Computer Vision for Tunisian License Plate Recognition Challenge

Author : Hamza Azizi
Code Source for the project:

https://colab.research.google.com/drive/1RT-esqInJLKDb1cPhJcBq8FE_GMdxIXp?usp=sharing

Abstract

In this paper, we propose an object detection and recognition methodology applied to License Plate Recognition Challenge Dataset.
You can access the challenge here:
https://zindi.africa/competitions/ai-hack-tunisia-2-computer-vision-challenge-2

I. Introduction

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos, and other visual inputs and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see. In the following project, we will understand how to detect License plates using CNN Models and recognize the License number plates using Python. We will utilize OpenCV for this project to identify the license number plates and the python pytesseract for the characters and digits extraction from the plate.
License plate detection is identifying the part of the car that is predicted to be the number plate. Recognition is identifying the values that make up the license plate. License plate detection and recognition is the technology that uses computer vision to detect and recognize a license plate from an input image of a car. This technology applies in many areas. On roads, it is used to identify the cars that are breaking the traffic rules. In security, it is used to capture the license plates of the vehicles getting into and out of certain premises. In parking lots, it is used to capture the license plates of the cars being parked. The list of its applications goes on and on.

II. Pipeline

The pipeline we will be going through will be presented in this section. The Challenge will be divided mainly into two parts:
- Licence Plate Detection
- Licence Plate Recognition

a) Licence Plate Detection: The competition host will provide a dataset that will be used to accomplish this task using a Convolutional Neural Network, then extracting the plate from the car image.

b) Licence Plate Recognition: To realize this task we will be using the OCR - pytesseract tool to recognize the content in the plate image given.
We will do some transformations on the image to better recognize the plate numbers.

III. License Plate Detection

In this section we will look at the data and the two deep learning models used to perform the detection in deep neural networks. In this context, we will be using either Transfert learning or building a CNN model from scratch.

A. Preprocessing

The data provided in the challenge is divided into 3 sets:

- A train set of 900 images, where each image contains only one car and one license plate. The annotations provided are the bounding box coordinates (ymin, xmin, ymax, xmax)
- A second train set of 900 images, where each image contains a license plate. The annotations provided are the characters in the license plate.

- A test set of 201 images, similar to the training set, where we have to perform both license plate detection and recognition.
In this project, I will not be using the second train set for recognition.I will be using the Pytesseract module for recognition.

In the preprocessing, we will take each and every image and convert it into an array using OpenCV and resize the image into 224 x 224.
After that, we will normalize the image just by dividing it by the number 255. We also need to normalize our labels too. Because for the deep learning model, the output range should

be between 0 to 1. For normalizing labels, we need to divide the diagonal points by the width and height of the image.

B. Model From Scratch

The model will be using 3 $convolutional$ layers each with $MaxPooling2D$ and 2 fully connected layers.
The final layer is a fully connected layer with 4 dimension vector output, representing the bounding box coordinates.

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 222, 222, 128)	3584
max_pooling2d_1 (MaxPooling 2D)	(None, 111, 111, 128)	0
conv2d_7 (Conv2D)	(None, 109, 109, 64)	73792
max_pooling2d_2 (MaxPooling 2D)	(None, 54, 54, 64)	0
conv2d_8 (Conv2D)	(None, 52, 52, 16)	9232
max_pooling2d_3 (MaxPooling 2D)	(None, 26, 26, 16)	0
flatten_2 (Flatten)	(None, 10816)	0
dense_4 (Dense)	(None, 8)	86536
dense_5 (Dense)	(None, 4)	36

Total params: 173,180
Trainable params: 173,180
Non-trainable params: 0

Fig. 1. Model Architecture

C. Transfert Learning

We will see try two pretrained models: $Resnet50$ and $VGG16$, with 3 additional fully connected layers.

IV. LICENSE PLATE RECOGNITION

In this section, we will explain the techniques used to do the recognition job and talk about one of the most powerful tools for image recognition.

A. Pytesseract

Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and read the text embedded in images. Optical character recognition (OCR) software that is used to extract text from the image.

1) LIMITATIONS OF PYTESSERACT: Tesseract works best when there is a clear segmentation of the foreground text from the background. In practice, it can be extremely challenging to guarantee these types of setups. There are a variety of reasons you might not get good quality output from Tesseract like if the image has noise in the background. The better the image quality (size, contrast, lightning) the better the recognition result. It requires a bit of preprocessing to improve the OCR results, images need to be scaled appropriately, have as much image contrast as possible, and the text must be horizontally aligned.
In addition, Tesseract does not recognize Arabic characters.

2) Image Transformation To recognize the numbers: To overcome the limitations of pytesseract we will go threw the following pipeline.
Transforming the image into gray-blur-binary-dilation.
This pipeline will generate 4 more versions of the original image.
We will apply the pytesseract function on the images and get the common substrings which are the plate numbers.
This method is avoiding the misrecognition of the word $Tunis$ in the car plate which is sometimes recognized as a number or a symbol.

In the following example the word $Tunis$ is recognized as 549 in the first image , the dollar symbol in the gray image, and 0 in the dilation image ... By following this method we will only extract the car plate numbers which will be recognized correctly $159/8894$

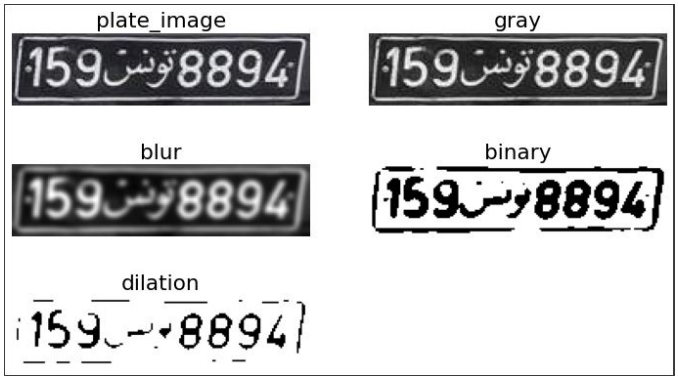

Fig. 2. Image Transformation

V. EXPERIMENTAL RESULTS

For the detection model, we use the Mean squared error as loss function

Models	Train Loss	Validation Loss
Model From Scratch	0.0119	0.0169
ResNet50	0.1552	0.1577
VGG16	4.2336e-04	0.0020

VGG16 was the best-performing model.

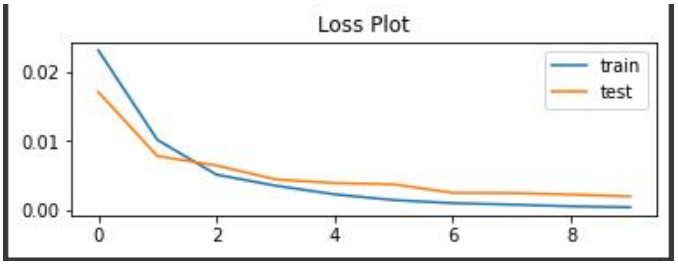

Fig. 3. Loss Function Plot

Here is a result from the model :

Fig. 4. Car Plate Detection Example

VI. Conclusion

In this paper, we demonstrated the use of Convolutional Neural Networks to train a model fit to perform object detection and used the pytesseract module to make the plate recognition.
We used a dataset from a challenge in Zindi.