Dog muzzle detection based on a cascade of Haar-like classifier

Students: Jerry Wang and Rebecca Liu
Supervisor: Dr. Andre Barczak

17/May/2018

Motivation

- Currently no trained dog detection classifier xml file available for others on Github

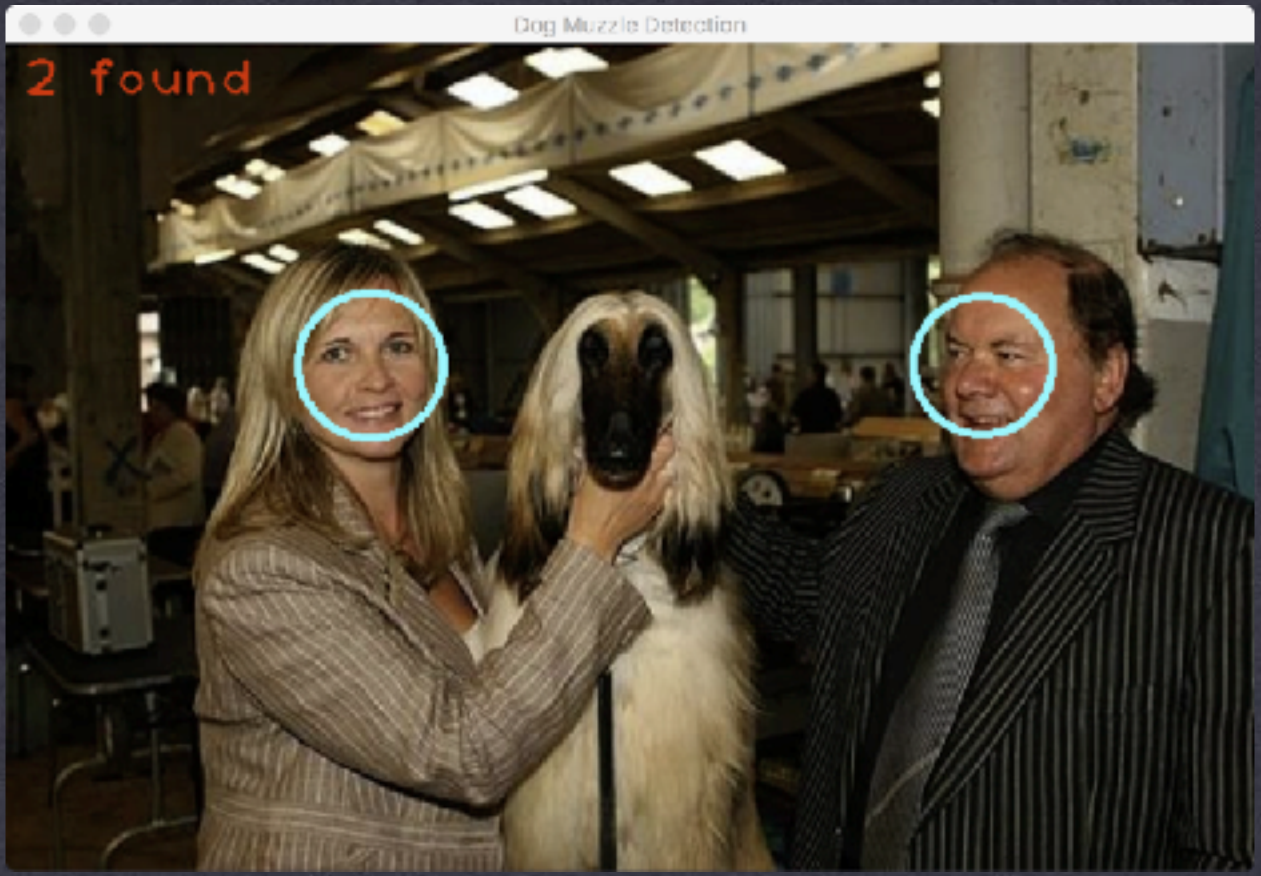
- To train a classifier that can detect dogs with different poses and expressions: frontal, profile and rotated faces

Outline

- Introduction
- Background
- Haar Classifier
- Methodology
- Testing & expected results
- Related works we have done

Introduction

- Tracking dog muzzle area——our project
- Face detection——hottest topic in resent years

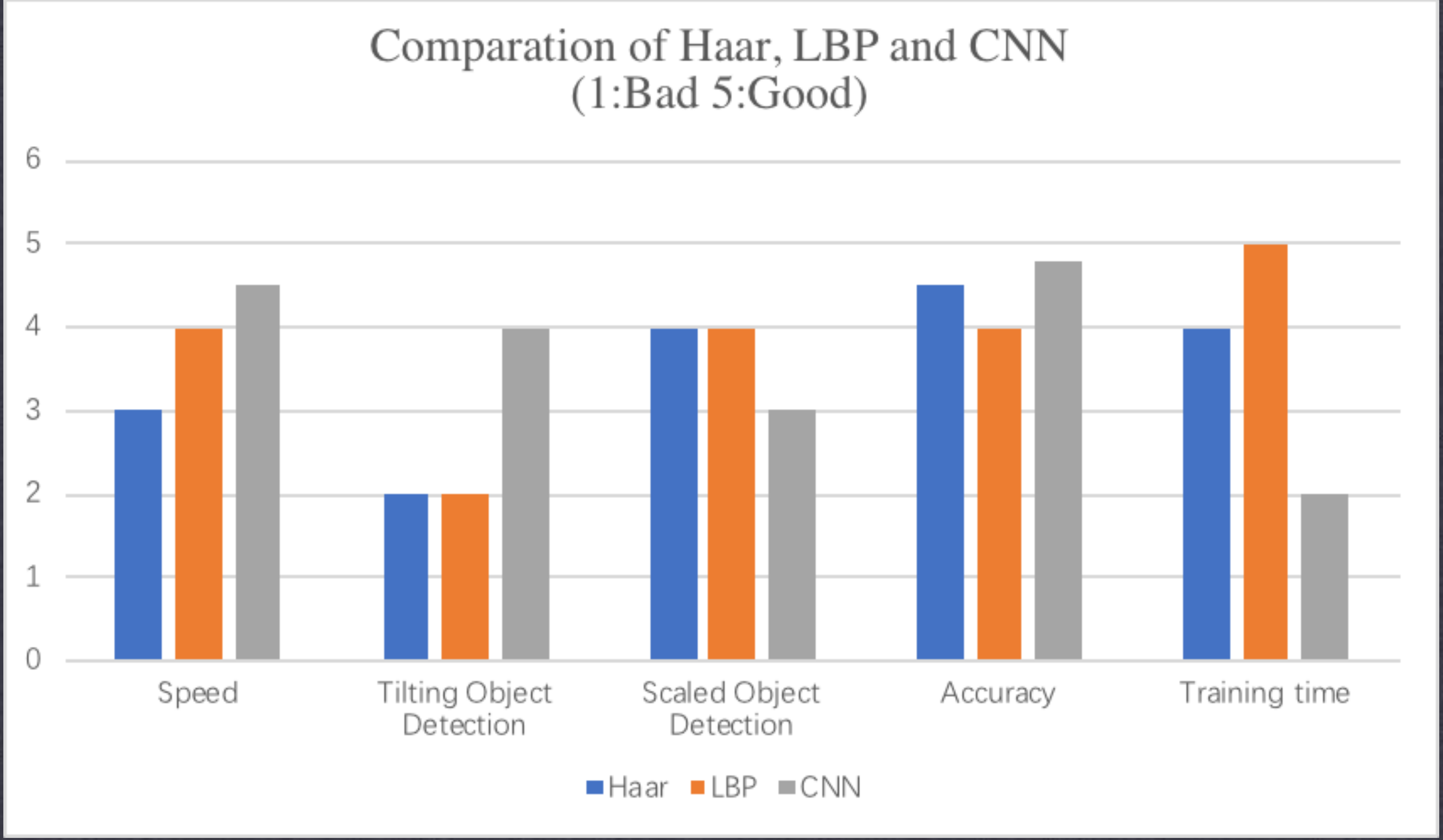


Background

- Comparison of three object detection classifiers
  - Haar
  - LBP
  - CNN

(LBP: Local Binary Pattern

CNN: Convolutional Neural Networks)

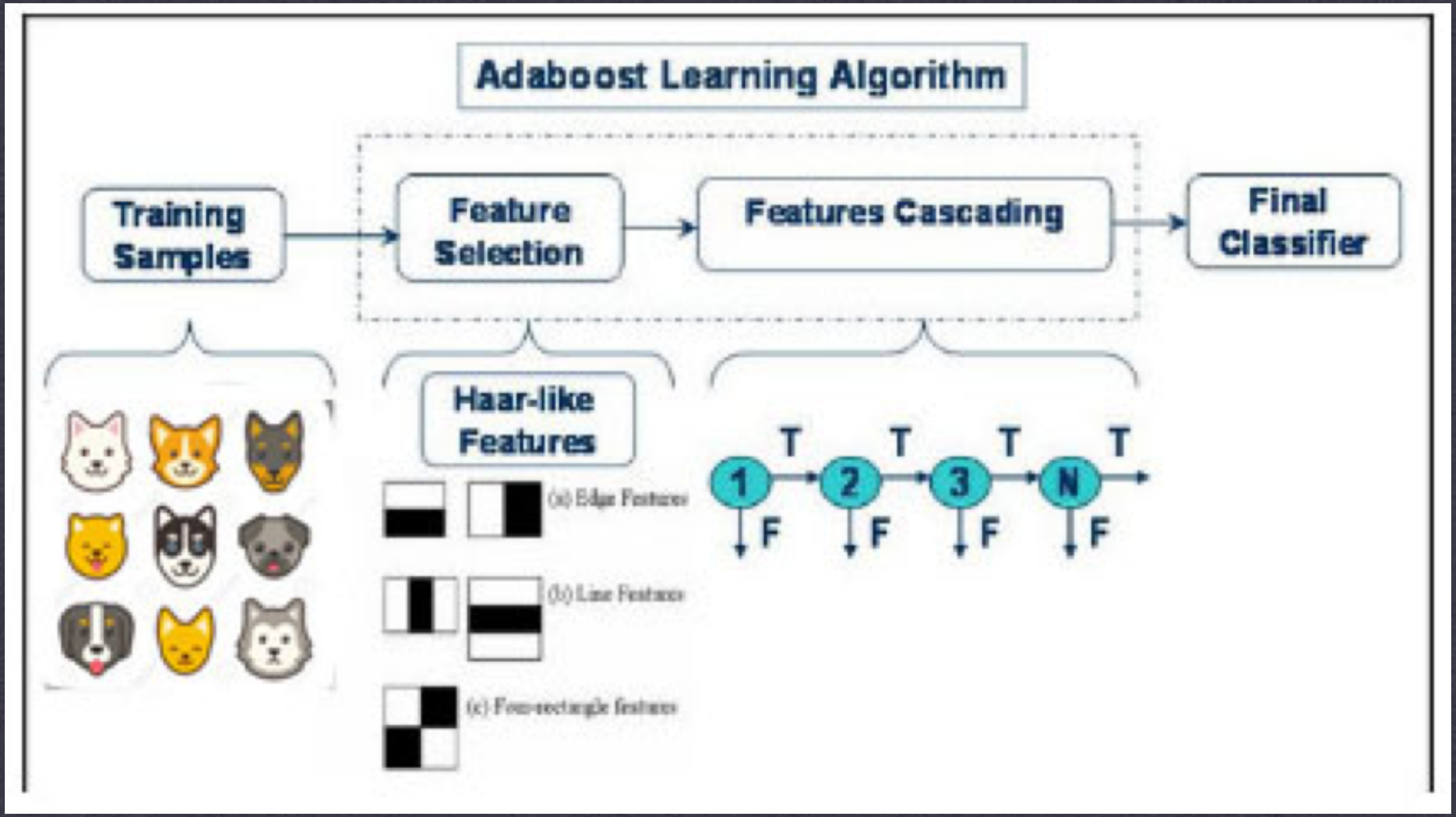


- Why haar? Has a better accuracy than LBP

  Although slightly slower and need more training time

Haar Classifier

The **Haar Classifier** is a machine learning based approach, an algorithm created by Paul Viola and Michael Jones, which is trained from positive images (with objects) and negatives images (without objects).

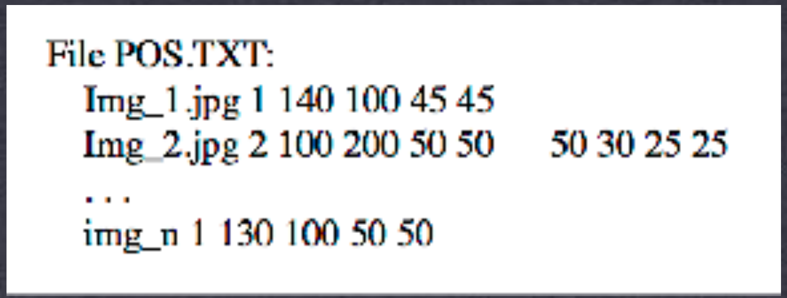


Methodology

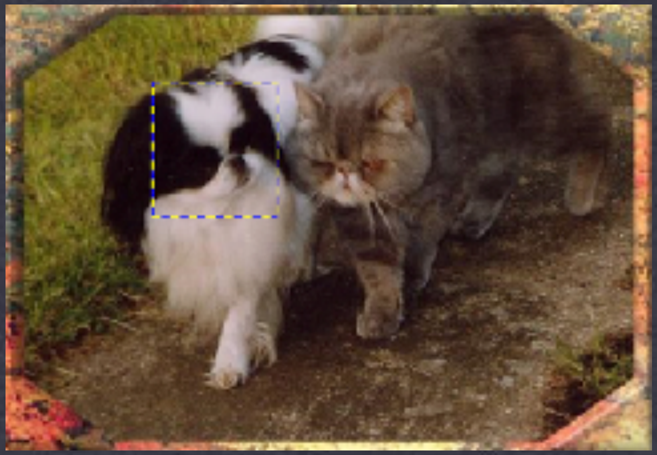
- Step 1: Collecting dog muzzle samples (positive images) and preprocessing
- Step 2: Collecting negative samples
- Step 3: Training Haar classifier
- Step 4: Coding
- Step 5: Testing

Methodology

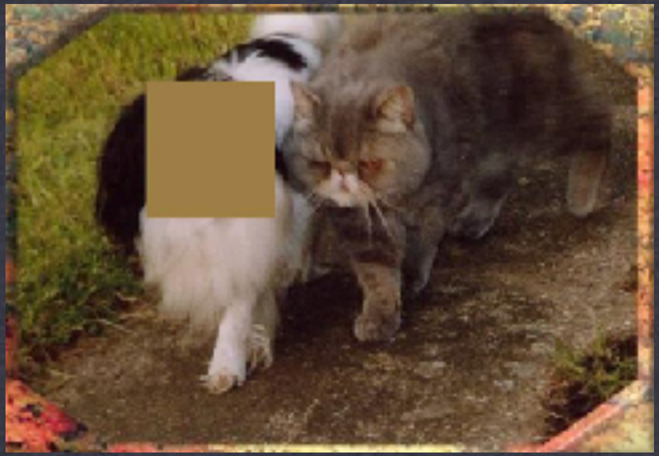
- Step 1: Collecting dog muzzle samples (positive images)-2000
  - Stanford Dogs Dataset:
    select 2000 images randomly
  - Outputting POS.TXT:



```
File POS.TXT:
   Img_1.jpg 1 140 100 45 45
   Img_2.jpg 2 100 200 50 50     50 30 25 25
   . . .
   img_n 1 130 100 50 50
```



- Step 2: Collect negative samples-4000
  - 2000 images from step1
  - 2000 images from a video without dogs



Methodology

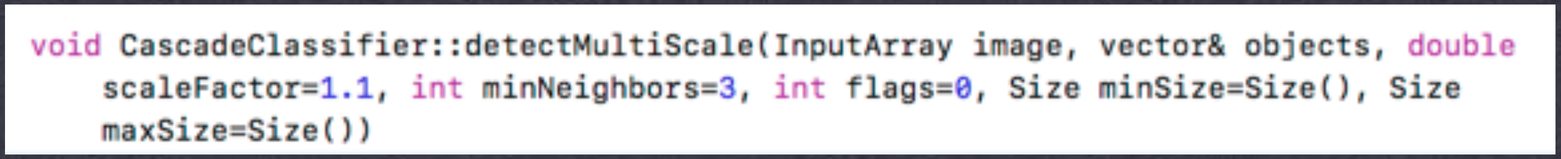
- Step 3: Training Haar classifier
  - Build a vector output file of the positive samples:

e:\ cascade\opencv_createsamples.exe -vec POS.VEC -info positives\POS.TXT -bg negatives\NEG.TXT -w 40 -h 40 -num 2000

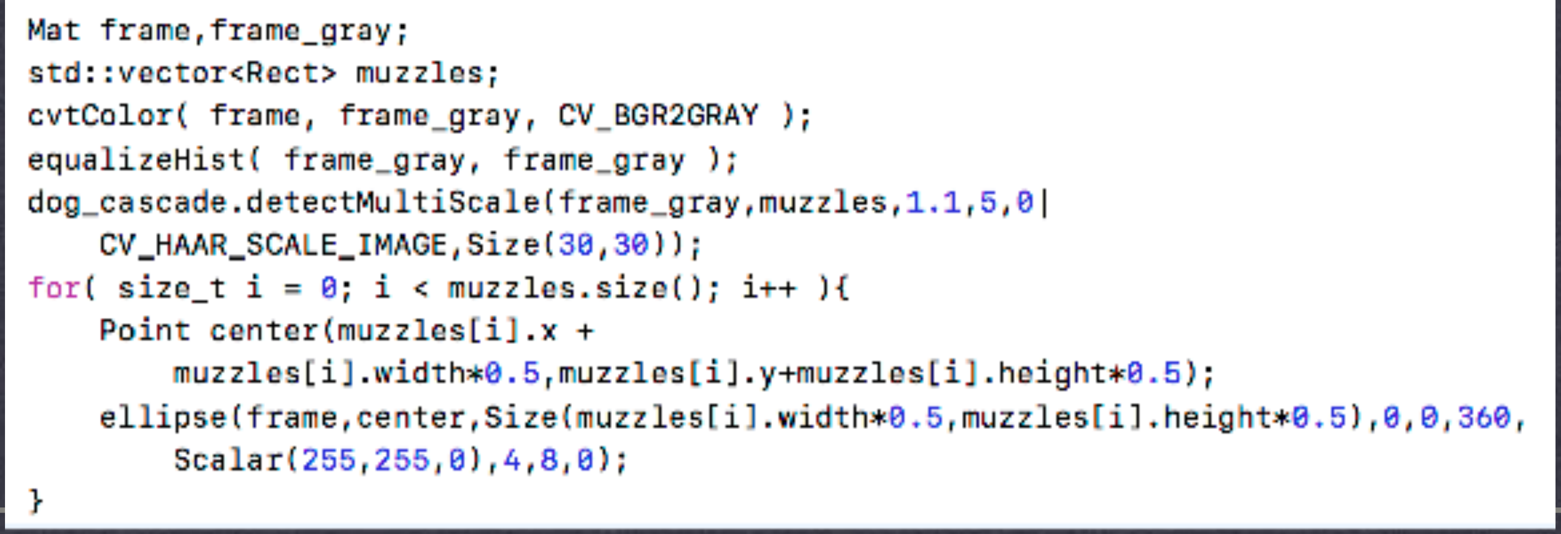
  - Create HAARCASCADE_DOG.XML file:

e:\cascade\opencv_traincascade.exe -data XML -vec POS.VEC -bg NEG.TXT -numPos 2000 -numNeg 4000 -numStages 12 -precalcValBufSize 1024 -precalcIdBufSize 1024 -w 40 -h 40 -mem 1024 -nonsym -mode ALL -maxFalseAlarmRate 0.5 -featureType HAAR

- Step 4: Coding



```
void CascadeClassifier::detectMultiScale(InputArray image, vector& objects, double
    scaleFactor=1.1, int minNeighbors=3, int flags=0, Size minSize=Size(), Size
    maxSize=Size())
```



```
Mat frame,frame_gray;
std::vector<Rect> muzzles;
cvtColor( frame, frame_gray, CV_BGR2GRAY );
equalizeHist( frame_gray, frame_gray );
dog_cascade.detectMultiScale(frame_gray,muzzles,1.1,5,0|
    CV_HAAR_SCALE_IMAGE,Size(30,30));
for( size_t i = 0; i < muzzles.size(); i++ ){
    Point center(muzzles[i].x +
        muzzles[i].width*0.5,muzzles[i].y+muzzles[i].height*0.5);
    ellipse(frame,center,Size(muzzles[i].width*0.5,muzzles[i].height*0.5),0,0,360,
        Scalar(255,255,0),4,8,0);
}
```

Testing & expected result

- Step 5: Testing
  - Experiment 1: Test on training dataset
  - Experiment 2: Test on testing dataset
  - Experiment 3: Test on video file or camera
- Results
  - 1: 100% accuracy on training dataset
  - 2: Goal above 90% accuracy
  - 3: fast enough with acceptable accuracy

Work we have done

- We have downloaded the Stanford Dog Datasets which contains of 20580 dog images, from where our 2000 positive images were chosen randomly.
- We are in the process of object annotation which is used for creating the positive file (POS.TXT) and VEC file.
- We have a code draft which works well on human face detection. When we finish training our classifier, it can work on dog detection.

References

1. Bradski G. and Kaehler A. (2008) Learning OpenCV. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

2. Khosla Aditya, Nityananda Jayadevaprakash, Bangpeng Yao & Li Fei-Fei. (2011) Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Website:http://vision.stanford.edu/aditya86/ImageNetDogs/main.html. Retrieved 18:00pm, April 19,2018.

3. OpenCV Documentation. (2018). Cascade Classifier. Website: https://docs.opencv.org/2.4/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html. Retrieved 18:45, April 19, 2018.

4. OpenCV Documentation. (2018). Cascade Classifier Training. Website: https://docs.opencv.org/3.1.0/dc/d88/tutorial_traincascade.html. Retrieved 18:50pm, April 19, 2018.

5. OpenCV Documentation. (2018). Face Detection Using Haar Cascades. Website: https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html. Retrieved at18:46pm, April 19,2018.

6. Ozhiganov I. (2017). Convolutional Neural Networks vs. Cascade Classifiers for Object Detection. Big Data Zone. Website: https://dzone.com/articles/cnn-vs-cascade-classifiers-for-object-detection. Retrieved 18:25pm, April 19,2018.

7. The CIFAR-10 dataset. (2018). Website: https://www.cs.toronto.edu/~kriz/cifar.html

8. Wikipedia. (2018). Website: https://en.wikipedia.org/wiki/Machine_learning. Retrieved at 19:57pm, 01/May,2018.

9. Wikipedia. (2018). List of datasets for machine learning. Website: https://en.wikipedia.org/wiki/List_of_datasets_for_machine_learning_research. Retrieved at 20:12pm, 01/May,2018.

Thank you