

High level Content-based image retrieval

Silvério Pereira, 76505

Pedro Santos, 76532



Introduction

Development of an application to do image retrieval by extracting high level information from the images.

The application is divided in two parts:

- Indexing set of images
- Search by image /test to get the results
- Rank the results



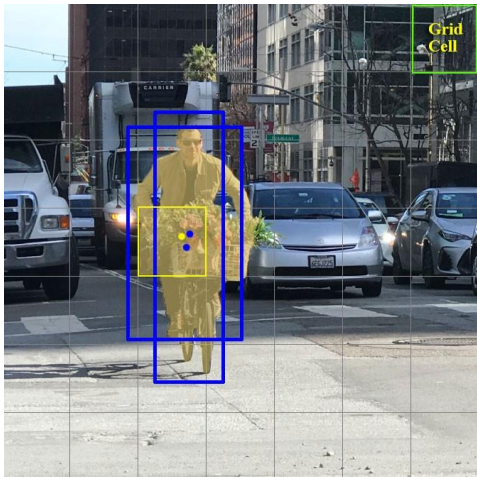
What do we use to score images?

- Object detection
- Visual Saliency
- Face recognition
- Object Character recognition

Yolov3

YOLO (You Only Look Once) is a state of the art object recognition algorithm.

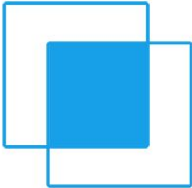

- Divides the image in grid cells and applies a CNN classifier to each, each grid cell has a maximum of objects it can predict.

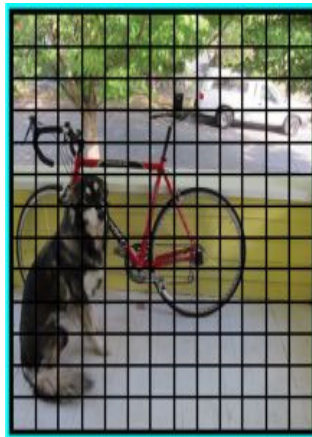


$$\hat{y} = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_y \\ c_1 \\ c_2 \\ \dots \\ c_n \\ P_c \\ b_x \\ b_y \\ b_y \\ c_1 \\ c_2 \\ \dots \\ c_n \end{bmatrix}$$

1 anchor box

Non-max suppression


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


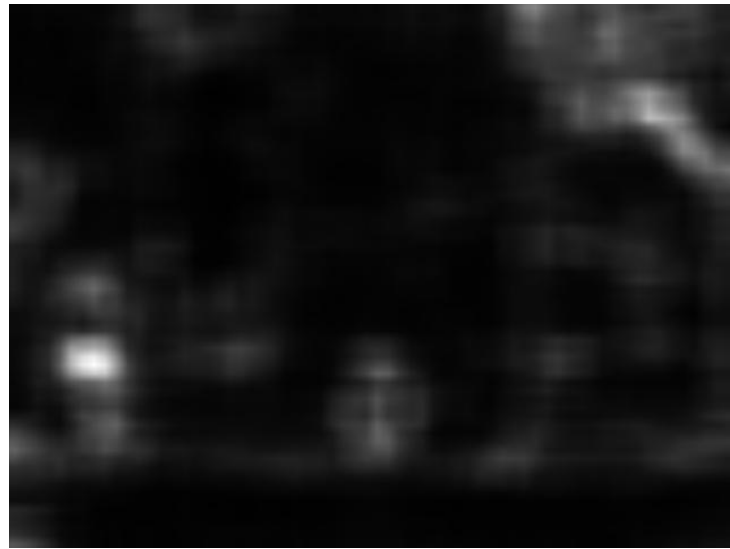
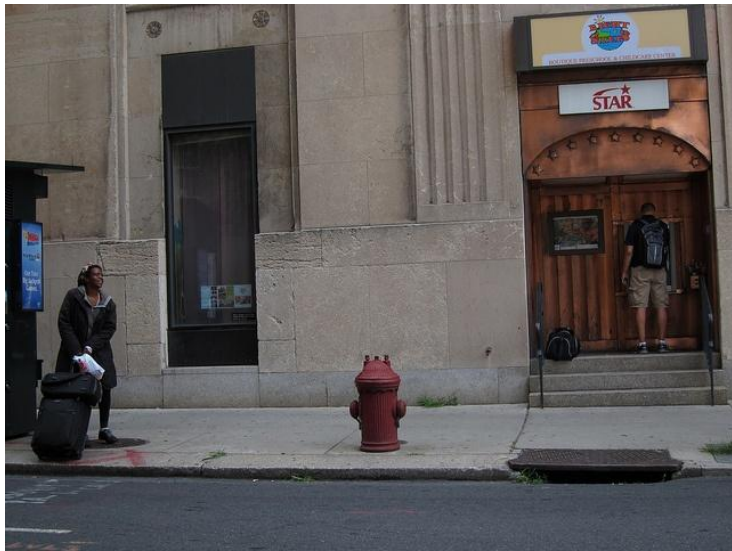




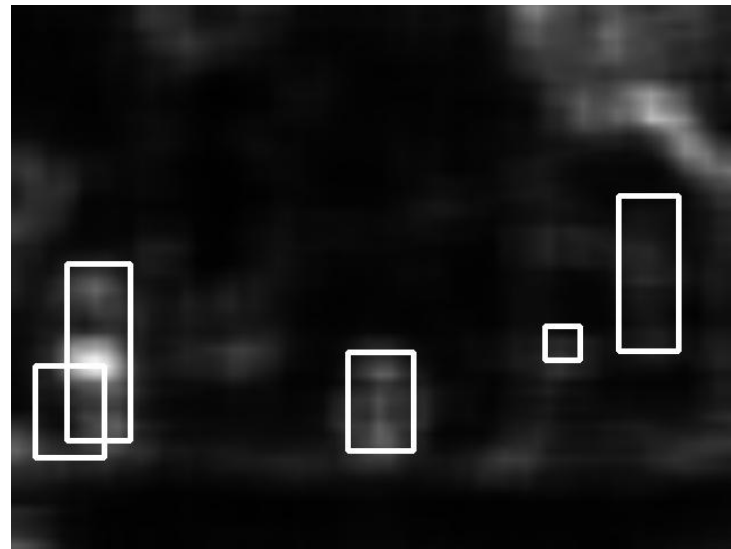
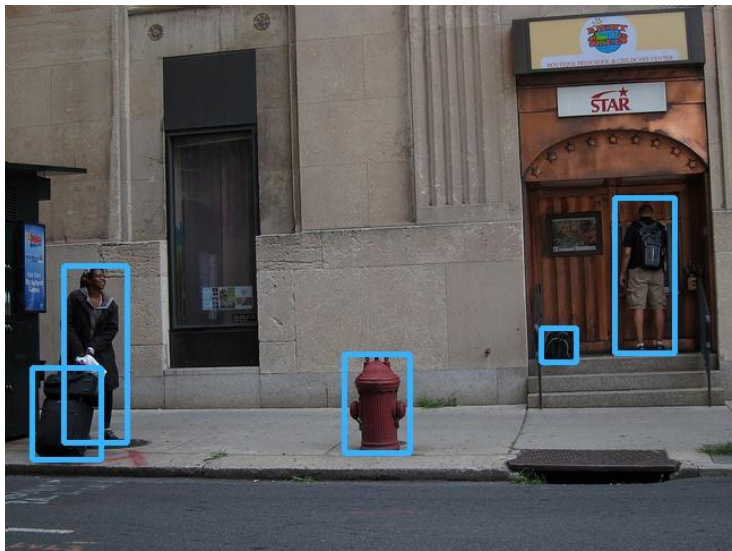
Tunable parameters

- Confidence threshold
- Non-max suppression threshold (IoU)
- Max number of objects in a cell grid
- Size of images

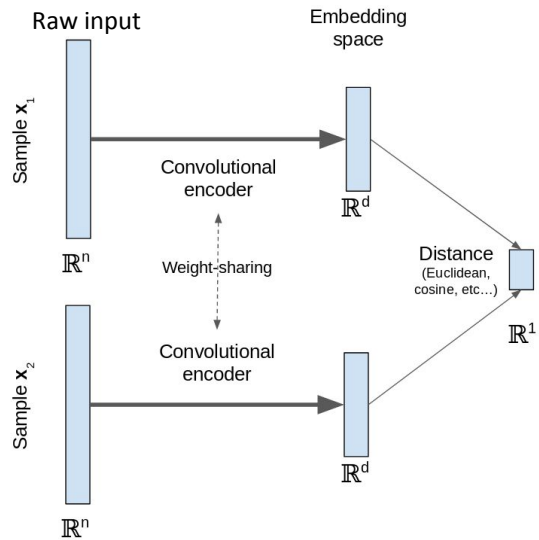
Visual Saliency



Relevance Calculation

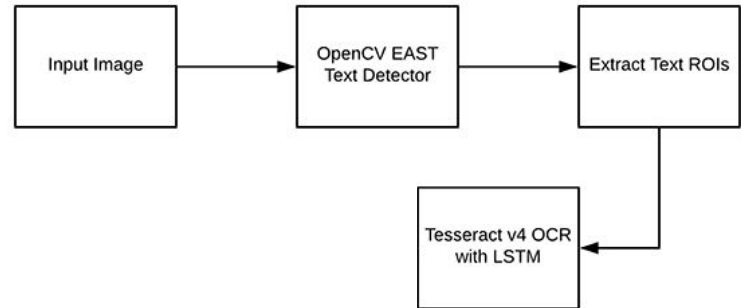


One Shoot Face Recognition



OCR (optical character recognition)

- The function `verifyText()` is called and it returns a list with the words detected in the given image;
- Verification if it was detected any text in the image;
- Removing Special Characters;
- Performing Stemming;





Transfer Learning

- It is not easy... it is necessary to have some installed dependencies and to do and manipulate configurations files.
- To be able to see the results of the configuration of the network is necessary to wait several hours.
- Collect the labeled data
- The Darknet implementation is used to train the data using the GPU
- It is necessary to have two files, one with the paths to the images of the training set and another to the test set.



Transfer Learning

- Some lines in file detector.c were changed to save the intermediate weights more frequently.
- Download the pre-trained model from darknet
- Create some files to configure Darknet and set the architecture of the network

```
rute@rute-X550JK:~/Documents/cadeiras/vc/progs/darknet$ ./darknet detector train  
obj.data cfg/yolov3-tiny.cfg backup/yolov3-tiny_5000.weights > /home/rute/Documents/cadeiras/vc/vc1819-76505-76532/project/proj1/training/log.txt
```

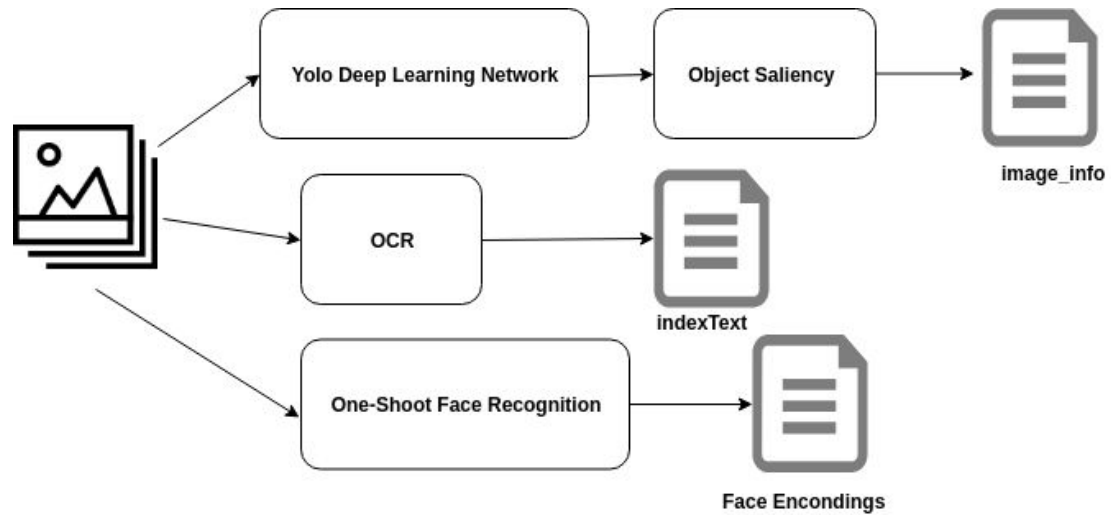
Transfer Learning

```
rute@rute-X550JK:~/Documents/cadeiras/vc/progs/darknet$ ./darknet detector test obj.data cfg/yolov3-tiny.cfg backup/yolov3-tiny_5000.weights data/tree1.jpg
layer  filters  size  input  output
0 conv    16  3 x 3 / 1  416 x 416 x 3  -> 416 x 416 x 16  0.150 BFLOPs
1 max      2  2 x 2 / 2  416 x 416 x 16  -> 208 x 208 x 16
2 conv    32  3 x 3 / 1  208 x 208 x 16  -> 208 x 208 x 32  0.399 BFLOPs
3 max      2  2 x 2 / 2  208 x 208 x 32  -> 104 x 104 x 32
4 conv    64  3 x 3 / 1  104 x 104 x 32  -> 104 x 104 x 64  0.399 BFLOPs
5 max      2  2 x 2 / 2  104 x 104 x 64  -> 52 x 52 x 64
6 conv   128  3 x 3 / 1   52 x 52 x 64  -> 52 x 52 x 128  0.399 BFLOPs
7 max      2  2 x 2 / 2   52 x 52 x 128  -> 26 x 26 x 128
8 conv   256  3 x 3 / 1   26 x 26 x 128  -> 26 x 26 x 256  0.399 BFLOPs
9 max      2  2 x 2 / 2   26 x 26 x 256  -> 13 x 13 x 256
10 conv  512  3 x 3 / 1   13 x 13 x 256  -> 13 x 13 x 512  0.399 BFLOPs
11 max      2  2 x 2 / 1   13 x 13 x 512  -> 13 x 13 x 512
12 conv  1024  3 x 3 / 1   13 x 13 x 512  -> 13 x 13 x 1024 1.595 BFLOPs
13 conv   256  1 x 1 / 1   13 x 13 x 1024 -> 13 x 13 x 256  0.089 BFLOPs
14 conv   512  3 x 3 / 1   13 x 13 x 256  -> 13 x 13 x 512  0.399 BFLOPs
15 conv   18  1 x 1 / 1   13 x 13 x 512  -> 13 x 13 x 18  0.003 BFLOPs
16 yolo
17 route   13
18 conv   128  1 x 1 / 1   13 x 13 x 256  -> 13 x 13 x 128  0.011 BFLOPs
19 upsample    2x 13 x 13 x 128  -> 26 x 26 x 128
20 route   19 8
21 conv   256  3 x 3 / 1   26 x 26 x 384  -> 26 x 26 x 256  1.196 BFLOPs
22 conv   18  1 x 1 / 1   26 x 26 x 256  -> 26 x 26 x 18  0.006 BFLOPs
23 yolo
Loading weights from backup/yolov3-tiny_5000.weights...Done!
data/tree1.jpg: Predicted in 0.027182 seconds.
tree: 93%
tree: 90%
tree: 80%
tree: 51%
rute@rute-X550JK:~/Documents/cadeiras/vc/progs/darknet$
```

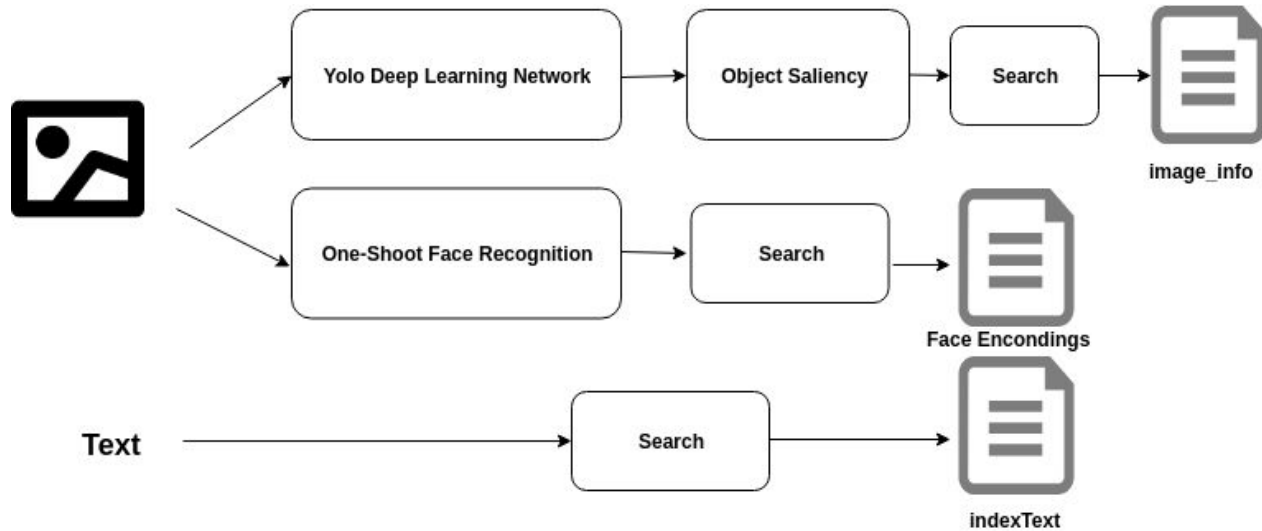
Transfer Learning



Indexing Data flow



Search Data flow





Ranking

TABLE I
OBJECT WEIGHT CALCULATION

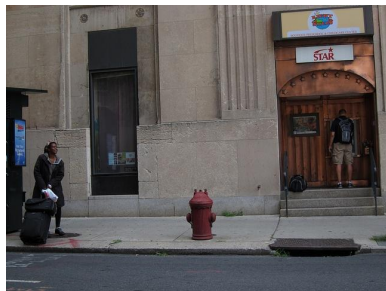
y	\hat{y}	cost
0	0	0
0	> 1	$\log(1 + \hat{y}) * e^{rel(\hat{y})}$
> 1	0	$1 + \log(y) * e^{rel(y)}$
> 1	> 1	$ \log(y) - \log(\hat{y}) * e^{ rel(y) - rel(\hat{y}) }$

TABLE II
FACE DISCOUNT CALCULATION

$d \leq 6$	-10
$0.6 < d < 1$	$\min(\log((d - 0.6) * 2.5)), -10)$
$d > 1$	0

Results

Query



Results

Query: "nokia" →



Query

