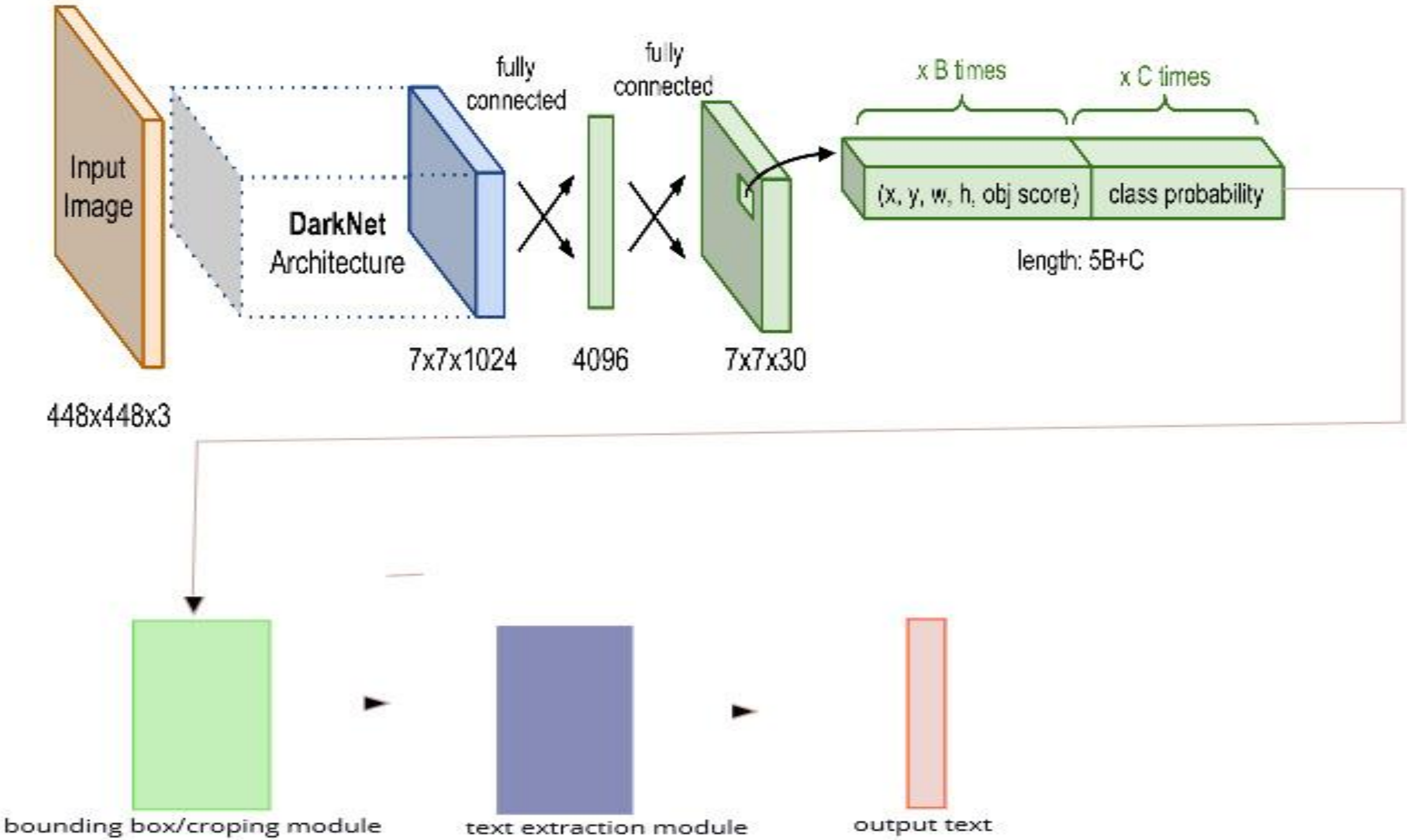
License Plates Detection and Text extraction

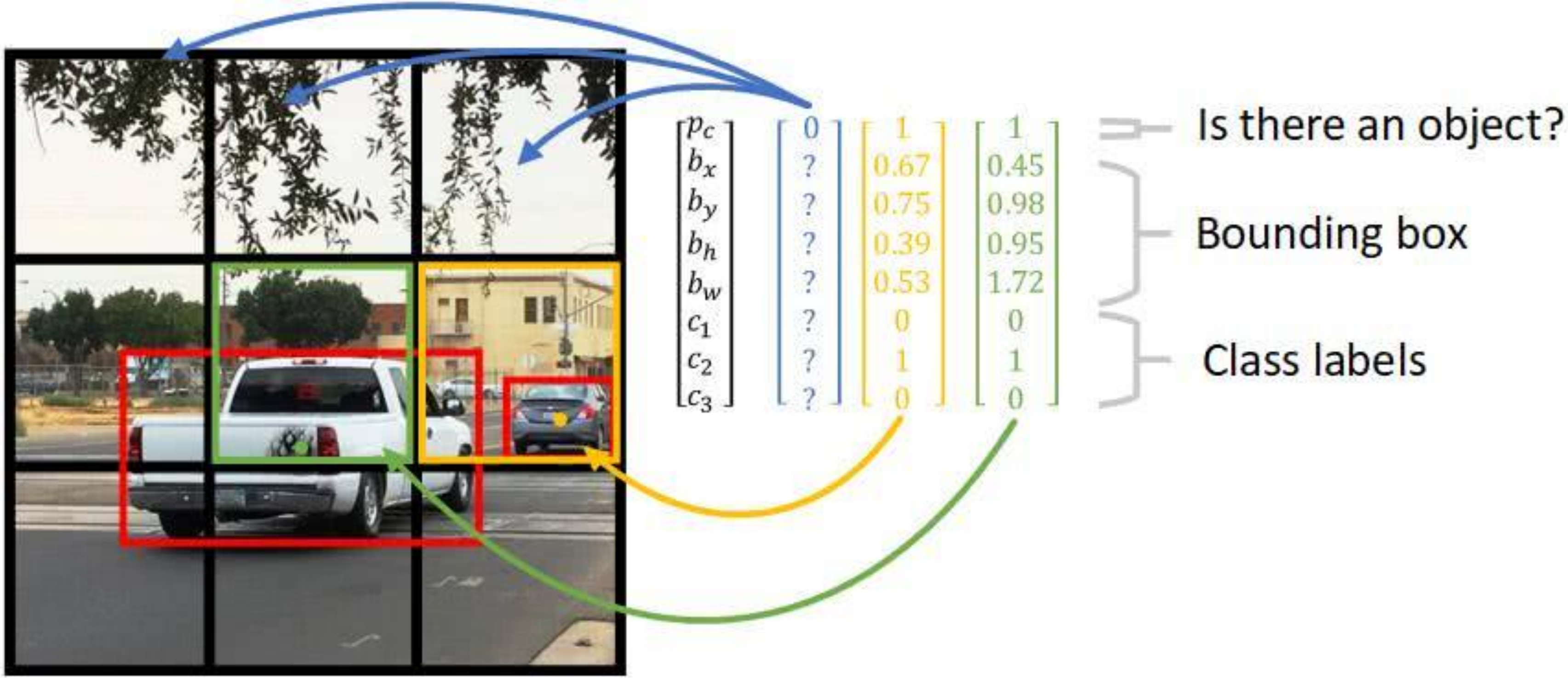
The System Architecture


Input Image
448x448x3
DarkNet Architecture
7x7x1024
fully connected
4096
fully connected
7x7x30
x B times
x C times
(x, y, w, h, obj score)
class probability
length: 5B+C
bounding box/croping module
text extraction module
output text

YOLOV3 (You Only Look Once v3)

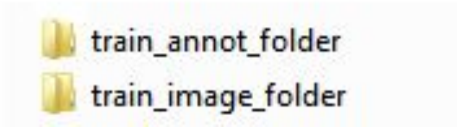
- A network for object detection. The object detection task consists in determining the location on the image where certain objects are present, as well as classifying those objects in one step.
- YOLOv3 has few incremental improvements on YOLOv2.
- Feature Extractor: darknet53 architecture with 53 convolutional layers.
- Keras implementation

YOLO



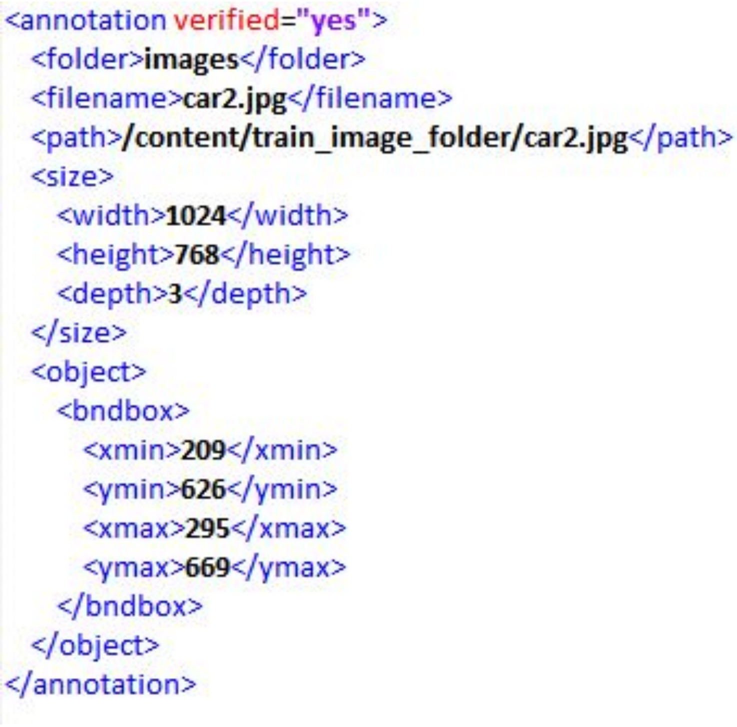
$$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0.67 \\ 0.75 \\ 0.39 \\ 0.53 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0.45 \\ 0.98 \\ 0.95 \\ 1.72 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Data preparation



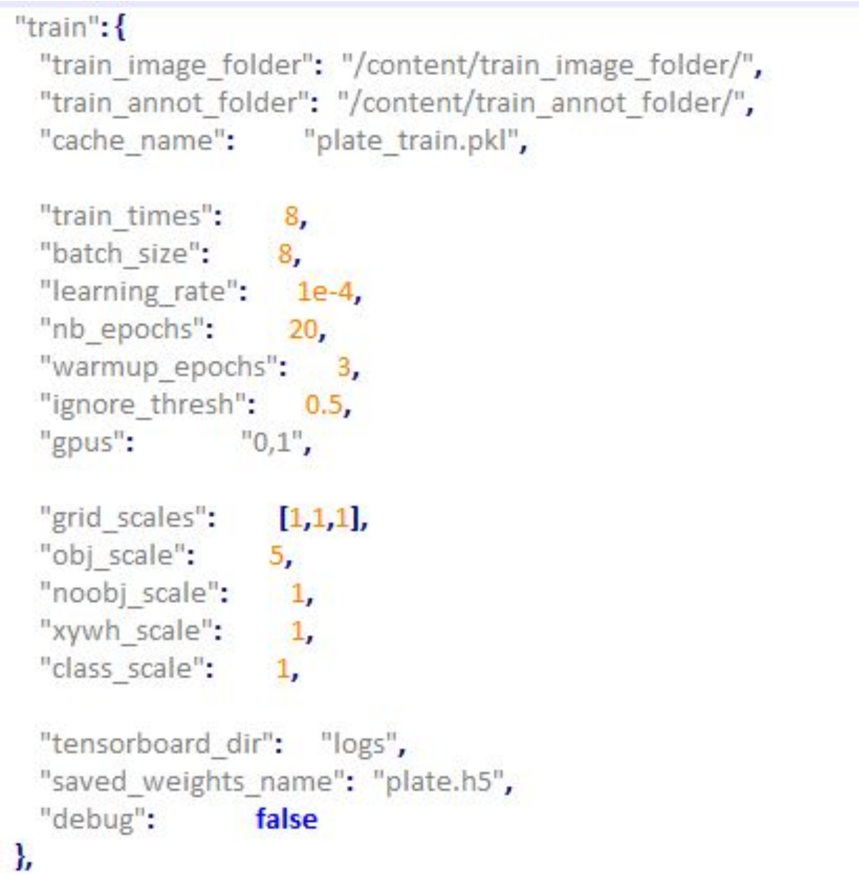
train_annot_folder

train_image_folder



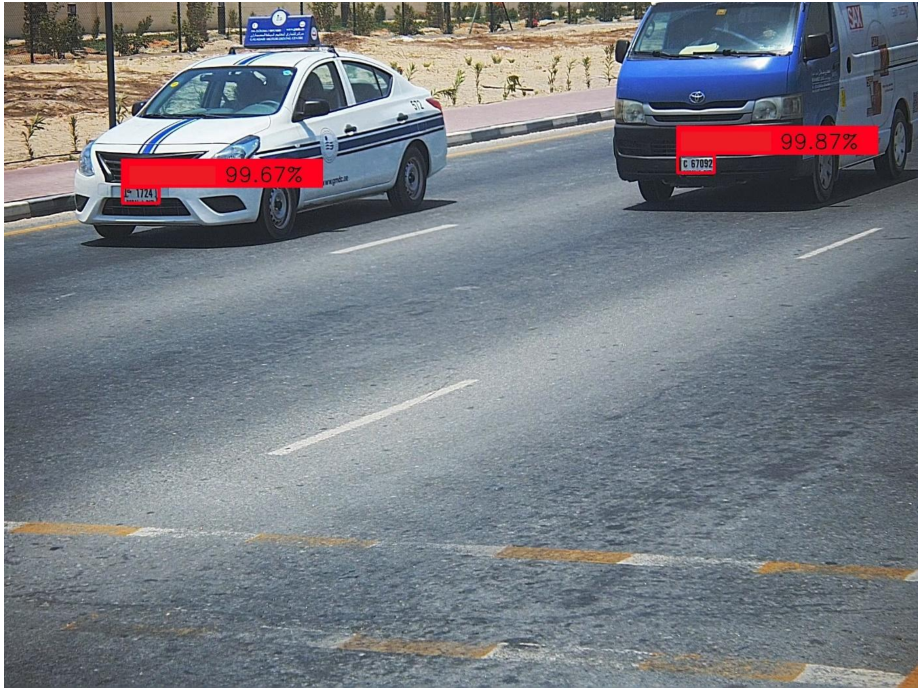
```xml
<annotation verified="yes">
  <folder>images</folder>
  <filename>car2.jpg</filename>
  <path>/content/train_image_folder/car2.jpg</path>
  <size>
    <width>1024</width>
    <height>768</height>
    <depth>3</depth>
  </size>
  <object>
    <bndbox>
      <xmin>209</xmin>
      <ymin>626</ymin>
      <xmax>295</xmax>
      <ymax>669</ymax>
    </bndbox>
  </object>
</annotation>
```

Training

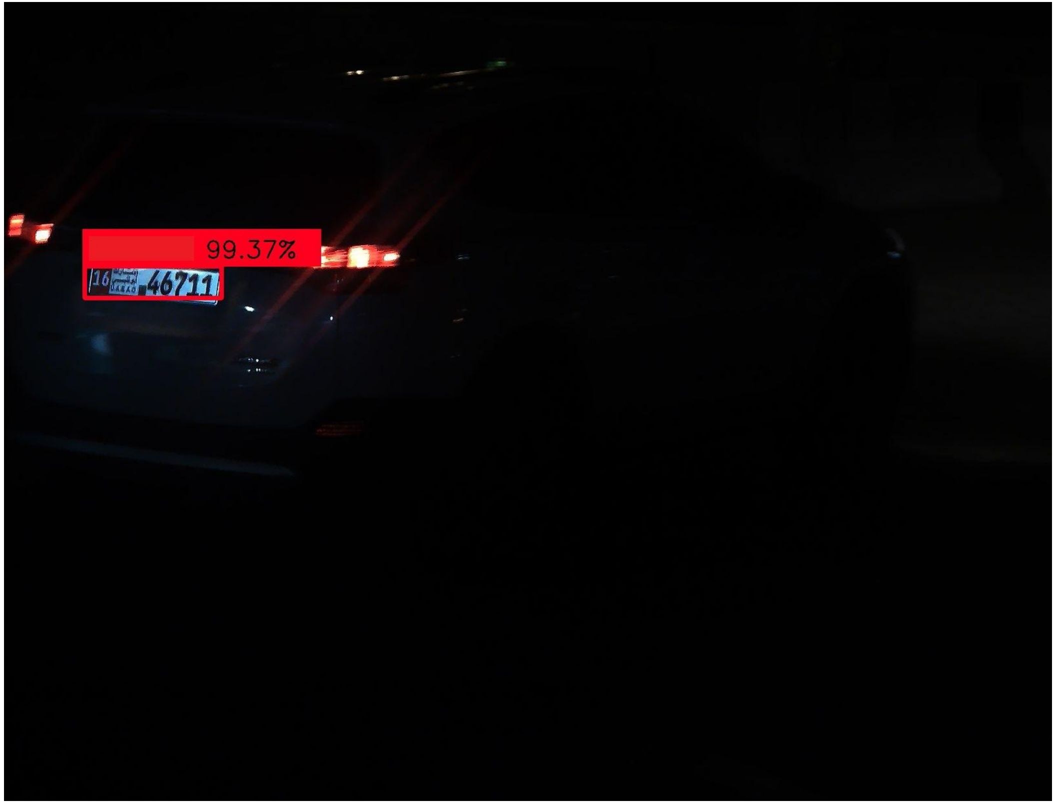


```
"train":{
  "train_image_folder":  "/content/train_image_folder/",
  "train_annot_folder":  "/content/train_annot_folder/",
  "cache_name":        "plate_train.pkl",

  "train_times":      8,
  "batch_size":       8,
  "learning_rate":     1e-4,
  "nb_epochs":        20,
  "warmup_epochs":      3,
  "ignore_thresh":     0.5,
  "gpus":            "0,1",

  "grid_scales":      [1,1,1],
  "obj_scale":        5,
  "noobj_scale":       1,
  "xywh_scale":        1,
  "class_scale":      1,

  "tensorboard_dir":    "logs",
  "saved_weights_name":  "plate.h5",
  "debug":           false
},
```

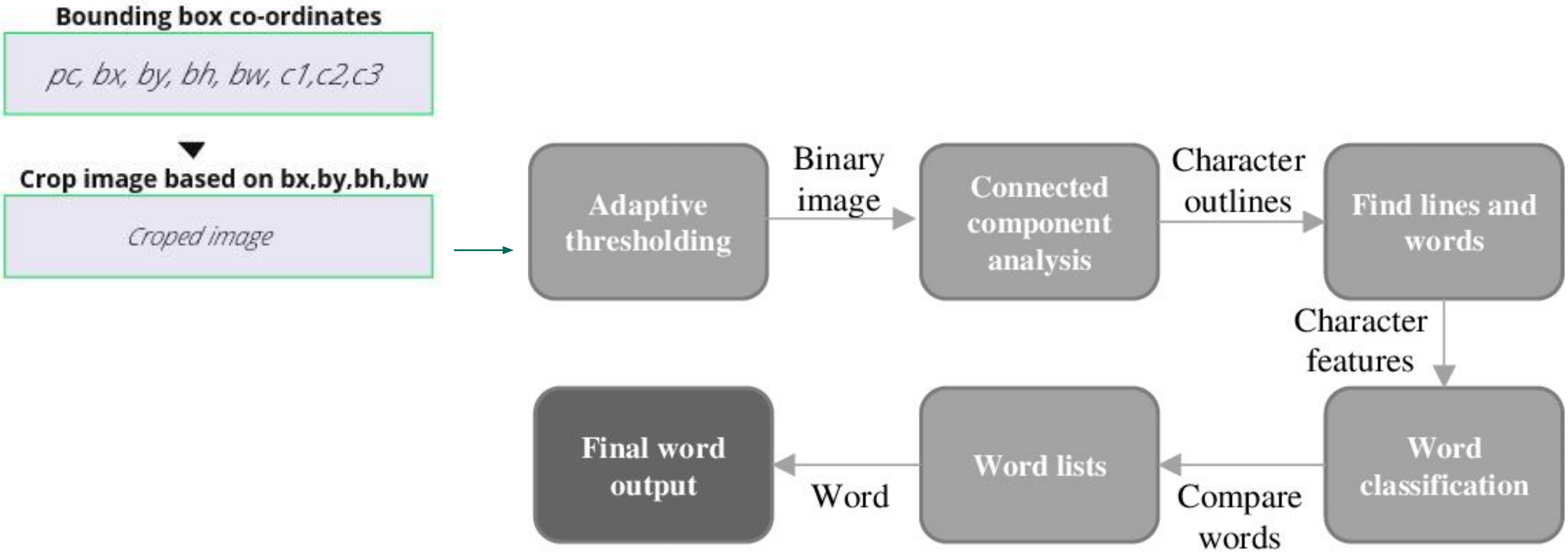
Detection results



Detection results



Extraction



Bounding box co-ordinates

pc, bx, by, bh, bw, c1,c2,c3

Crop image based on bx,by,bh,bw

Croped image

Adaptive thresholding

Binary image

Connected component analysis

Character outlines

Find lines and words

Character features

Word classification

Compare words

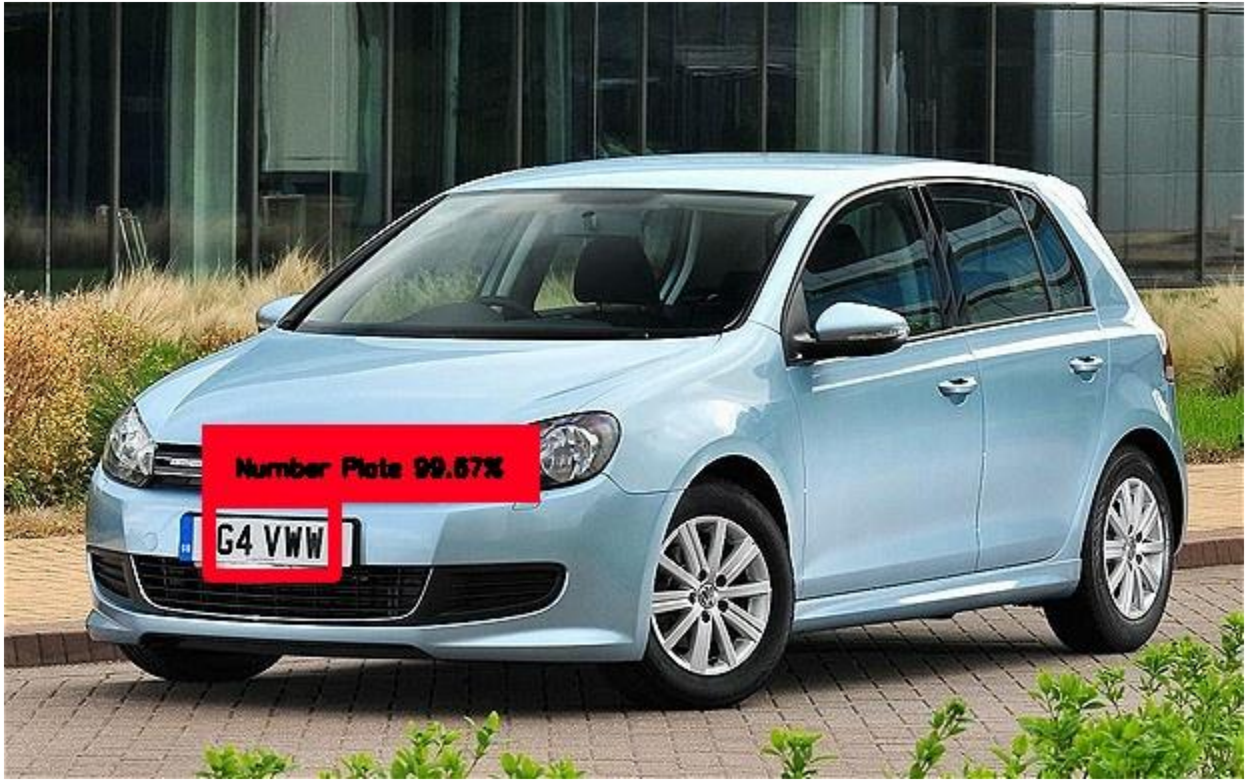
Word lists

Word

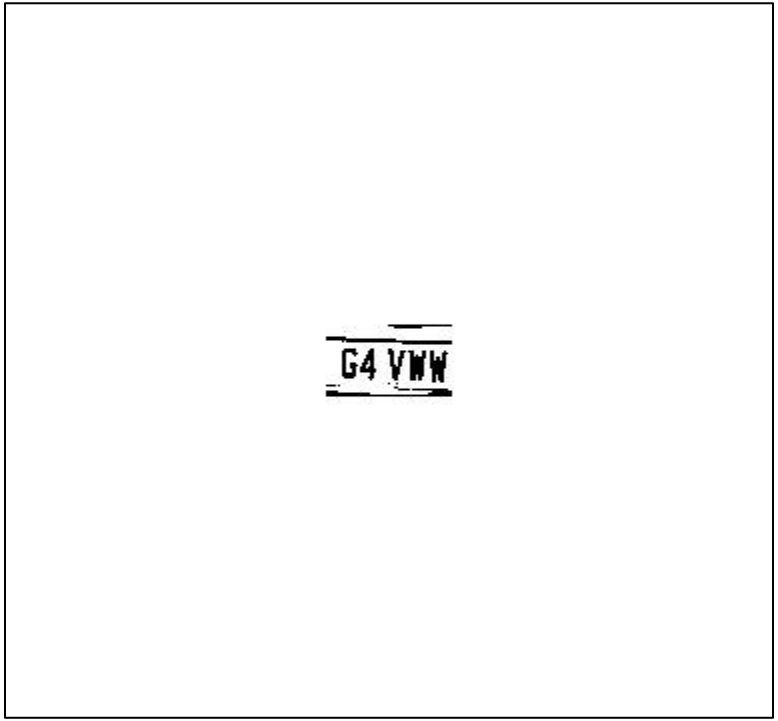
Final word output

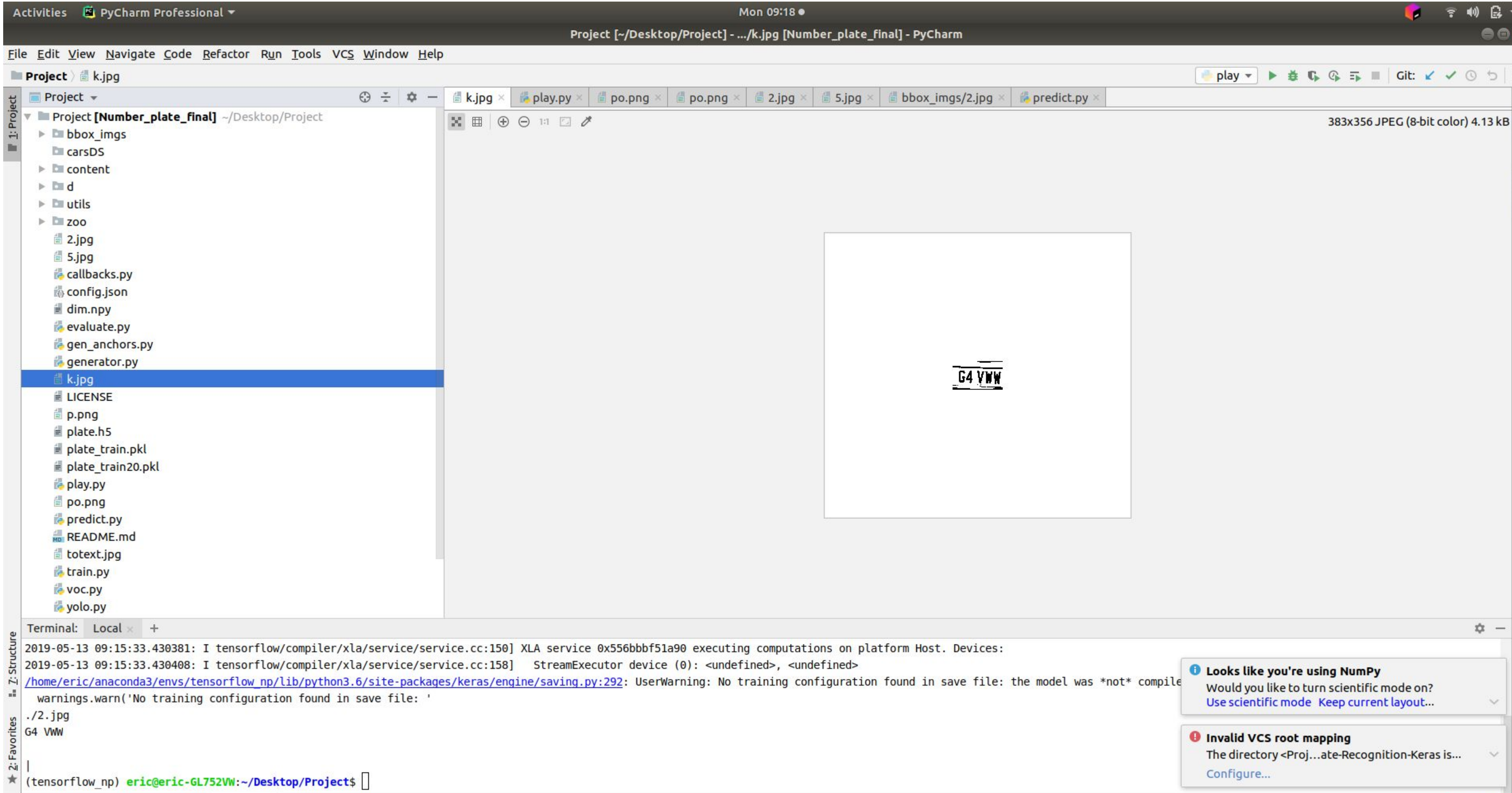
RESULT

G4 VWW


Number Plate 99.67%
G4 VWW

Preprocessed Image

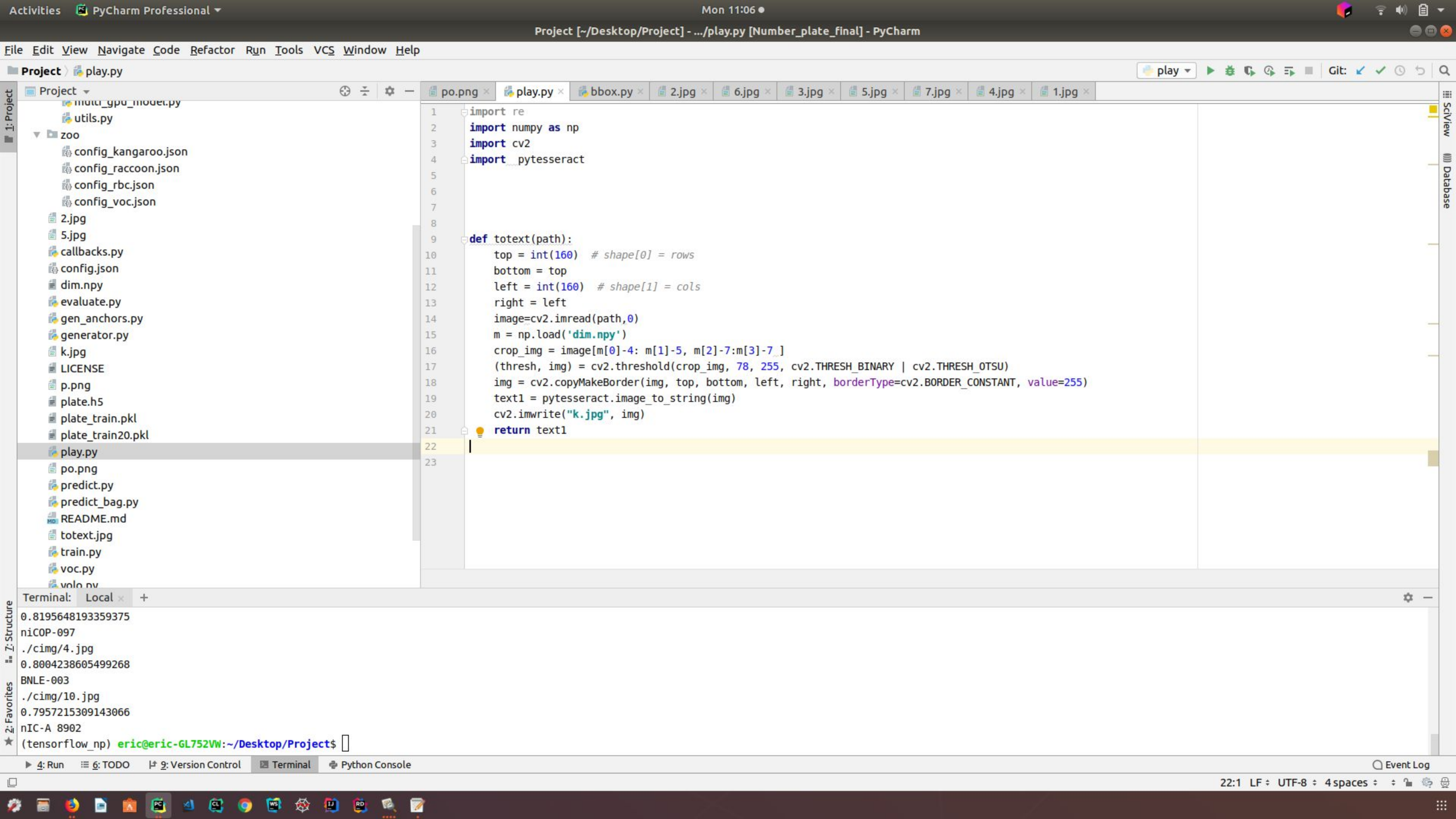




```
2019-05-13 09:15:33.430381: I tensorflow/compiler/xla/service/service.cc:150] XLA service 0x556bbbf51a90 executing computations on platform Host. Devices:
2019-05-13 09:15:33.430408: I tensorflow/compiler/xla/service/service.cc:158]   StreamExecutor device (0): <undefined>, <undefined>
/home/eric/anaconda3/envs/tensorflow_np/lib/python3.6/site-packages/keras/engine/saving.py:292: UserWarning: No training configuration found in save file: the model was *not* compile
  warnings.warn('No training configuration found in save file: '
./2.jpg
G4 VWW

|
(tensorflow_np) eric@eric-GL752VW:~/Desktop/Project$
```

Looks like you're using NumPy
Would you like to turn scientific mode on?
Use scientific mode Keep current layout...

Invalid VCS root mapping
The directory <Proj...ate-Recognition-Keras is...
Configure...



```python
import re
import numpy as np
import cv2
import  pytesseract



def totext(path):
    top = int(160)  # shape[0] = rows
    bottom = top
    left = int(160)  # shape[1] = cols
    right = left
    image=cv2.imread(path,0)
    m = np.load('dim.npy')
    crop_img = image[m[0]-4: m[1]-5, m[2]-7:m[3]-7 ]
    (thresh, img) = cv2.threshold(crop_img, 78, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)
    img = cv2.copyMakeBorder(img, top, bottom, left, right, borderType=cv2.BORDER_CONSTANT, value=255)
    text1 = pytesseract.image_to_string(img)
    cv2.imwrite("k.jpg", img)
    return text1

```

```
0.8195648193359375
niCOP-097
./cimg/4.jpg
0.8004238605499268
BNLE-003
./cimg/10.jpg
0.7957215309143066
nIC-A 8902
(tensorflow_np) eric@eric-GL752VW:~/Desktop/Project$
```

Thanks for listening