

Capstone Project

Machine Learning Nanodegree

Abdullah Ahmed Sayed
03/04/2018

Definition

Project Overview:

License Plate Recognition with using the camera's input as video or picture algorithm be able to detect all License in the frame and then extract Plate numbers.

Using Computer vision and convolution Neural network for detecting the license and extract plate numbers.

Problem Statement :

- 1- Scrap web for image of license plate and other images not license plate(people,car Parts,etc.)
- 2- train Sigmoid Classifier if the image contain license plate or not
- 3- use sliding window to extract ROI in input image extract license plate in the image
- 4- create images for Characters and numbers($A \rightarrow Z, 0 \rightarrow 9$) for five different fonts
- 5- create Softmax Classifier for recognize the Character (we have 36 class from $A \rightarrow Z$, $0 \rightarrow 9$)

Intend Solution will be:

Detecting the license plate number from an image

Metric:

I will use accuracy as metrics for my models (License Classifier,digits Recognition)

Accuracy = (True Positive+True Negative)/data Size

Analysis

Data Exploration:

There are two different type of datasets

1- images contain(not) License plate resized to (100x100)

Where contain License plate Labeled = 1 , not contain license Images for people, car parts , animals , and empty streets Labeled = 0

2- images contain different shape of Characters also resized (100x100)

Where each unique character Label as it's own Class



Character



License



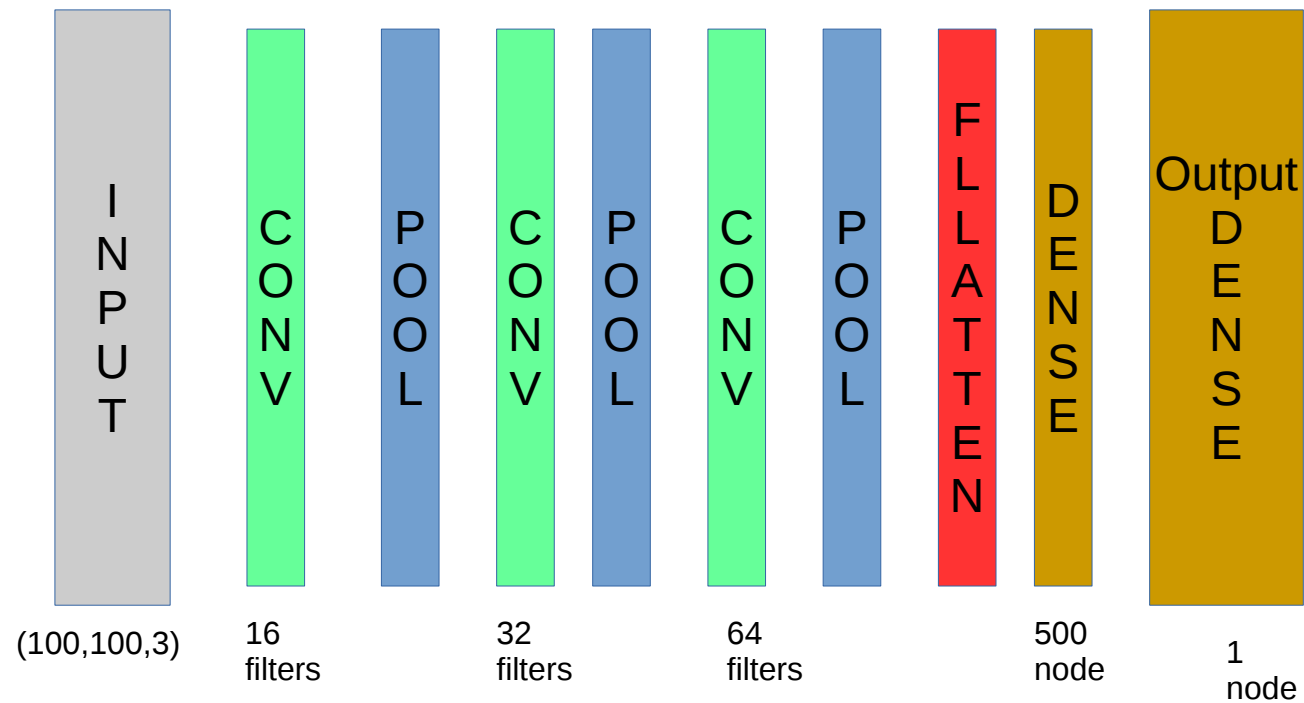
Not License



Input Image

Algorithms and Techniques:

At the beginning After Preprocessing I create a model With CNN to classify license plate form Not license plate,I use Input shape of (100,100,3) create the architecture below , each Convolutional layer consist of number of filters and each filter consist of 3 channels (R,G,B).



After Training the model and saving the best Weights , then used Sliding window technique to detect license plate in the image then extract the license part from the image.

Use pyramid technique for different scale of the input image , specify min size and Scale which specify how many layer of the image.

Also use CNN to detect Character and digits in the license plate

Benchmark:

the model for Classify Character accuracy of 80 %
and the model for detect the license plate in the scene should be with accuracy of 90% or detect 2 more license or 2 less from the scene

Methodology

Data Preprocessing:

- 1-Starting with loading the images
- 2- resize the images into the same size
- 3- adding labels to the images
- 4- randomize the data
- 5- split the data into training , validation , testing

Implementation:

After Loading the data and split it into training ,testing,and validation I create the first model to classify (license or not license) , I build architecture like on the right image with total params = 4millions

After Training the model

Use technique Sliding window and pyramids on the input image :Sliding window for selecting different small parts of the input images ,Pyramid for different scale of the input image to apply sliding window on it.

using tensorflow backend:

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 100, 100, 16)	208
max_pooling2d_1 (MaxPooling2)	(None, 50, 50, 16)	0
conv2d_2 (Conv2D)	(None, 50, 50, 32)	2080
max_pooling2d_2 (MaxPooling2)	(None, 25, 25, 32)	0
conv2d_3 (Conv2D)	(None, 25, 25, 64)	8256
max_pooling2d_3 (MaxPooling2)	(None, 12, 12, 64)	0
dropout_1 (Dropout)	(None, 12, 12, 64)	0
flatten_1 (Flatten)	(None, 9216)	0
dense_1 (Dense)	(None, 500)	4608500
dropout_2 (Dropout)	(None, 500)	0
dense_2 (Dense)	(None, 1)	501
Total params: 4,619,545		
Trainable params: 4,619,545		
Non-trainable params: 0		

Implementation:

second part to build a model to classify image of each character/number to the correspond Label (Character) I have output Softmax with different unique 36 class from A → Z ,0 → 9 where (a,A) in the same class (A) then train the model with architecture in the image

the last step After extract the license plate I need to extract every Character by it self to be run the model to classify which character is. I use opencv function cv2.findContour() to find every character

Refinement:

My initial model for classify license or not license I used opencv image processing library for detecting the license, but I want to do more so I build the model to detect license.

For Character Recognition I start with model consist of 3 layer of Convolution and Max Pooling layer with filters (16,32,64) and Dense layer of size 500 node the test accuracy = 51%

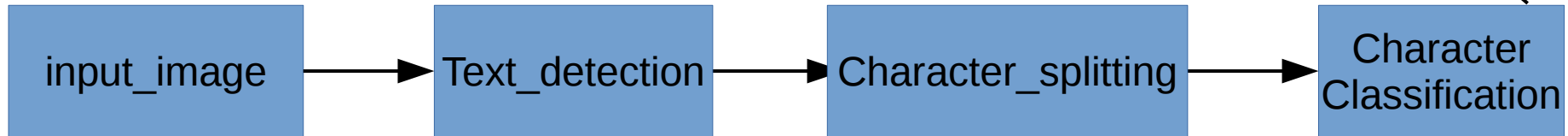
Then adjust the parameter as in the image on the right to get the test accuracy = 70%

The final step instead of create classifier for detect character from white space between the character (because the lack of data set image and low test accuracy) I use opencv library to find contour in the image with minimum number of contour point = 500 and the height of the rectangle = 70

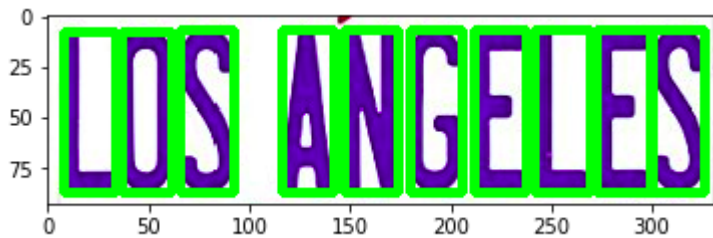
Layer (type)	Output Shape	Param #
conv2d_28 (Conv2D)	(None, 100, 100, 16)	208
max_pooling2d_27 (MaxPooling)	(None, 50, 50, 16)	0
conv2d_29 (Conv2D)	(None, 50, 50, 32)	2080
max_pooling2d_28 (MaxPooling)	(None, 25, 25, 32)	0
conv2d_30 (Conv2D)	(None, 25, 25, 64)	8256
max_pooling2d_29 (MaxPooling)	(None, 12, 12, 64)	0
conv2d_31 (Conv2D)	(None, 12, 12, 128)	32896
max_pooling2d_30 (MaxPooling)	(None, 6, 6, 128)	0
dropout_29 (Dropout)	(None, 6, 6, 128)	0
flatten_8 (Flatten)	(None, 4608)	0
dense_29 (Dense)	(None, 1000)	4609000
dropout_30 (Dropout)	(None, 1000)	0
dense_30 (Dense)	(None, 1200)	1201200
dropout_31 (Dropout)	(None, 1200)	0
dense_31 (Dense)	(None, 1000)	1201000
dropout_32 (Dropout)	(None, 1000)	0
dense_32 (Dense)	(None, 36)	36036
Total params: 7,090,676		
Trainable params: 7,090,676		
Non-trainable params: 0		

Conclusion

Free-Form Visualization:



Input Image



q (q)	u (U)	r (r)	8 (8)	h (h)	f (T)	10 (10)	Y (Y)
q	U	r	8	h	T	10	Y
s (s)	w (W)	c (c)	B (H)	7 (7)	B (E)	o (o)	A (A)
S	W	C	H	7	E	O	A
c (G)	i (j)	i (i)	P (P)	V (V)	x (X)	r (K)	t (t)
G	J	l	P	V	X	K	t
3 (3)	e (e)	g (0)	5 (5)	g (g)	t (L)	B (B)	B (O)
3	e	0	5	g	L	B	0
B (D)	B (F)	z (z)	q (Q)	x (x)	B (a)	4 (4)	y (y)
D	F	Z	Q	X	a	4	y
u (u)	V (v)	b (b)	v (M)	f (f)	p (p)	j (j)	6 (6)
u	V	b	M	f	p	j	6
2 (2)	w (w)	n (n)	s (S)	m (m)	d (d)	k (k)	i (i)
2	w	n	s	m	d	k	i
1 (1)	c (C)	x (N)	9 (9)	z (Z)	R (R)	i (i)	
2	w	n	s	m	d	k	i
1	C	N	9	Z	R	I	

Predict Character

Reflection:

summarized step:

- 1- gather/create the data set
- 2- add label to the data set
- 3- split the data into train validation and testing
- 4- build the model
- 5- train the model
- 6- sliding window and Pyramid
- 7- ROI and extract character
- 8- predict license label

I found Sliding window interesting part

Improvement:

- * instead of using CNN for detect License plate we can use opencv to detect license plate in the image this will give better result
- * use word correction for better result in license Plate (instead Ca1ifornia → California)
- * gather more image dataset for better accuracy result.
- * apply distribution in the create data image (artificial data synthesis) for more images