

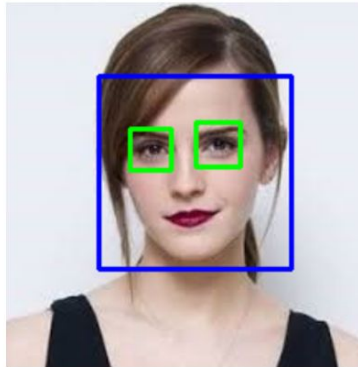


Computer Vision

Object Detection - Haar Cascade Classifiers

Object Detection / Classification

- What is object detection?
- How can we detect objects in images?



Applications of Object Detection

- Object Tracking
- Security Systems
- Autonomous Vehicles
- Etc



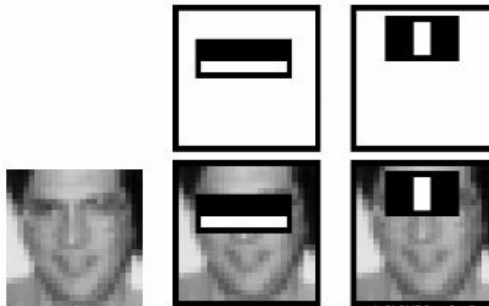
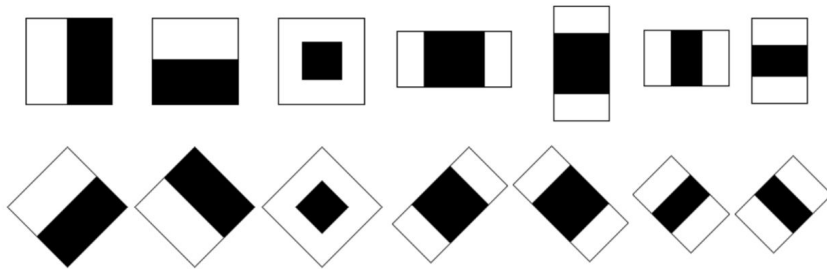
Haar Classifier - Overview

- Haar Features in images
 - Efficient calculation
- Training the classifier with images
 - Both positive and negative samples
- Selects features during training to create strong classifiers
- Weak and Strong Classifiers
 - Adaboost to make it more efficient and having less features



Haar - Features

- Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle
- Specific feature depends on what we want to detect
- How can we select the best features?
 - Adaboost - finds the best threshold that classifies the faces to positive and negative
 - Is used to find the best features and reduce them



Haar - Training

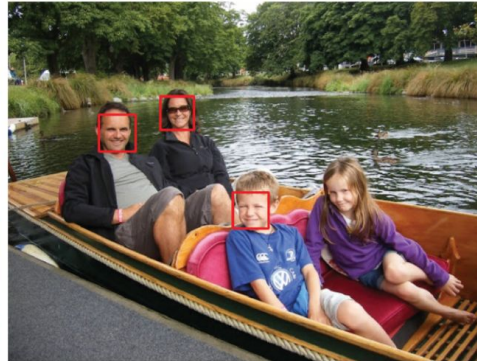
- It is possible to train it on whatever you want to detect or recognize
- Needs a lot of both positive and negative samples
- Trying to find the best features to use at each stage when training
- Adaboost
 - Optimize the number of features - only using the best ones

Haar Cascade Classifier

- Cascade of Classifiers
 - Features are grouped into different stages and applied one by one
 - The window or feature that passes all stages is a face region
- Weak and Strong Classifiers
 - Strong classifiers is a combination of weak classifiers
- Object recognition with a strong classifier
- How to improve detections and computation time?
 - Cascade of strong classifiers
 - Each stage can accept or reject features
 - Adaboost

Haar - Recognition

- What is recognition?
- Difference between detection and recognition
- Face Recognition
 - A lot of data both positive and negative samples is needed
 - And a lot of features to tell the differences
 - Needs lots of different samples of the person that we want to recognize



Haar - OpenCV

- Use the pre trained classifier
 - Optimized and robust
 - Best to use if possible
- You can make your own haar cascade classifier
 - Need a lot of images
 - Optimization

📄	haarcascade_eye.xml
📄	haarcascade_eye_tree_eyeglasses.xml
📄	haarcascade_frontalcatface.xml
📄	haarcascade_frontalcatface_extended.xml
📄	haarcascade_frontalface_alt.xml
📄	haarcascade_frontalface_alt2.xml
📄	haarcascade_frontalface_alt_tree.xml
📄	haarcascade_frontalface_default.xml
📄	haarcascade_fullbody.xml
📄	haarcascade_lefteye_2splits.xml
📄	haarcascade_licence_plate_rus_16stages.xml
📄	haarcascade_lowerbody.xml
📄	haarcascade_profileface.xml
📄	haarcascade_righteye_2splits.xml
📄	haarcascade_russian_plate_number.xml
📄	haarcascade_smile.xml
📄	haarcascade_upperbody.xml



Deep Learning - Neural Networks

- Deep Learning Tutorial in the near future
- How neural networks work and the elements of neural networks
- We are going to use python modules like Tensorflow and PyTorch
- If there is interest for it, we can code a neural network and its elements from scratch in either C++ or Python
- We are going to see how to train a neural network and how it can be combined with computer vision
- Many modern applications can be solved by using deep learning and computer vision together