

LECTURA DE PLAQUES DE MATRICULES AMB RASPBERRY PI 3+. VISIO ARTIFICIAL.

Salvador Marín i Montero

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

Índex

| | |
|---|----|
| 1. Resum..... | 3 |
| 2. Prefaci..... | 3 |
| 3. Lectura de matrícules..... | 3 |
| 4. Visió artificial..... | 4 |
| 5. Objectiu..... | 6 |
| 6. Hardware utilitzat..... | 7 |
| 7. Software utilitzat..... | 8 |
| 8. Instal·lar el software i programari..... | 9 |
| 9. Muntatge de lectura..... | 14 |
| 10. Desenvolupament de la lectura..... | 18 |
| 11. Discussió..... | 19 |
| 12. Annexe..... | 22 |
| 13. Bibliografia/Webgrafia..... | 28 |

RESUM.

La lectura de matrícules de vehicles es fonamental per als sectors de seguretat i compteig de vehicles en estacionaments i control d'entrades i sortides. Es realitza amb visió artificial per la qual cosa intervé software sofisticat en un entorn controlat per tal de tenir una lectura uniforme i fiable, amb un hardware suficient per donar-hi una resposta efectiva en el mínim temps possible.

PREFACI.

L'idea inicial era fer lectura de caràcters xinesos per, amb un software de OCR, realitzar una conversió del caràcter a un idioma occidental. Aixó, dissortadament, no ha estat possible arran de les primeres proves fetes en un PC amb llenguatge python i llibreries a l'ús, degut a la forta demanda d'espai, potència de càlcul, temps i complexitat que demana aquesta conversió.

Per tant s'han hagut de rebaixar i transformar les demandes inicials, llegint un text curt i definit en un entorn industrial controlat alhora que pràctic. L'exemple fòra extret d'altre projecte, lectura de caps de medicaments, per finalment acabar en lectura de plaques de matrícules de vehicles de l'estat espanyol, ja que se disposa d'impressions d'una pàgina web.

LECTURA DE MATRÍCULES DE VEHICLES.

Així doncs, finalment s'ha optat per aquesta lectura.

Les plaques de matrícula han estat generades amb una web dissenyada a l'ús.

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

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

Amb ella s'han generat un total de 10 imatges de matrícula de l'estat espanyol tal i com mostra l'imatge, que han estat impreses en A4 apaisat tot retallant la part inferior del text de la web.



L'intenció es oferir a la càmera sempre la mateixa imatge de matrícula en la mateixa posició i amb el mateix fons i textura per tal de minimitzar les variables d'entrada a la càmera usada en conjunció amb la raspberry.

És important minimitzar les condicions d'entrada per que, tal i com es descriu a l'apartat següent, la visió artificial demana unes condicions de lectura el més simples possibles.

VISIO ARTIFICIAL.

En principi per fer la lectura de les matrícules s'ha pensat en un muntatge de visió artificial amb OCR (optical characters reconnaissance; reconeixement òptic de caràcters). Per aconseguir-ho, es basa en el següent esquema mostrat a l'imatge.

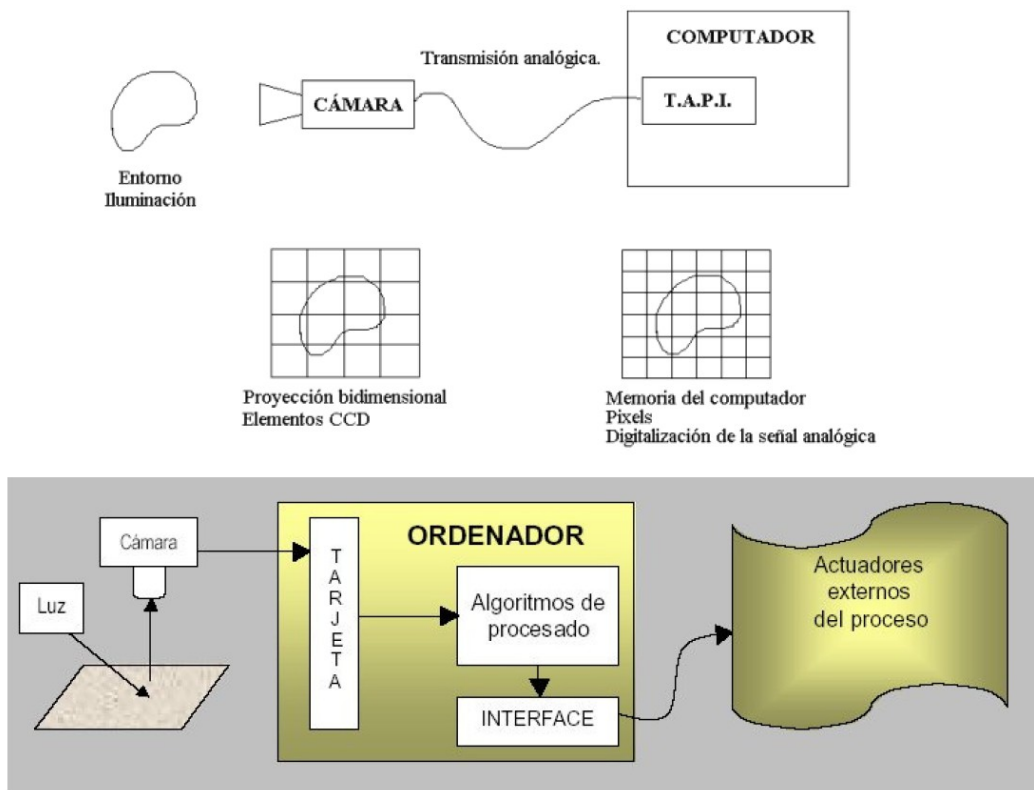


Figura 1. 17 Subsistemas físicos de un equipo de visión artificial

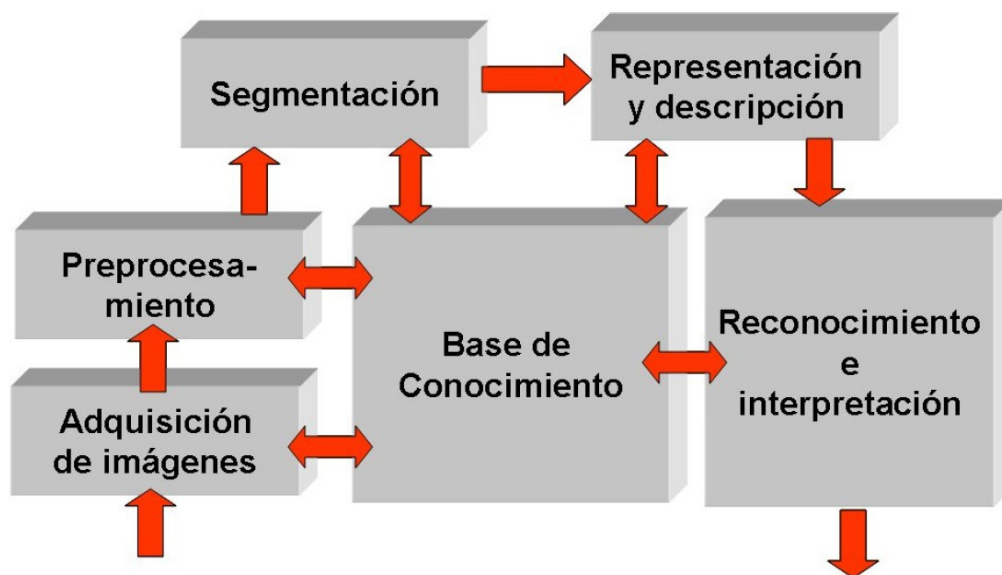


Figura 1. 19 Etapas de una aplicación de visión artificial

A continuació es descriuen les etapes necessàries per desenvolupar el nostre sistema de visió artificial.

La primera etapa és la construcció del sistema de formació de les imatges. El seu objectiu és realçar, mitjançant tècniques fotogràfiques (il·luminació, òptica, càmeres, filtres, pantalles, ...), les característiques visuals dels objectes (formes, textures, colors, ombres, ...). L'èxit de moltes aplicacions depèn d'un bon disseny en aquesta primera etapa.

Un cop adquirida la imatge es passarà a l'etapa de preprocessat. L'objectiu és millorar la qualitat informativa de la imatge adquirida. S'inclouen operacions de millora de la relació senyal-soroll (denoising), SNR, d'atenuar les imperfeccions de l'adquisició causa de la funció de transferència del sistema de captació d'imatges (deconvolution), de regularitzar la imatge, de millorar el contrast o d'optimitzar la distribució de la intensitat (enhancement) o de realçar algunes característiques de la imatge, com vores o àrees.

Segmentació és la fase on es particiona la imatge en àrees amb significat. Per exemple, en una imatge de satèl·lit es determina les zones d'aigua, de cultiu, urbanes, carreteres, ... Hi ha diverses tècniques: umbralizaciones, discontinuïtats, creixement de regions, ús del color o de moviment, etc. Aquestes estratègies han de ser analitzades. Un cop dividida la imatge en zones amb característiques de més alt nivell es passarà a la seva extracció de les característiques. Bàsicament són de tipus morfològic, com l'àrea, perímetre, excentricitat, moments d'inèrcia, esquelets, però també es poden emprar característiques basades en la textura o en el color.

Un projecte de Visió Artificial sol ser de tipus multidisciplinar. L'execució de les diferents etapes, esmentades en l'apartat anterior, requereix del maneig de les següents tècniques:

1- Fotografia i Òptica: Crear l'ambient d'il·luminació adequada en l'adquisició de les imatges, moltes vegades requereix de l'ús de tècniques professionals de fotografia i vídeo. La selecció de l'òptica i de la càmera, l'ús de filtres i polaritzadors, les tècniques d'il·luminació amb

pantalles i l'elecció dels tipus de focus són algunes habilitats que es poden esmentar.

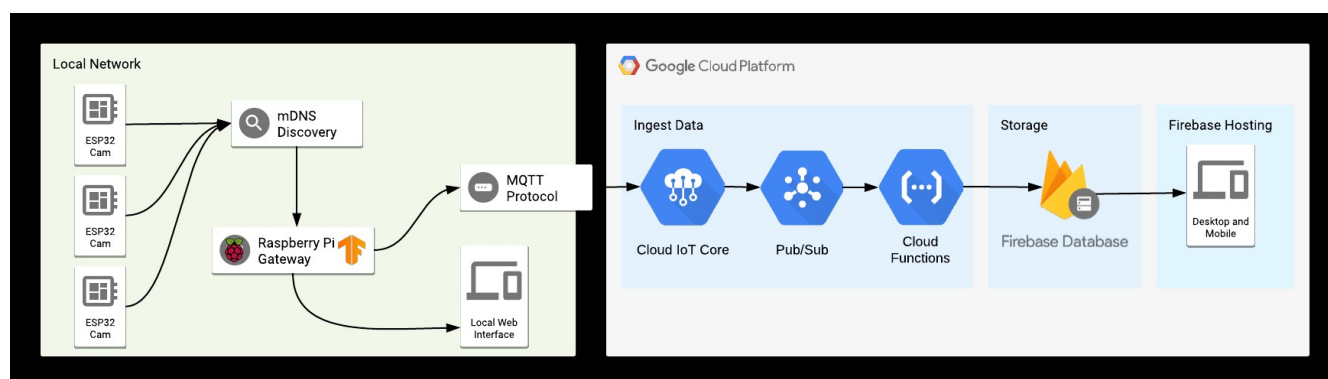
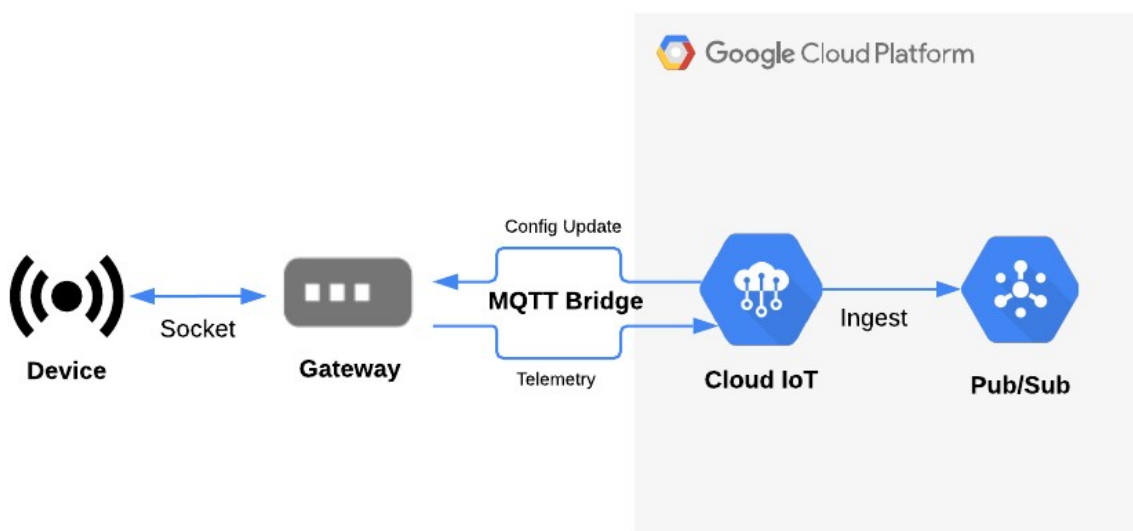
2- Processament Digital de les Imatges (Image Processing): Fa referència als algorismes de computació que converteix la imatge digital adquirida en una altra de més rellevància. És molt difús la separació entre el processament d'imatges i la Visió Artificial.

3- Reconeixement de Patrons (Pattern Recognition): disciplina, dins la Intel·ligència Artificial, dedicada a la classificació dels senyals i de la recerca de patrons existents dins d'aquestes. Es troben incloses les tècniques de classificadors estadístics, Xarxes Neuronals, Sistemes Experts, Lògica Borrosa, ...

4- Computació Gràfica (Computer Graphics): Presenta el problema invers de la Visió Artificial. Si en Visió es vol extreure les característiques físiques de les imatges, la Computació Gràfica es dedica a la presentació visual dels models geomètrics. Cada vegada més, la Visió Artificial emprà la Computació Gràfica per a representar les conclusions extretes de l'anàlisi de les imatges adquirides.

OBJECTIU.

Demostrar que, amb una Raspberry PI 3+, es pot realitzar un sistema de visió artificial que, amb un esquema de Internet of Things (Internet de les coses), pogui pujar una sèrie de 10 lectures de matrícules de vehicles en format d'arxiu CSV a un núvol format Google Drive. Un exemple podria ser el mostrat a les imatges.



Se disposa dels següents elements de hardware per assolir aquesta pràctica:

A collection of Raspberry Pi components is displayed. In the background, there is a white box for the 'Raspberry Pi Universal Power Supply 2.5A, 5.1V' and a red box for the 'Raspberry Pi 3 Model B+'. In the foreground, there is a white and red Raspberry Pi case, a black micro SD card with the Raspberry Pi logo, a small black micro SD card labeled '16GB', and a Raspberry Pi 3 Model B+ circuit board.

The image displays four perspectives of the Nikon COOLPIX P100 camera. The top-left view is a front-three-quarter angle showing the lens with 'NIKON COOLPIX P100' and 'ED ELEM.' text, and the 'COOLPIX' logo on the body. The top-right view is the back of the camera, featuring a large LCD screen showing a landscape photo, a viewfinder, and various control buttons. The bottom-left view is a front-facing angle with the pop-up flash extended. The bottom-right view is the bottom of the camera, showing the tripod mount, battery cover, and the 'COOLPIX P100' branding.

SOFTWARE UTILITZAT.

A l'ordinador o PC:

- Windows 10.
- Putty. Per connectar-nos-en en remot a la Raspberry.
- WinSCP. Amb el mateix fi per fer la transferència de fitxers entre PC i Raspberry.
- Linux.
- Python versió 2 i versió 3 instal·lats.

A la Raspberry PI 3+:

- Sistema operatiu Raspbian Lite Stretch.
- Python versió 2 i versió 3.
- Instal·lació dels següents paquets i programes sota Python:

Pydrive. Llibreria per pujar arxius de la Raspberry a Google Drive.

Numpy. Llibreria matemàtica de python.

Pytesseract. Llibreria de reconeixement de caràcters ORC de python.

Gphoto2. Software en python de captura d'imatges amb càmeres reflex.

OpenCV2. Software escrit en python per tractament d'imatges.

a més de les descrites a les importacions de python fetes al començament dels programes.

INSTAL·LACIÓ DEL SOFTWARE I PROGRAMARI.

Cal mencionar que, a la Raspberry Pi3+ s'ha instal·lat el sistema operatiu Raspbian Linux Lite Stretch en modus textu, per alleugerir la càrrega interna de processament i memòria de la Raspberry.

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

Comencem definint a una carpeta de treball a la Raspberry tots els arxius, llibreries i executables en python amb els quals s'ha de treballar.

Serà la `cd/home/pi`.

Des de la terminal de Putty, es comença per actualitzar el següent.

```
sudo apt-get install pip
```

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

L'intenció és actualitzar el gestor de llenguatge Python per tal d'instal·lar els programes a dalt indicats, que són els que en calen per fer la captura, preprocessat i lectura final dels caràcters impresos a la matrícula.

Un cop fet, es procedeix a instal·lar un lector de caràcters OCR que pugui treballar en python i que sigui prou compacte per ser instal·lat a la memòria SDK de la Raspberry. Es tria el anomenat Pytesseract, que es prou conegut i funciona, d'acord amb les proves prèvies que es realitzaren al PC del curset del CIFO La Violeta, amb suficient rapidesa.

```
sudo pip install pytesseract
```

<https://pypi.org/project/PyTesseract/>

Tot seguit, el software Gphoto2, que permet, amb una càmera abans descrita i una connexió USB a un dels ports de la Raspberry, pendre fotografies, guardar-les a la memòria SDK i, posteriorment, tractar-les. És un software molt voluminós i caldrà fer els passos segons una pàgina on es descriu tot el procés d'instal·lació.

<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/>

Finalment, caldrà fer proves de funcionament i comprovació amb la càmera a dalt mencionada.

Posteriorment, caldrà posar el software de tractament d'imatges OpenCV -4, que ens permet agafar les imatges en color de la càmera digital, fer-les en blanc i negre, treure el soroll i els artefactes o petites taques, i finalment deixar l'imatge tota llesta per fer el reconeixement de caràcters.

<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/>

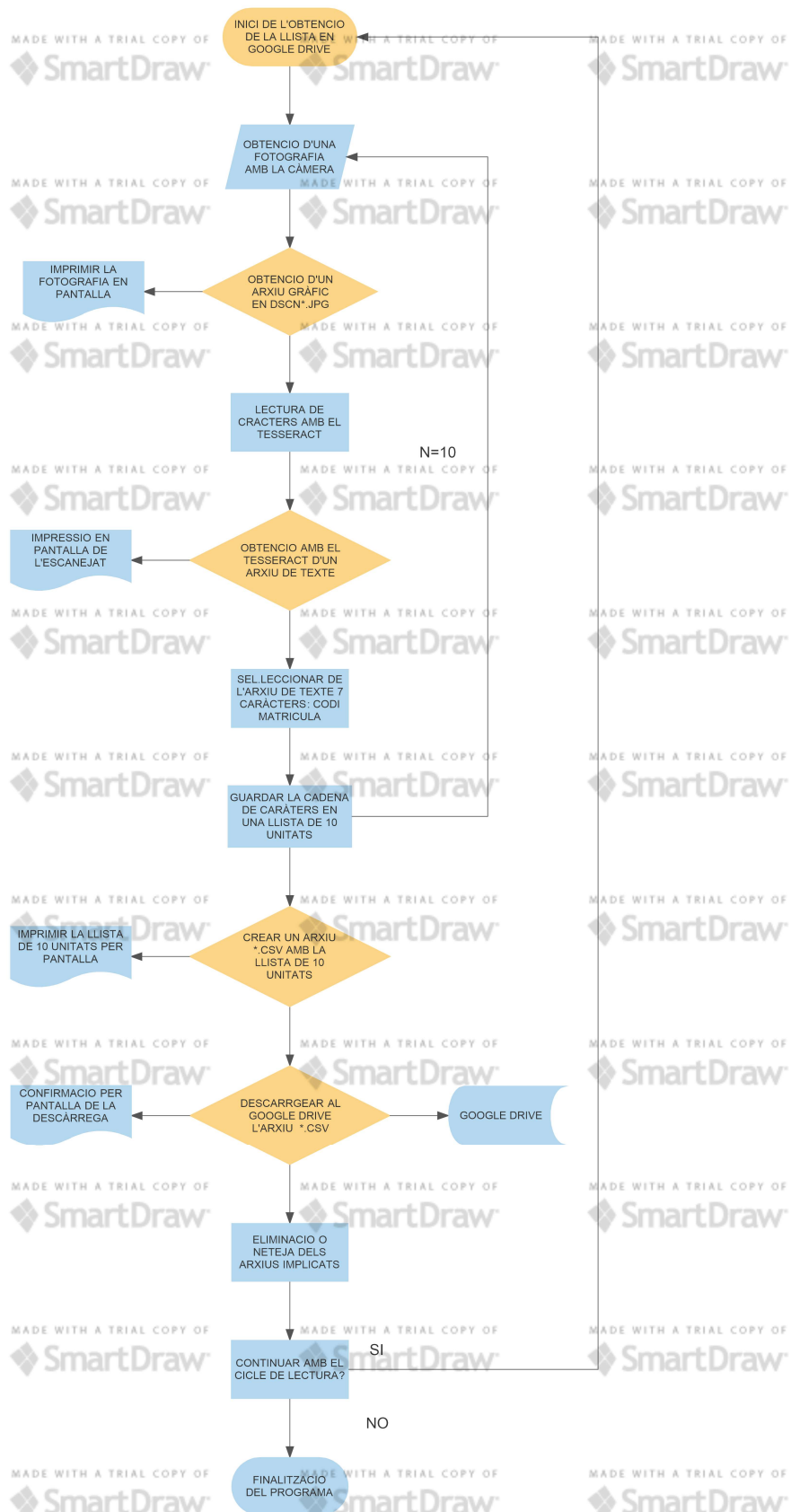
<https://circuitdigest.com/microcontroller-projects/how-to-install-python-opencv-on-raspberry-pi>

També instal·lem la llibreria numpy per tal de disposar de llibreries matemàtiques per tractar les imatges i discriminar les regions de l'imatge que ha de ser reconegudes per OCR.

<https://www.numpy.org/>

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

A continuació es posen els dos programes implicats en el programari. Es posen els codis per tal de que es puguin reproduir a posteriori. Es posa, a més un diagrama de fluxe que descriu el modus de treball i funcionament del programes implicats.



```
provaocr10.py - C:\Users\Alumne\Desktop\provaocr10.py (3.7.3)
File Edit Format Run Options Window Help

from PIL import Image

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

def main():
    lista = []
    comptador = 1

    while comptador<11:
        time.sleep(1)
        os.system('gphoto2 --capture-image-and-download --filename DSCN1

        img2 = cv2.imread('DSCN1.JPG',0)

        img2gray = cv2.bilateralFilter(img2, 11, 17, 17)
        gray1 = cv2.GaussianBlur(img2gray, (7, 7), 3)
        crop_img = gray1[500:1828, 800:3200]

        #Read the number plate
        text = pytesseract.image_to_string(crop_img, config='--psm 3')
        text = text[1:]
        print("Detected Number is:",text)

        cv2.imwrite('DSCN1.JPG',crop_img)

        #os.system('tesseract DSCN1.JPG 1 -ll')
        print(text)
        lista.append(text)

        f = open ('1.txt','w')
        for i in lista:
            f.write(i)
        f.close()

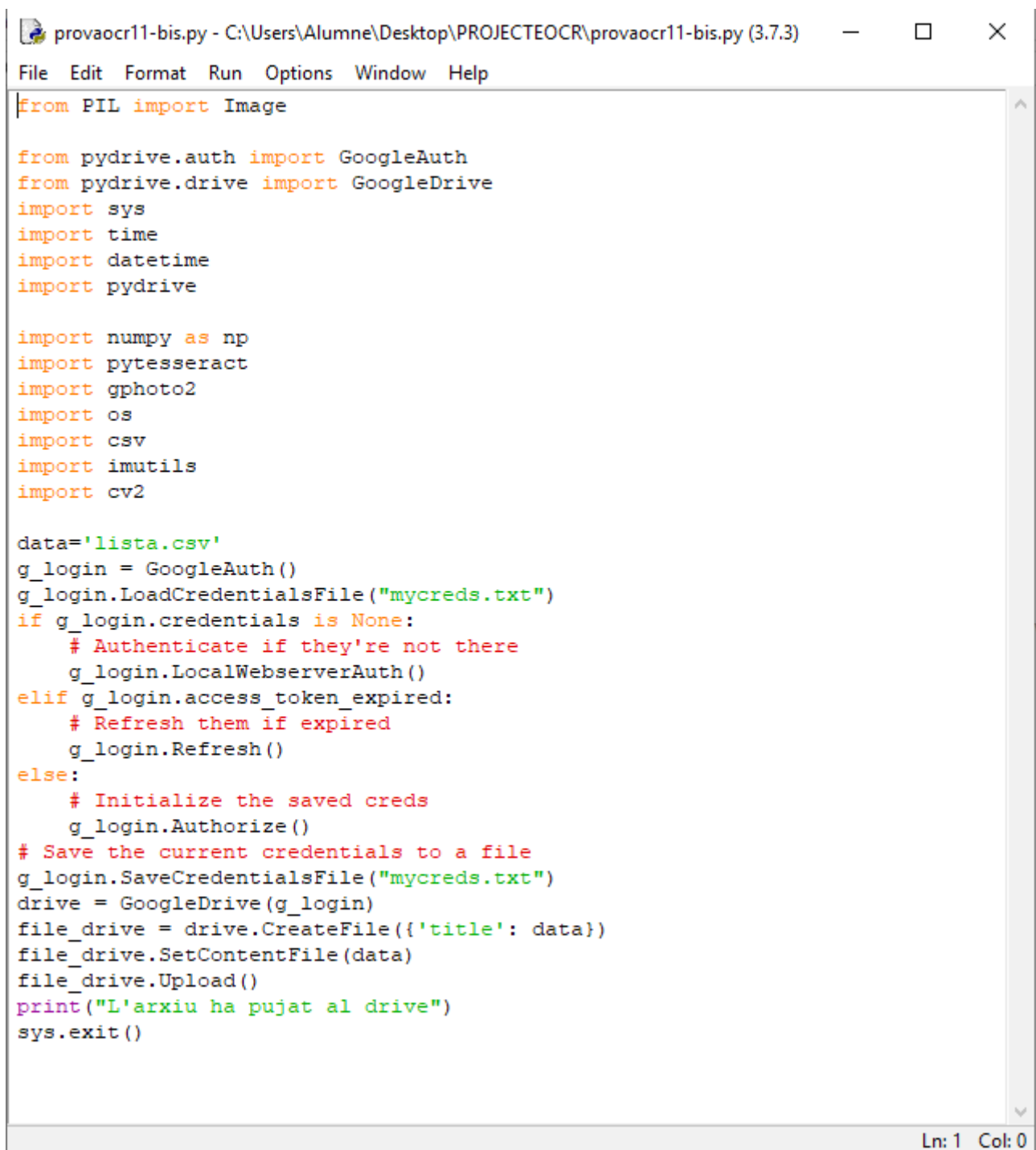
        comptador=comptador+1

    print (lista)
    file = open('lista.csv', 'w')
    spamreader = csv.writer(file)
    spamreader.writerow(lista)
    file.close()

    #os.system(cd /home/pi)
    #os.system(mkdir Lista)
    #os.system(cd Lista)

main()
```

Ln: 51 Col: 0



```
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

Aquests dos arxius cal que primer s'executin en el PC, per tal de demanar, des d'una API Google, un arxiu de credencials.

Primer cal tenir habilitada una Google API tal i com es mostra en l'imatge adjunta. Un cop demanat l'arxiu de credencials s'executa al PC, generant-s'hi un arxiu anomenat mycreds.txt, al qual copiarem l'informació continguda a l'arxiu client_secrets.json.

Llavors es pot endur l'arxiu mycreds.txt, i els dos programes en python a la raspberry i des d'allà ja es pot treballar amb la lectura de les 10 imatges preses per la càmera.

Google APIs

CLASSE-27-MAIG

API

APIs y servicios

Panel de control

Biblioteca

Credenciales

Credenciales

Pantalla de consentimiento de OAuth

Verificación de dominio

Crear credenciales

Eliminar

Crea credenciales para acceder a tus API activadas. Para obtener más información, consulta la [documentación de autenticación](#).

IDs de cliente de OAuth 2.0

| Nombre | Fecha de creación | Tipo | ID de cliente |
|----------------------------|-------------------|------|-----------------|
| CLASSE IOT VIOLETA 27 MAIG | 27 may. 2019 | Web | 634082673785-cr |

| Nombre | Fecha de modifica... | Tipo |
|----------------------------|----------------------|----------------------|
| arxiuspujadadriveDHT11 | 18/06/2019 12:15 | Carpeta de archivos |
| AdafruitDHT.py | 21/08/2018 16:25 | Python File |
| arxiuspujadadriveDHT11.zip | 18/06/2019 12:14 | Archivo WinRAR Z... |
| client_secrets.json | 18/06/2019 13:13 | Archivo JSON |
| lista.csv | 18/06/2019 10:57 | Hoja de cálculo d... |
| mycreds.txt | 18/06/2019 13:14 | Documento de tex... |
| provaocr11.py | 18/06/2019 12:24 | Python File |
| provaocr11-bis.py | 18/06/2019 14:08 | Python File |

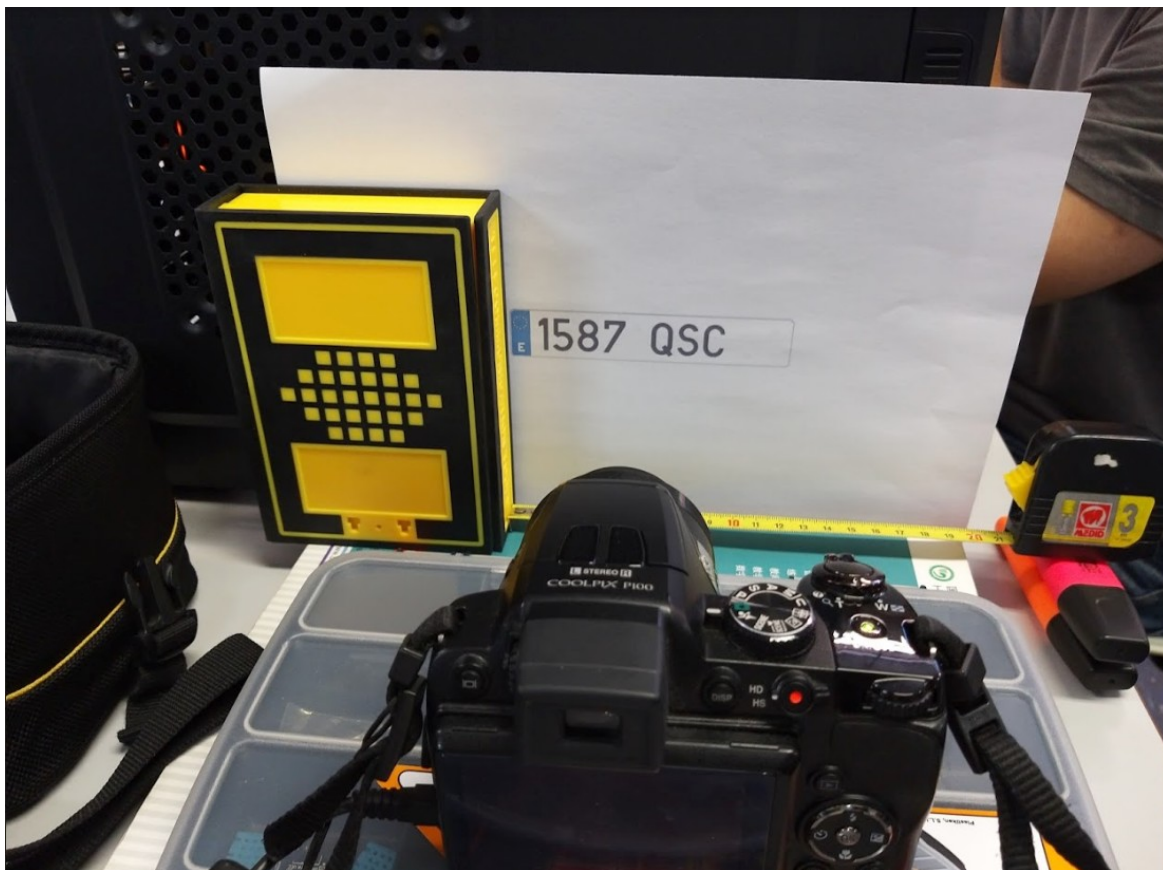
MUNTATGE DE LECTURA

Per realitzar la lectura dels fulls amb les plaques de matrícula impreses, caldrà implementar un muntatge de la càmera amb situació adequada per la presa de fotografies que, amb la raspberry, podem preprocessar i tractar per fer el OCR.

Se detallen a continuació una sèrie d'imatges on es modtra el muntatge realitzat a classe. S'ha de tenir en compte que es parteix d'uns mitjans molt precaris però no per aixó menys efectius.

Finalment, es realitza una captura de pantalla on es mostra la lectura de la fotografia amb la matrícula llegida.

I, per últim i com a tall d'exemple, un exemple d'arxiu lista.csv on es poden veure els cràcters de les matrícules llegides.







```

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 cámara
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'')

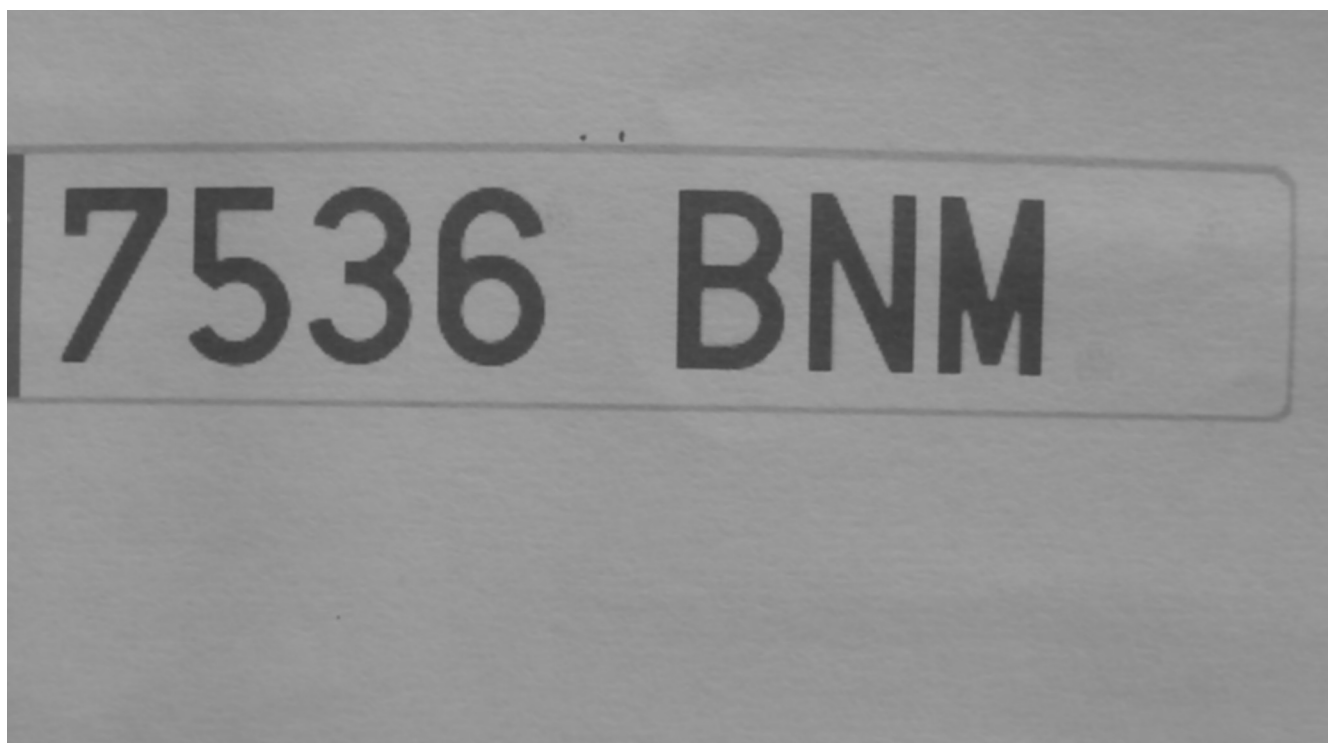
[u'']
root@raspberrypi:/home/pi# python provaocr10.py
El nuevo fichero está en la ubicación /store_00010001/DCIM/100NIKON/DSCN0089.JPG en la cámara
El fichero DSCN1.JPG existe. ¿Sobreescribir? [y|n] y
Guardando el fichero como DSCN1.JPG
Deleting file /store_00010001/DCIM/100NIKON/DSCN0089.JPG on the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

DESENVOLUPAMENT DE LA LECTURA.



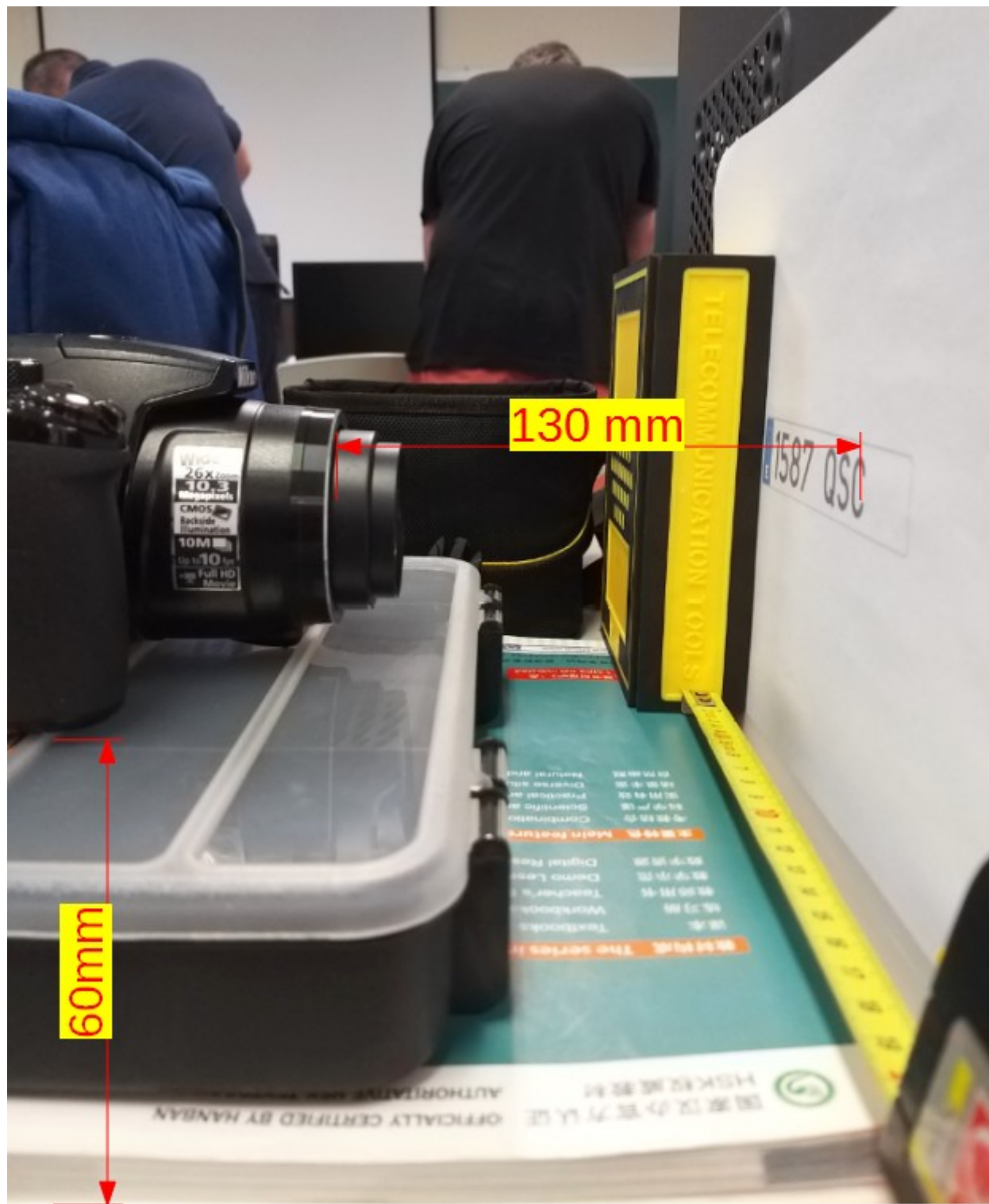
Aquesta seria la lectura de matrícula resultant de la fotografia de la càmera NIKON ja convenient retallada, passada a escala de grisos i trets tot el soroll i artefactes (o petites taques) presents a l'imatge. Comparar amb les imatges prèvies de com era convenient el format A4 apaisat d'impressió.

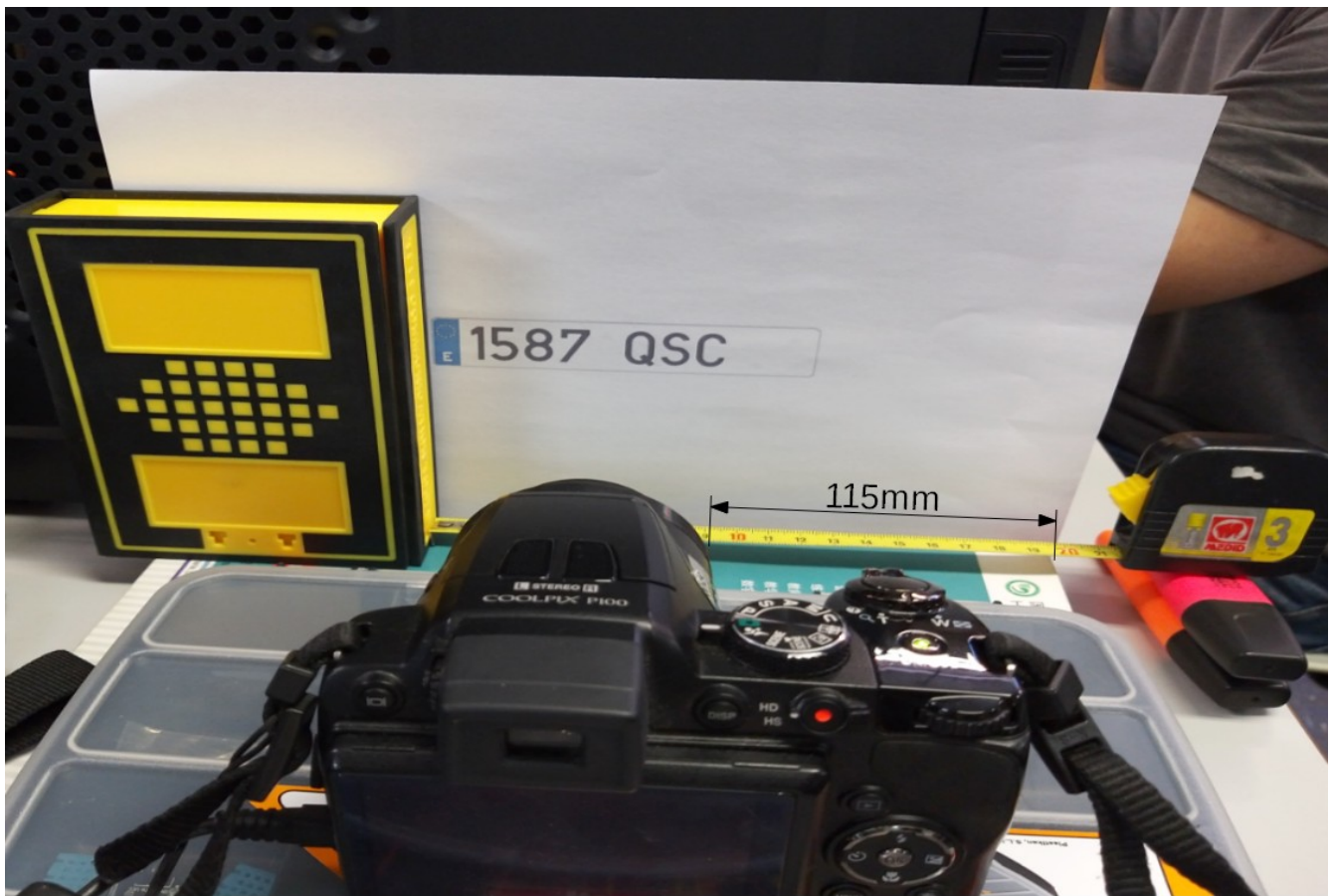
DISCUSSIÓ DEL MUNTATGE I DESENVOLUPAMENT GENERAT A LA PRÀCTICA.

Cal tenir en compte els següents fenòmens per portar a la pràctica aquesta pràctica. Aquests fenòmens són resultat d'unes 40 hores de pràctiques amb la programació OCR y de python, el muntatge, la realització i el posterior anàlisi i revisió d'aquesta pràctica.

- La càmera ha d'estar amb la bateria plena per funcionar, no s'ha d'alimentar amb el USB de 5V de la Raspberry, per que si no li treu força de funcionament a aquesta última.
- La SDK card de la càmera ha d'estar convenientment formatejada.
- Les fotografies es carreguen al directori allà on s'executa el codi python i el seu programari, per al nostre cas cd/home/pi
- Les fotografies han de tenir fons blanc i no de color, sino el OCR tesseract no funciona bé, de fet no pren lectura.
- Les captures de caràcters depenen bàsicament de l'il.luminació circundant, per tant es recomana fer unes poques proves abans de fer al sèrie de 10 fotografies.
- Les lectures a vegades són incomplertes, podent-se deixar alguns caràcters en alguns casos.
- L'escriptura a mà no es pot reconèixer.
- Cal un format únic de lectura per tal de simplificar la lectura i que siguin totes uniformes amb una alçada i format unic de lletra.
- La distància foacl és importantíssima. Està determinada en aquest muntatge en 130mm (veure fotografies).
- L'il.luminació també és cabdal. Cal evitar l'il.luminació amb fluorescents i anar a una llum difosa, essent la millor la llum solar indirecta.
- La posició de la fulla també és molt important, per que qualsevol alabeig, vinclament a la fulla o inclinació o descentratge pot ocasionar la falla de la lectura. Ha de ser totalment vertical.
- La cadena de caràcters s'ha de situar preferiblement en el centre de la lectura de la càmera.
- L'imatge està reduïda segons les següents operacions realitzades amb el software Open CV:
 - Canvi a escala de grisos de l'imatge en color original.
 - Sense soroll a l'imatge, es a dir, sense punts ni línies que puguin confondre el OCR.
- S'aplica un retall del fons per treballar nomès amb el marc amb els caràcters tot eliminant els voltants.

Es mostra a continuació les cotes de situació de la càmera respecte a la fulla a llegir. Cal respectar-les rigorosament per reproduir les mateixes condicions.





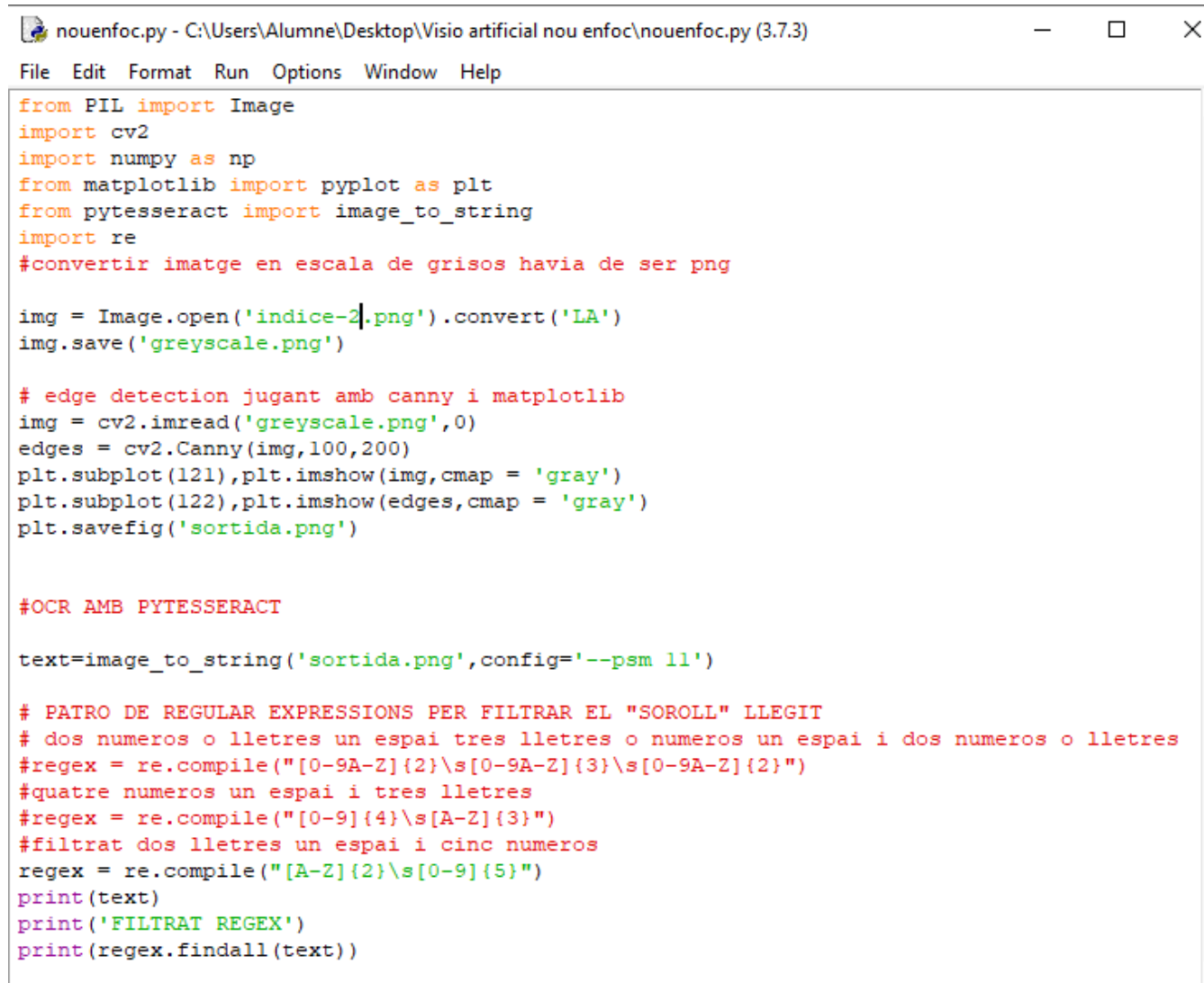
- Aquestes cotes són molt importants. Normalment en muntatges industrial de visió artificial tant l'il.luminació com la distància focal com la situació de la càmera són determinades en funció de moltes proves, amb el mètode d'assaig i error, fins que finalment l'experiència aconseguida estableix una posició de conjunt que fa que les lectures siguin finalment correctes.

- També cal remarcar que proves realitzades amb càmeres reflex a l'exterior, prenent com a base les fotografies preses, s'han de poder tractar amb unes ordres de OpenCV per eliminar el soroll en imatges de pur blanc i negre o binàries. Segons sembla, també dóna resultats positius.

ANNEXE: ALTRE METODE AMB EXPRESIONS REGULARS.

Normalment, sempre hi han més camins per aconseguir el resultat demanat en programació. Amb aquesta intenció, es mostra una camí alternatiu amb software OPEN CV amb el paràmetre canny que genera molt soroll en l'imatge. Aquest soroll o artefactes, com a vegades també s'anomena, cal netejar-lo per tal d'aconseguir una lectura òptima de caràcters.

Es detalla a continuació el codi resultant en python.



```
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')

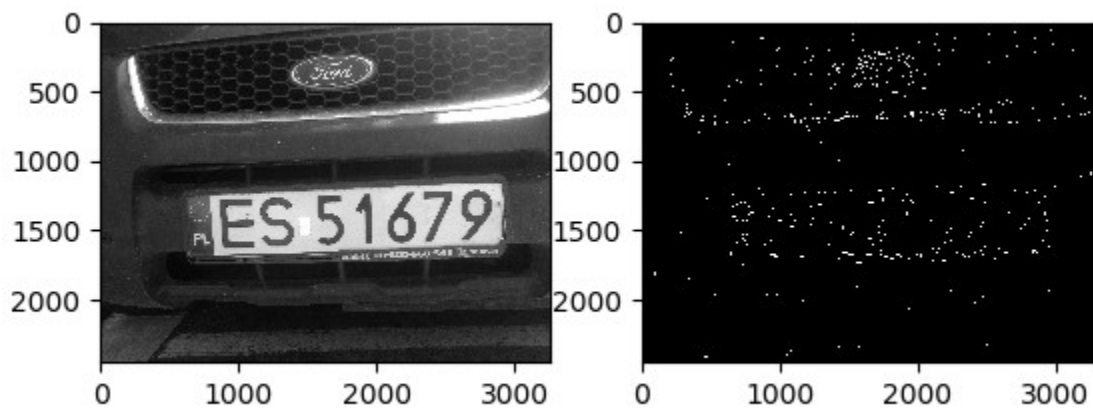
# PATRO DE REGULAR EXPRESSIONS PER FILTRAR EL "SOROLL" LLEGIT
# dos numeros o lletres un espai tres lletres o numeros un espai i dos numeros o lletres
#regex = re.compile("[0-9A-Z]{2}\s[0-9A-Z]{3}\s[0-9A-Z]{2}")
#quatre numeros un espai i tres lletres
#regex = re.compile("[0-9]{4}\s[A-Z]{3}")
#filtrat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

Es mostra en el codi els mètodes Canny i d'expressions regulars que calen per llegir una placa de matrícula que es detalla a continuació.



```
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



BIBLIOGRAFIA I ENLLAÇOS WEB DE REFERÈNCIA.

Treball amb l'inies de texte i arxius.

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=1kMBXdTqDZKZlwShqgn4Cw&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>

Treball amb arxius csv.

https://www.google.cat/search?ei=ks4BXZvXMly7gwfxYa2lAg&q=como+crear+un+archivo+csv+en+python&oq=como+crear+un+archivo+csv+en&gs_l=psy-ab.1.0.0i6j0i22i30.4172.4645..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?](https://www.google.cat/search?source=hp&ei=pZlIXdcNlyeUIbMhEA&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)

[source=hp&ei=pZlIXdcNlyeUIbMhEA&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://www.google.cat/search?source=hp&ei=pZlIXdcNlyeUIbMhEA&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>

////////////////////////////////////

Treball amb raspberry pi i opencv.

[https://www.google.cat/search?](https://www.google.cat/search?q=tesseract+ocr+car+plate+numbers&ei=_hgCXaqzA8GwaaqLiPgM&start=10&sa=N&ved=0ahUKEwiqqquOIObiAhVBWBoKHaoFAs8Q8NMDCMgB&biw=1920&bih=937)

[q=tesseract+ocr+car+plate+numbers&ei=_hgCXaqzA8GwaaqLiPgM&start=10&sa=N&ved=0ahUKEwiqqquOIObiAhVBWBoKHaoFAs8Q8NMDCMgB&biw=1920&bih=937](https://www.google.cat/search?q=tesseract+ocr+car+plate+numbers&ei=_hgCXaqzA8GwaaqLiPgM&start=10&sa=N&ved=0ahUKEwiqqquOIObiAhVBWBoKHaoFAs8Q8NMDCMgB&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?](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)

[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://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://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://coday.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://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://coday.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_4XJvIfGKlwTa64ho&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=62cM2Ni2UIM>

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

Treball de raspberry pi amb càmera NIKON i 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/>

Treball de python amb 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.....0i71j0i67j0i22i30.37hadx3anK0

<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..0i131j0i22i30j0i22i10i30j0i19j0i13i30i19j0i22i30i19.QN-fSJKAwg

<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+raspberrypi+3+teressact&oq=how+to+read+chinese+characters+with+a+raspberrypi+3+teressact&gs_l=psy-ab.3...6715.12165..13786...1.0..0.195.1101.8j3.....0....1..gws-wiz.....33i160j33i21.2z-SjID44yM

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

////////////////////////////////////

Miscel.lania.

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>

Treball de matrícules per imprimir.

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