

RESEARCH AND IMPLEMENTATION OF AUTOMATIC IMAGE LABELING ALGORITHMS

< ORA PROJECT _ IT-ENGLISH PROJECT >



TOP TECHNOLOGY TRENDS FOR 2020

NLP (Natural Language Processing)

Conversational AI

CV (Computer Vision)

RL (Reinforcement Learning)

PROJECT DESCRIPTION AND GOALS

- A continuation of exercise 6 of TP3 image analysis
- Study and implement (in *python + opencv*) a software prototype associating keywords with images or photos
- The keywords must describe the content (*color, content, quality, etc.*) of the images.
- Generate a new tags *txt* file
- Compare two tags *txt* files and Assess the accuracy of detection

ENVIRONMENT SETUP

- System (*Windows 10*), Graphics driver (*NVIDIA GeForce 930MX , Inter® HD Graphics 620*), Processor (*Inter® Core™ i5-7200U CPU @ 2.50GHz 2.71 GHz*)
- *OpenCV*
- *Python*
- Other necessary packages

OBTAIN THE MAIN COLORS OF THE IMAGE

Method:

1. Convert the image color mode from RGB to HSV
2. Use cv2.inRange() function for background color filtering
3. Binarize the filtered colors
4. Use cv2.dilate() to make dilation and erosion
5. Statistics white area

	black	gray	white	red		orange	yellow	green	cyan	blue	purple
hmin	0	0	0	0	156	11	26	35	78	100	125
hmax	180	180	180	10	180	25	34	77	99	124	155
smin	0	0	0	43		43	43	43	43	43	43
smax	255	43	30	255		255	255	255	255	255	255
vmin	0	46	221	46		46	46	46	46	46	46
vmax	46	220	255	255		255	255	255	255	255	255

Note: H: 0 - 180, S: 0 - 255, V: 0 - 255

OBTAIN THE SCENES OF THE IMAGE

Method:


- Call the pre-trained model *PlacesCNN* trained on dataset *Places365*.
- Label the scene attributes which is from dataset *SUNattribute*.
- Generate *Heatmap* with the *CAMtechnique*.



Fig. 1. Image samples from various categories of the Places Database (two samples per category). The dataset contains three macro-classes: Indoor, Nature, and Urban.

Places Database is a repository of 10 million scene photographs, labeled with scene semantic categories.

OBTAIN THE MAIN OBJECTS OF THE IMAGE

1. Call *Mask_Rcnn - Tensorflow* pre-trained model trained on the dataset *MSCOCO*
2. Download the models
3. Initialize the parameters
 - File: *mscoco_labels.names* - objects labels
 - File: *colors.txt* - mask colors
4. Load the model and classes 
 - File: *frozen_inference_graph.pb* - network ==> the pre-trained weights
 - File: *mask_rcnn_inception_v2_coco_2018_01_28.pbtxt* - The text graph file to make network to be loaded by using OpenCV
5. Read the input
6. Process each frame
7. Post-processing the network's output
8. Draw the predicted boxes

DETECT THE SHARPNESS AND THE BRIGHTNESS OF THE IMAGE

SHARPNESS

Method:

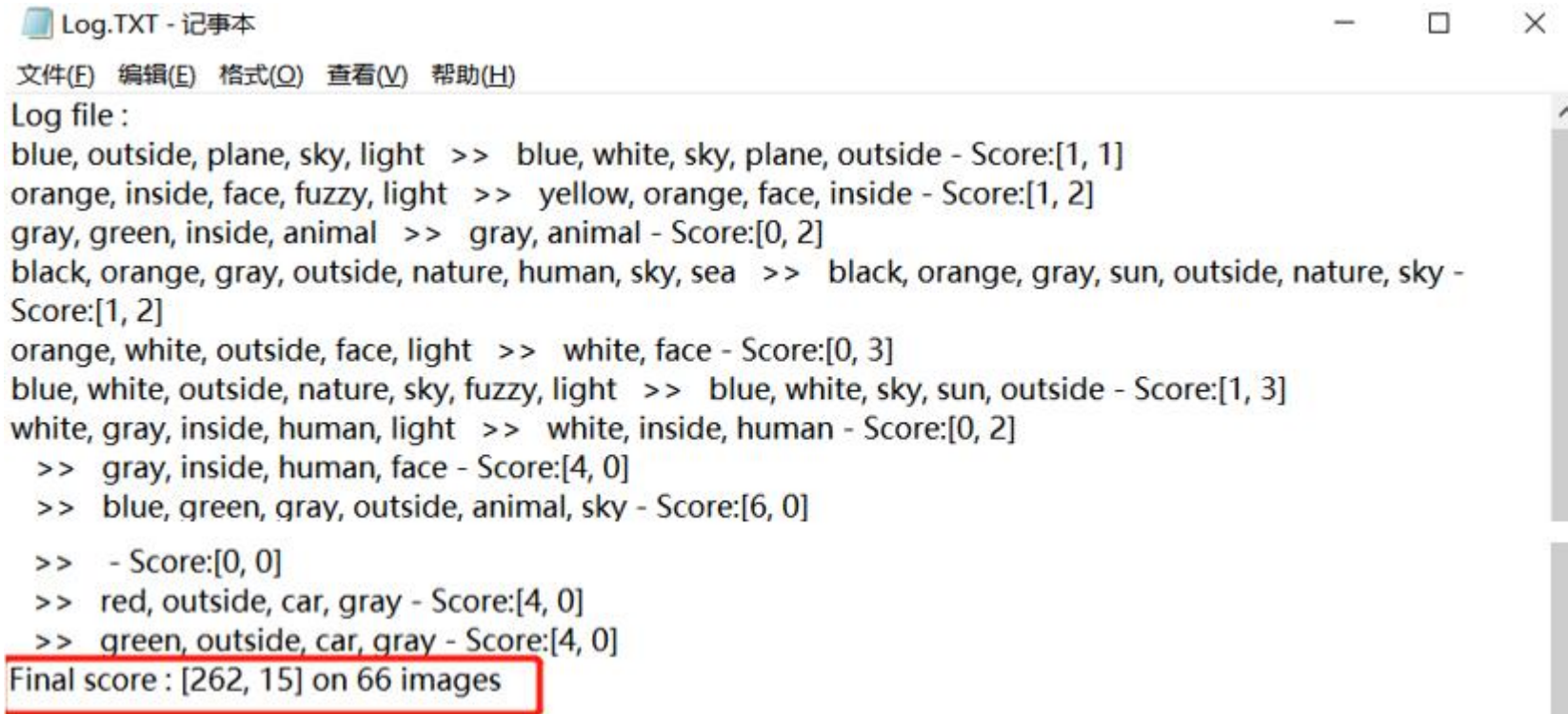
1. Make a *convolution* operation on a channel in the image through a *Laplace mask* [*edge detection*]
2. Then calculate the standard deviation / variance,
3. If the image has a higher variance, then it has a wider frequency response range, representing a normal, accurately focused image, the image is very clear.
4. Set the appropriate threshold

BRIGHTNESS

Method:

1. Obtain the *grayscale histogram* of the image → Judge according to the distribution of grayscale values
2. Dark image gray value: concentrated in the front, such as around 0-30
3. Bright image gray value: concentrated in the back
4. Need to count the number of dark pixels, and then divide by the total number of image pixels to get the percentage p
5. Set the threshold of p

COMPARE TWO TAGS TXT FILES AND ASSESS THE ACCURACY OF DETECTION



```
Log.TXT - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)
Log file :
blue, outside, plane, sky, light >> blue, white, sky, plane, outside - Score:[1, 1]
orange, inside, face, fuzzy, light >> yellow, orange, face, inside - Score:[1, 2]
gray, green, inside, animal >> gray, animal - Score:[0, 2]
black, orange, gray, outside, nature, human, sky, sea >> black, orange, gray, sun, outside, nature, sky -
Score:[1, 2]
orange, white, outside, face, light >> white, face - Score:[0, 3]
blue, white, outside, nature, sky, fuzzy, light >> blue, white, sky, sun, outside - Score:[1, 3]
white, gray, inside, human, light >> white, inside, human - Score:[0, 2]
>> gray, inside, human, face - Score:[4, 0]
>> blue, green, gray, outside, animal, sky - Score:[6, 0]

>> - Score:[0, 0]
>> red, outside, car, gray - Score:[4, 0]
>> green, outside, car, gray - Score:[4, 0]
Final score : [262, 15] on 66 images
```

*** OPERATION DEMO ***

THANKS FOR YOUR ATTENTION

< ORA PROJECT _ IT-ENGLISH PROJECT >

