# CSIT 5410 (Spring 2020)

Assignment 4

# Overview

- 1 Written question + 1 Programming question
- Written question: **PCA**
- Programming question: **Object detection with Adaboost**

# Written assignment specifics

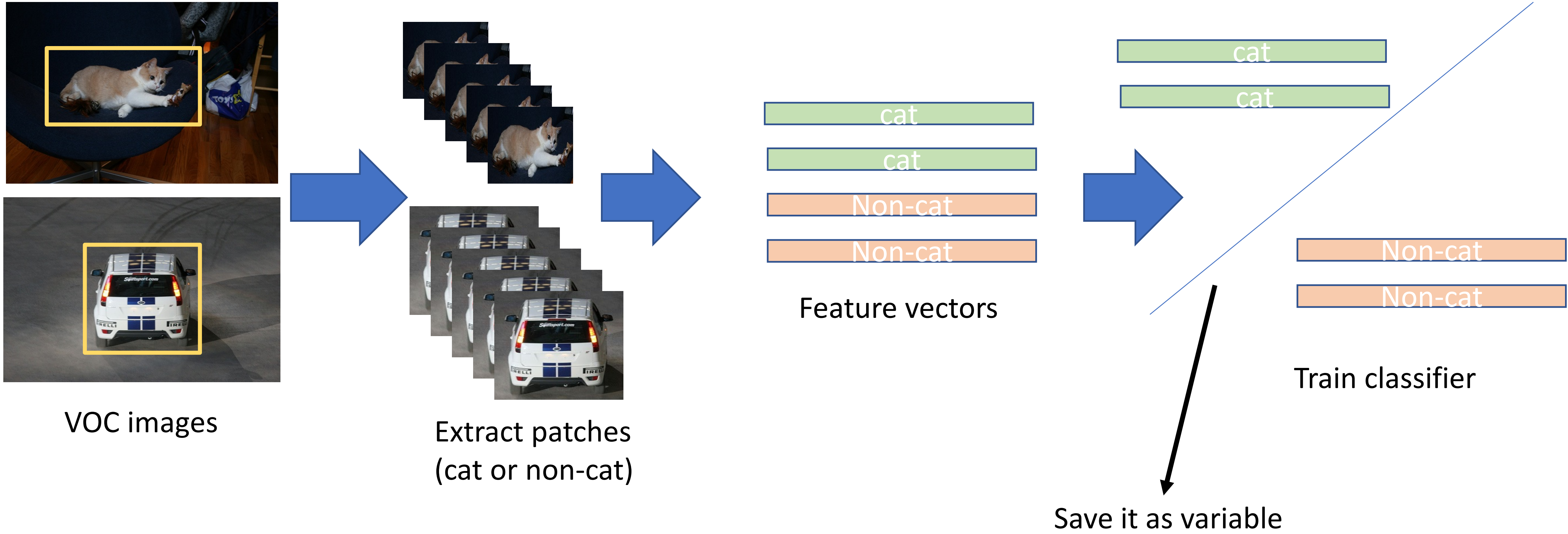
- PCA

1. Compute the scatter matrix
2. Compute the projection for each $x_i$
3. Compute the distance between $\{x_1', x_2', x_3'\}$ and $x_4'$
4. Identify the closest labeled face (The nearest neighbor)
5. **Show all steps**

# Programming assignment specifics

- Task 1 – Feature extraction (LBP/Harr-like feature)
- Task 2 – Weak classifiers
- Task 3 – Adaboost algorithm and sliding window
- Task 4 – Report

# Overview (Training a classifier)



VOC images

Extract patches
(cat or non-cat)

cat

cat

Non-cat

Non-cat

Feature vectors

cat

cat

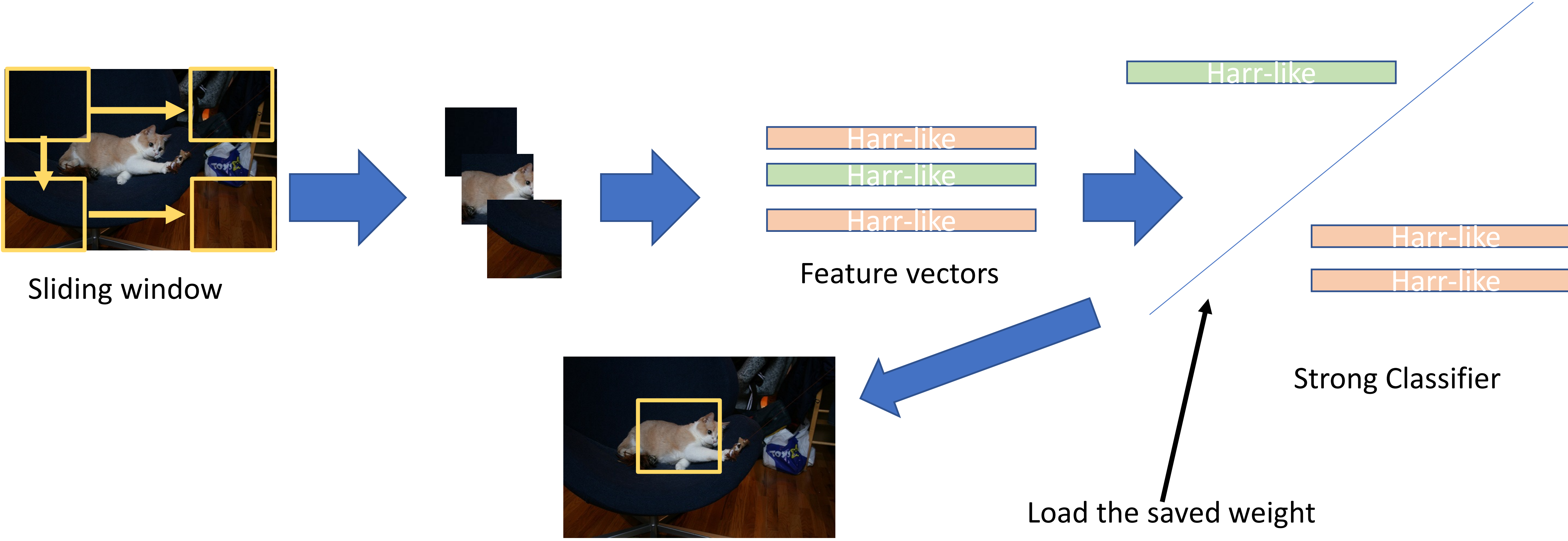
Non-cat

Non-cat

Train classifier

Save it as variable

# Overview (Detection)



Sliding window

Harr-like

Harr-like

Harr-like

Feature vectors

Harr-like

Harr-like

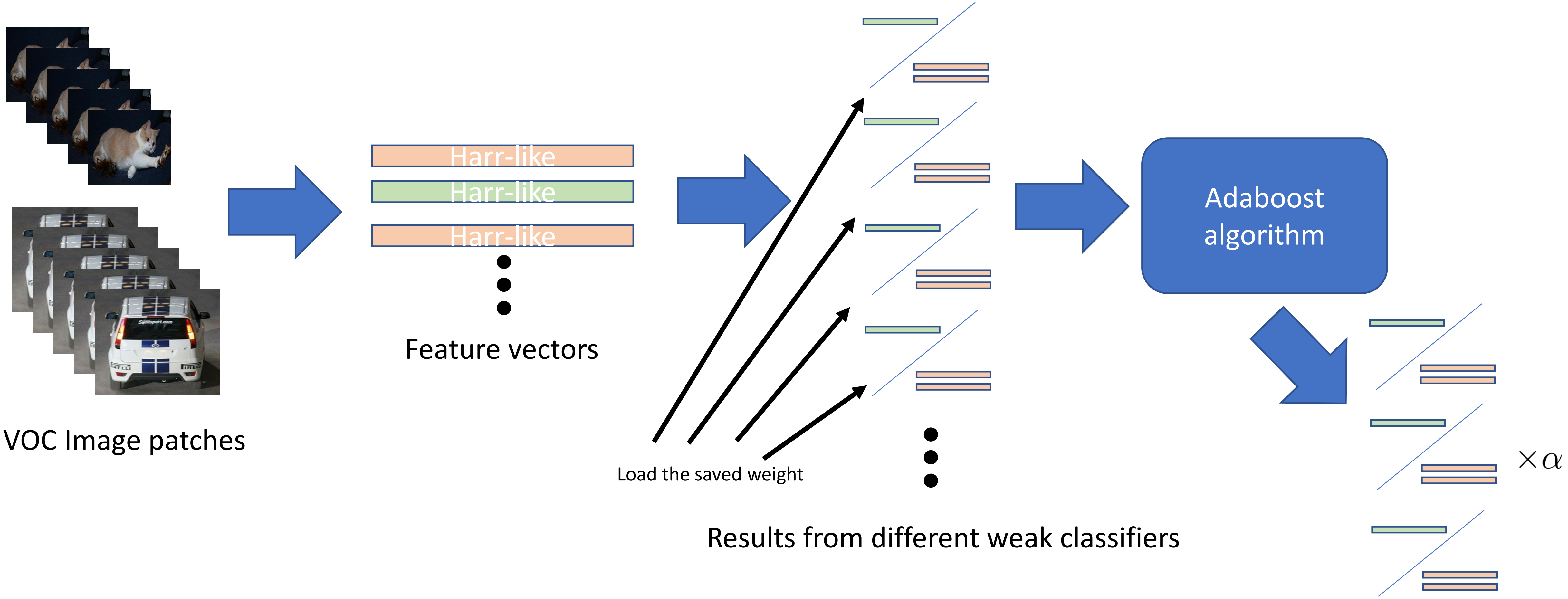
Harr-like

Strong Classifier

Load the saved weight

Result

# Overview (Adaboost)



VOC Image patches

Harr-like

Harr-like

Harr-like

Feature vectors

Load the saved weight

Results from different weak classifiers

Adaboost algorithm

$\times\alpha$

**Strong Classifier**

# Task 1 – Feature extraction (LBP/Harr-like feature)

- Create and complete the function in feature_extract.m
- fea=feature_extract(Im), fea: (LBP/Harr-like feature vector), Im: input image
- Flexible argument list, e.g.: fea = feature_extract(Im, feature_type)

# Task 2 – Weak classifiers

## 2.1 Prepare the training data

- See example in viewanno.m
- Cropped bounding box image

## 2.2 Pre-processing

- E.g.: RGB to gray, Histogram equalization, Resize (imresize())

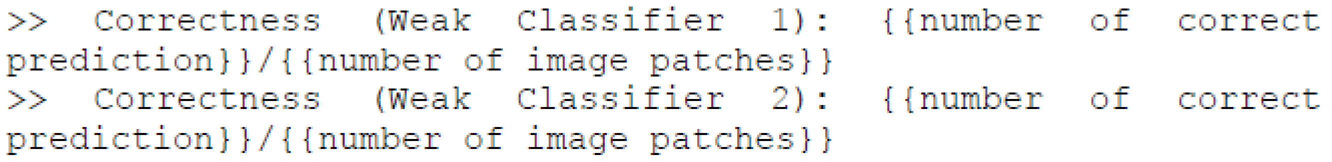
## 2.3 Training a classifier

- Binary classification (Cat or Non-cat)
- Compelete train_weak_classifier.m, model=train_weak_classifier(feature_type, ...)
- Built-in/third party classifier is allowed, e.g.: SVM, decision tree, thresholding
- **At least five unique sets of weak classifier, e.g.: different in feature type/feature-length/classifier.**
- !!! Save your models/weight/threshold as .mat file !!!
- **Model size limitation: must not exceed 15MB**

# Task 3 – Adaboost algorithm and sliding window

## 3.1 Weak Classifiers on test set

- Complete csit5410_assignment4.m
- Test all your weak classifier with bounding box images in csit5410_test.txt and output the result in the followings format:

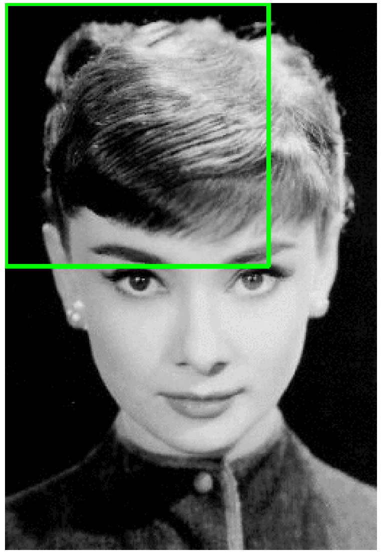


```
>>  Correctness  (Weak  Classifier  1):  {{number  of  correct
prediction}}/{{number of image patches}}
>>  Correctness  (Weak  Classifier  2):  {{number  of  correct
prediction}}/{{number of image patches}}
```

## 3.2 Adaboost Algorithm

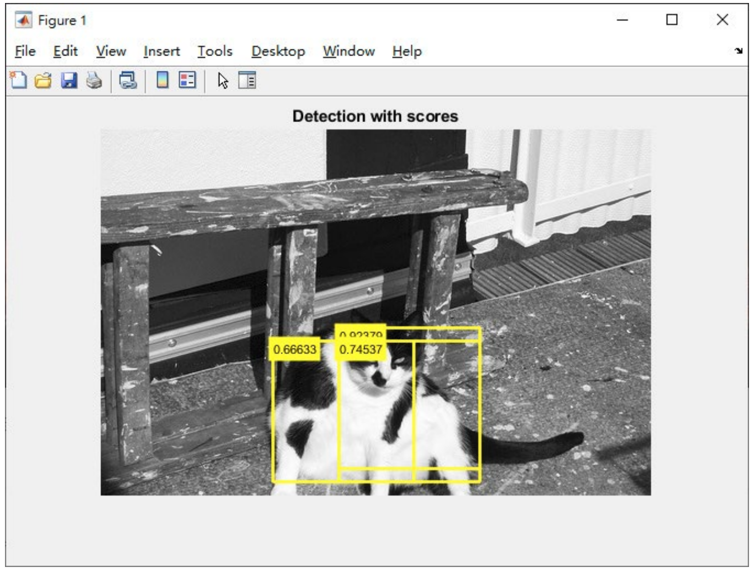
- Implement the Adaboost Algorithm (recognizing-faces.pdf)
- Display the result with images in cat_val.txt
- !!! Report your result with images in csit5410_test.txt in your written report (No need to include it in csit5410_assignment4.m)

# Task 3 – Adaboost algorithm and sliding window



## 3.3 Sliding window algorithm

- Extract fixed-size image patches from images
- Stride size: flexible
- Draw bounding box (maximum: 3 boxes) for the patches with the highest scores/confidences



# Task 3 – Adaboost algorithm and sliding window

- 3.4 Running time
  - Within 10 minutes (csit5410_assignment4.m)
  - **Make sure no runtime error**

# Task 4 – Report

- A brief explanation of your weak classifiers
- A brief explanation of your preprocessing chain (if any)
- The selected weak classifiers and its weight after the Adaboost algorithm
- The classification accuracy of each weak classifier and strong classifier on images specified in csit5410_test.txt
- The detection results of the given images in the "test_images" folder, a maximum of 3 bounding boxes per image.

# Demo

- 1. Example: viewanno('Main/cat_train')
- 2. Training: train_weak_classifier("LBP", 128, 'LBP_svm_model.mat');
- 3. Main function: csit5410_assignment4
- 4. Report

# Demo

# Q & A