking, bending on the blade or inclining or disengaging may cause the failure of the reading. It must be totally vertical.

- The character string should preferably be placed in the center of the camera's reading.

- The image is reduced according to the following operations carried out with the Open CV software:
- Gray scale change of the original color image.
- No noise in the image, that is, without points or lines that may confuse the OCR.
- A cut of the fund is applied to work only with the frame with the characters eliminating the surroundings.

It shows below the situation levels of the camera with respect to the sheet to read. We must strictly respect them to report the same conditions.


130 mm
60mm
TELECOMMUNICATION TOOLS
1587 QSC



- These levels are very important. Normally, in artificial vision assemblies, both lighting and focal length as the camera situation are determined according to many tests, with the test and error method, until finally the experience achieved establishes a position as a whole that makes the readings finally correct.

- It should also be noted that tests carried out with reflex cameras on the outside, based on photographs taken, should be able to handle OpenCV commands to eliminate noise in images of pure black and white or binary. Apparently, it also gives positive results.

ANNEXE: OTHER METHOD WITH REGULAR EXPRESSIONS.

Normally, there are always more ways to achieve the requested result in programming. With this intention, there is an alternative path with software OPEN CV with the canny parameter that generates a lot of noise in the image. This noise or artefacts, as sometimes also called, must be cleaned up in order to achieve an optimal reading of characters.

The resulting code in python is reported continuously.

nouenfoc.py - C:\Users\Alumne\Desktop\Visio artificial nou enfoc\nouenfoc.py (3.7.3)

File Edit Format Run Options Window Help

```
from PIL import Image
import cv2
import numpy as np
from matplotlib import pyplot as plt
from pytesseract import image_to_string
import re
#convertir imatge en escala de grisos havia de ser png

img = Image.open('indice-2.png').convert('LA')
img.save('greyscale.png')

# edge detection jugant amb canny i matplotlib
img = cv2.imread('greyscale.png',0)
edges = cv2.Canny(img,100,200)
plt.subplot(121),plt.imshow(img,cmap = 'gray')
plt.subplot(122),plt.imshow(edges,cmap = 'gray')
plt.savefig('sortida.png')


#OCR AMB PYTESSERACT

text=image_to_string('sortida.png',config='--psm 11')

# PATRO DE REGULAR EXPRESSIONS PER FILTRAR EL "SOROLL" LLEGIT
# dos numeros o lletres un espai tres lletres o numeros un espai i dos numeros o lletres
#regex = re.compile("[0-9A-Z]{2}\s[0-9A-Z]{3}\s[0-9A-Z]{2}")
#quatre numeros un espai i tres lletres
#regex = re.compile("[0-9]{4}\s[A-Z]{3}")
#filtrat dos lletres un espai i cinc numeros
regex = re.compile("[A-Z]{2}\s[0-9]{5}")
print(text)
print('FILTRAT REGEX')
print(regex.findall(text))
```

It is shown in the code of Canny methods and regular expressions that read a license plate is required to read below.

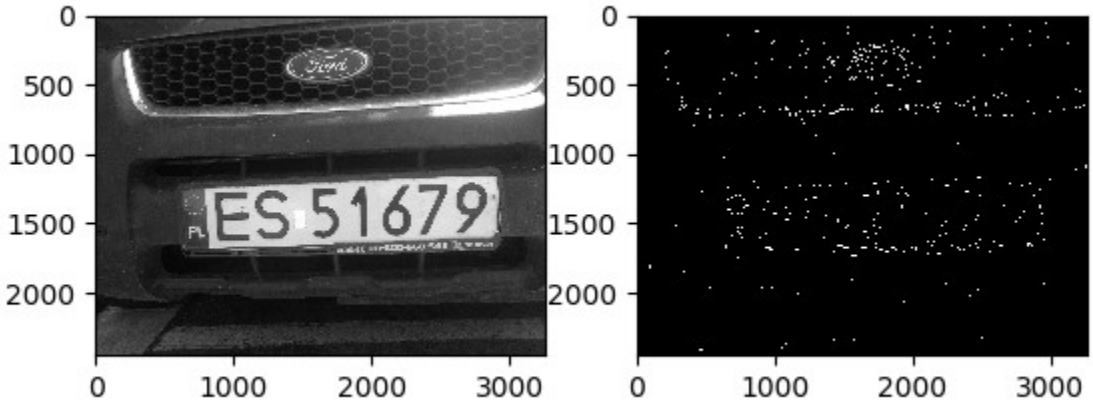

Ford
PL
ES 51679
leśna 16 800-660-565
SERWIS

Python 3.7.3 Shell

File Edit Shell Debug Options Window Help

```
>>>
== RESTART: C:\Users\Alumne\Desktop\Visio artificial nou enfoc\nouenfoc.py ==
500

500

__

1000

ooo

1500

500

BES 51679]

2000

2000

0

1000

2000

3000

0

1000

2000

3000
FILTRAT REGEX
['ES 51679']
>>>
```

Ln: 42 Col: 4



BIBLIOGRAPHY AND LINKS REFERENCE WEBSITE.
Work with lines of text and files.

https://www.google.cat/search?ei=EEEBXdC1DoaalwTwp6-IDQ&q=buscar+texto+en+archivo+python&oq=python+texto+en+achvo&gs_l=psy-ab.1.0.0i8i13i30.80686.114238..116796...4.0..0.156.3285.35j3.....0....1..gws-wiz.....0i71j0j0i22i30j0i131j0i19j0i10i19j0i22i10i30i19j0i22i30i19j0i22i10i30j33i22i29i30.i86hHmjOhR4

http://chocolatesexyconsulting.es/index.php/2016/05/23/python-parte-3-tipos-de-datos/

https://docs.python.org/3/library/os.html

https://mail.python.org/pipermail/python-es/2007-July/017723.html

https://uniwebsidad.com/libros/python/capitulo-10/modulos-de-sistema

https://programminghistorian.org/es/lecciones/manipular-cadenas-de-caracteres-en-python#encontrar

http://chocolatesexyconsulting.es/index.php/2017/05/23/busqueda-de-cadenas-de-texto-con-python/

http://chocolatesexyconsulting.es/index.php/2017/05/23/leer-ficheros-python-linea-linea/

https://www.google.cat/search?ei=1kMBXdTqDZKZlwShqqn4Cw&q=python+imprimir+en+pantalla+archivo&oq=python+imprimir+en+pantalla+archiv&gs_l=psy-ab.1.0.33i22i29i30.2655.8678..10646...1.0..0.131.799.6j2.....0....1..gws-wiz.....0i71j33i160j33i21.Ndp4fQ_ls5Y

https://programminghistorian.org/es/lecciones/trabajar-con-archivos-de-texto

---

Work with csv files.

https://www.google.cat/search?ei=ks4BXZvXMIy7gwfXya2IAg&q=como+crear+un+archivo+csv+en+python&oq=como+crear+un+archivo+csv+en&gs_l=psy-ab.1.0.0l6j0i22i30.4172.4645..6417...0.0..0.94.252.3.....0....1..gws-wiz.....0i71j0i67.Xb3zHcBgM68

https://www.lawebdelprogramador.com/foros/Python/1609359-Crear-un-archivo-csv-a-partir-de-una-lista.html

https://code.tutsplus.com/es/tutorials/how-to-read-and-write-csv-files-in-python--cms-29907

https://es.stackoverflow.com/questions/11698/como-generar-un-archivo-csv-a-partir-de-una-lista

https://python-para-impacientes.blogspot.com/2015/05/operaciones-con-archivos-csv.html

https://www.google.cat/search?source=hp&ei=pZIIXdjcNIyeUIbMhEA&q=writecol+python&oq=writecol&gs_l=psy-ab.1.0.0i19l2j0i10i19l3j0i10i30i19j0i30i19j0i5i10i30i19.1266.4595..7552...0.0..0.101.653.7j1.....0....1..gws-wiz.....0..0i131j0j0i10.ZM6wUuxApYQ

https://stackoverflow.com/questions/17704244/writing-python-lists-to-columns-in-csv

https://stackoverflow.com/questions/4155106/python-csv-write-by-column-rather-than-row

//

Work with raspberry pi and opencv.

https://www.google.cat/search?q=tesseract+ocr+car+plate+numbers&ei=_hgCXaqzA8GwaaqLiPgM&start=10&sa=N&ved=0ahUKEwiqquqOlObiAhVBWBoKHaoFAs8Q8NMDCMgB&biw=1920&bih=937

https://circuitdigest.com/microcontroller-projects/license-plate-recognition-using-raspberry-pi-and-opencv

https://www.google.cat/search?source=hp&ei=GiMCXYT5H4iMlwSlgoPgDg&q=gray+%3D+cv2.cvtcolor%28image+cv2.color_bgr2gray%29+error&oq=gray+%3D+cv2.cvtColor%28img%2C+cv2.COLOR_BGR2GRAY%29&gs_l=psy-ab.1.2.0j0i22i30l2j0i22i10i30l3.13829.13829..39349...0.0..0.236.236.2-1.....0....2j1..gws-wiz.....0.hMIjrRnkvFc

https://www.google.cat/search?source=hp&ei=9ygCXfOCDtiLjLsP_oqsoAs&q=cv2+help&oq=cv2+help&gs_l=psy-ab.3..0i19j0i22i30i19l8.1228.6500..9379...0.0..0.83.572.8.....0....1..gws-wiz.....0..0i131j0j0i22i30.eOXXc8PeWw0

https://docs.opencv.org/3.0-beta/modules/text/doc/ocr.html?highlight=characters

https://www.google.cat/search?q=(topy,+topy)+%3D+(np.min(x),+np.min(y))&lr=lang_es&sa=X&ved=0ahUKEwiA2pnttubiAhXJAWMBHdeICLgQuAEIKw&biw=1920&bih=937

https://codeday.me/es/qa/20190407/440030.html

https://rstopup.com/como-recortar-el-area-interna-de-un-contorno-2.html

https://cvisiondemy.com/license-plate-detection-with-opencv-and-python/

https://github.com/Link009/LPEX

http://cvisiondemy.com/character-recognition-alpr-ocr-case/

Treball amb tractament de opencv.

https://www.google.cat/search?rlz=1C2CHBF_caES840ES844&source=hp&ei=Lx4DXYPqJKO5gwf7kIDwCQ&q=_%2C+contours%2C+_+%3D+cv2.findContours%28...%29&oq=_%2C+contours%2C+_+%3D+cv2.findContours%28...%29&gs_l=psy-ab.12..0i22i30l4.13298.13298..23132...0.0..0.78.78.1.....0....2j1..gws-wiz.....0.o6LhCwYN56M

https://docs.opencv.org/3.1.0/d3/dc0/group__imgproc__shape.html#ga17ed9f5d79ae97bd4c7cf18403e1689a

https://www.pyimagesearch.com/2014/04/21/building-pokedex-python-finding-game-boy-screen-step-4-6/

https://foro.hacklabalmeria.net/t/consulta-codigo-tratamiento-de-imagen-en-python/9247/3

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html

https://www.geeksforgeeks.org/find-and-draw-contours-using-opencv-python/

https://www.google.cat/search?q=_,+contours,+_+%3D+cv2.findContours(...)&rlz=1C2CHBF_caES840ES844&lr=lang_es&sa=X&ved=0ahUKEwjUutCmjejiAhUFBGMBHaRXCGUQuAEIKw&biw=1920&bih=937

https://es.stackoverflow.com/questions/153275/c%C3%B3mo-usar-contours-apropiadamente

http://acodigo.blogspot.com/2017/08/deteccion-de-contornos-con-opencv-python.html

https://codeday.me/es/qa/20190512/687606.html

---

https://www.alatortsev.com/2018/09/05/installing-opencv-3-4-3-on-raspberry-pi-3-b/

https://www.alatortsev.com/2018/11/21/installing-opencv-4-0-on-raspberry-pi-3-b/

https://www.learnopencv.com/install-opencv-3-4-4-on-raspberry-pi/

---

https://www.google.cat/search?source=hp&ei=g7_4XJvlIfGKlwTa64ho&q=gphoto2&oq=gph&gs_l=psy-ab.1.2.0i10j0l9.3457.6395..10232...0.0..0.82.210.3.....0....1..gws-wiz.....0..0i131.wh-X6JxUt88

Treball de python amb gphoto.

http://www.gphoto.org/

https://www.youtube.com/watch?v=1eAYxnSU2aw

https://www.youtube.com/watch?v=nQKe3rc4j6M

http://www.jpereira.net/software-revisiones-y-consejos/gphoto2-control-remoto-de-camara-open-source

https://github.com/gphoto/gphoto2

https://en.wikipedia.org/wiki/GPhoto

https://pypi.org/project/gphoto2/

---

https://www.youtube.com/watch?v=X6HkcJp2tSI

https://www.youtube.com/watch?v=_L2hxlWpIJY

---

https://www.google.cat/search?ei=mcD4XPGMMoeRlwTzpKjIBA&q=gphoto2+raspberry+pi+3+nikon&oq=gphoto2+raspberry+pi+3+nikon&gs_l=psy-ab.3...42163.47113..49831...0.0..0.1542.4536.0j1j1j1j7-2j1.....0....1..gws-wiz.....0i22i30j33i22i29i30j33i21j33i160.NeKYIDhQ8bM

http://thezanshow.com/electronics-tutorials/raspberry-pi/tutorial-41

https://www.google.cat/search?biw=1920&bih=937&q=gphoto2+tutorial&sa=X&ved=2ahUKEwjezabvydTiAhV0qnEKHdLMBNgQ1QIoBXoECAoQBg

http://www.gphoto.org/doc/manual/

https://www.youtube.com/watch?v=62cM2Ni2UlM

http://vamox.blogspot.com/2015/02/tutorial-generar-un-timelapse-con-gphoto.html

---

Raspberry Pi work with NIKON camera and gphoto.

https://www.google.cat/search?ei=hOf4XPqBE-aF1fAP4oGUqA0&q=raspberry+pi+3+b+gphoto2+how+to+install&oq=raspberry+pi+3+b+gphoto2+how+to+install&gs_l=psy-ab.3...11467.17974..18556...0.0..0.162.1520.11j4.....0....1..gws-wiz.....0i71j33i160j33i21j33i22i29i30.GmsfyVKIb3o

https://hyfrmn.wordpress.com/2015/02/03/install-libgphoto2-and-gphoto2-from-source-on-raspberry-pi/

https://medium.com/@cgulabrani/controlling-your-dslr-through-raspberry-pi-ad4896f5e225

https://pimylifeup.com/raspberry-pi-dslr-camera-control/

Work with python with tesseract.

https://www.google.cat/search?rlz=1C2ASUT_enES838ES847&ei=9er5XM62NYKKaqbak5gI&q=tesseract+raspberry+pi+3&oq=tesseract+ras&gs_l=psy-ab.1.2.0i19l3j0i22i30i19l7.5608.8228..11271...0.0..0.96.350.4.....0....1..gws-wiz.....0i71j0i67j0j0i22i30.37hadx3anK0

https://github.com/thortex/rpi3-tesseract

https://circuitdigest.com/microcontroller-projects/optical-character-recognition-ocr-using-tesseract-on-raspberry-pi

https://www.google.cat/search?rlz=1C2ASUT_enES838ES847&source=hp&ei=I7r7XNqIF5KcaYD9l9gO&q=how+to+use+tesseract+in+raspberry+pi+3&oq=how+to+use+tesseract+in+ras&gs_l=psy-ab.1.1.33i22i29i30l2.5697.16265..18756...0.0..0.193.2607.21j6.....0....1..gws-wiz.....0..0i131j0j0i22i30j0i22i10i30j0i19j0i13i30i19j0i22i30i19.QN-fSJlKAwg

https://circuitdigest.com/microcontroller-projects/how-to-install-python-opencv-on-raspberry-pi

https://www.raspberrypi.org/forums/viewtopic.php?t=45645

https://raspberrypi.stackexchange.com/questions/89231/tesseract-ocr-4-x-beta-for-raspberry-pi

https://raspberrypi.stackexchange.com/questions/42622/rpi-ocr-or-how-to-read-a-number-from-the-camera

****

https://www.google.cat/search?rlz=1C2ASUT_enES838ES847&ei=gd_7XK_-EMLngwfJnbrgDQ&q=how+to+read+chinese+characters+with+a+raspberry+pi+3+teressact&oq=how

+to+read+chinese+characters+with+a+raspberry+pi+3+teressact&gs_l=psy-ab.3...6715.12165..13786...1.0..0.195.1101.8j3.....0....1..gws-wiz.....33i160j33i21.2z-SjlD44yM

https://askubuntu.com/questions/793634/how-do-i-install-a-new-language-pack-for-tesseract-on-16-04

//

Miscellaneous.

https://www.google.cat/search?ei=AYT_XM-ENbXmgwel-Zi4Cg&q=tesseract+best+format+image&oq=tesseract+best+format+image&gs_l=psy-ab.3..0i71l8.3119.4278..4780...0.0..0.0.0.....0....1..gws-wiz.0He2v5Ri8o0

https://github.com/tesseract-ocr/tesseract/wiki/ImproveQuality#still-having-problems

http://www.gphoto.org/doc/manual/ref-gphoto2-cli.html#cli-examples

---

Enrollment job to print.

http://www.simec.es/sistemas_identificacion.aspx?where=504

https://www.matriculasdelmundo.com/creador.php

http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap1IntroVA.pdf