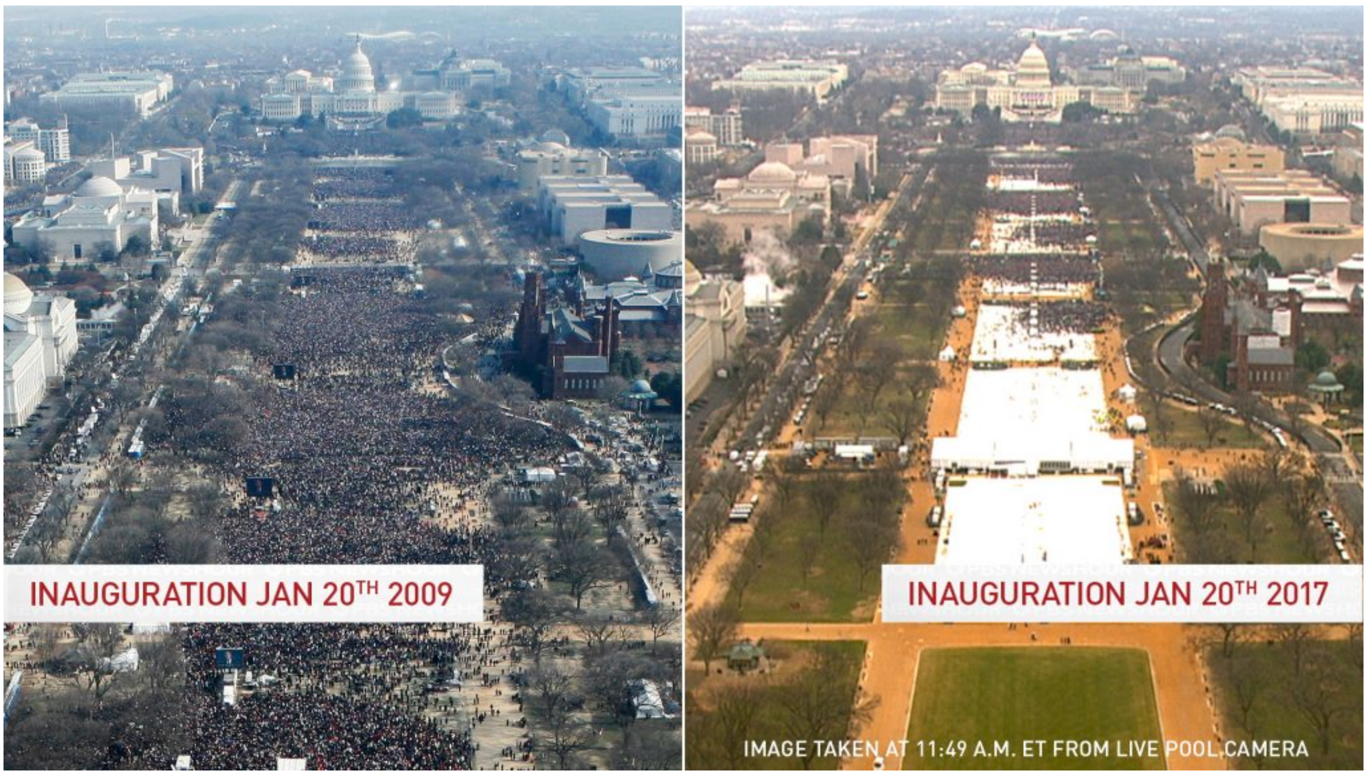
People detection and counting with Computer Vision

Why count people?

Counting people may sound like an easy task at first thought, but it has often proven itself to be a delicate and controversial task. Take Donald Trump's inauguration in Januaray 2017 for example. President Trump and his administration were not shy to claim that "This was the largest audience to ever witness an inauguration". However, given the image below, would you agree?


INAUGURATION JAN 20TH 2009
INAUGURATION JAN 20TH 2017
IMAGE TAKEN AT 11:49 A.M. ET FROM LIVE POOL CAMERA

A view of the National Mall during Barack Obama’s first inauguration in 2009 and for Donald Trump’s inauguration in 2017

Photos by Reuters and Pool Camera

Objectives

For this project, I will attempt, from a given photograph, to identify and and count the people in it using traditional computer vision techniques.

As this is an introductory project to Computer Vision, to avoid excessive complexity, I will focus my work on photographs including only a few people and not large crouds. I will work with photographs taken from angles where the people's silouhettes are fairly clear.

An Object Detection problem

The task of interest in this project is the detection of a person from an image. Indeed, once we can do this, counting them is a trivial task. The detection of people is a well studied problem in the field of Image Recognition and Object Detection.

An Image Recognition and Object Detection problem is generally solved by training a binary image classifier to identify whether a patch of an image represents the considered object or not. The methodology to build such a classifier goes as follows.

Given a dataset of labelled images:

- The images are preprocessed.
- Significant features are extracted from these preprocessed images.
- The extracted feature data is split into two subsets, a training set and a test set.
- The training data is used to train a classifier model.
- The model is evaluated and optimised using the test data.

The key step in this process is the feature extraction. To train a machine learning model effectively, the input data should describe the features that are actually significant for the considered problem. Irrelevant information, "noise" data, should be minimised. This is especially true in image recognition where only little information carried by an image is actually useful for detecting an object.

For Imgae Recogntion and Object Detection, Haar-like features, Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Feature (SURF) are extensively used.

Once a suitable feature extraction process is defined. The next issue to consider is the machine learning model. Many binary classifier model exist, it is all about finding the one that will perform best with our data.

As the options for feature extraction and classifying are vast, I looked into the academic litterature of the field and found three solutions for this people detection problem:

- **The Viola-Jones object detection framework**

Objects are detected here using Haar features of images and cascade classifiers that are a variant of the AdaBoost learning algorithm. This framework was first proposed in a paper by Paul Viola and Micheal Jones and proved to be efficient in detecting human faces.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001, 2001

- **Histogram of Orientend Gradients (HOG) and Support Vector Machines (SVM)**

The significant features of an image are extracted and vectorised using a HOG descriptor. SVMs are then trained and used to detect objects within an image. This framework was proposed by Dalal and Triggs who published a paper, proving it's efficiency for human detection fitting this project's needs very well.

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05, 2005

- **Deep Learning**

Nowadays, deep learning algorithms are by far the best performing solutions to image recognition and object detection. Human detection using a deep learning framework should be no exception.

As I want to explore traditional computer vision techniques in this project I will not go with Deep Learning right away. Although the Viola Jones framework is a reference in the field of image recognition and performs well with human faces, it is not so efficient when it comes to detecting entire people. On the other hand, using HOG and SVMs, as detailed in Dalal and Triggs paper, has a proven efficiency for human detection. I will thus go forward implementing a **HOG + SVM** people detector in this project.

Building a HOG + SVM people detector

For this experimental part of the project, I will extensively use the OpenCV (http://opencv.org) library.

Collecting a dataset of images

Several relevant datasets are available online:

- The INRIA Person Dataset (http://pascal.inrialpes.fr/data/human/)
- The Penn-Fudan Database for Pedestrian Detection and Segmentation (https://www.cis.upenn.edu/~jshi/ped_html/)
- The Caltech Pedestrian Detection Benchmark (http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/)

I collected images from these sources and stored them in a Samples folder at the root directory of my project. These images are organised in two sub-folders, one for positive samples (images with people) and the other for negative samples (images with no people).

Extracting the significant features of the images

The fundamental idea behind the HOG descriptor is that the appearance and local form of an object in an image can be described by the distribution ("Histogram...") of the direction of it's edges ("...of Oriented Gradients").

The input image is divided into given size cells. Each cell is filtered using [-1,0,1] and [-1,0,1]T kernels to obtain the intensity and orientation of the gradients. Sobel and Gaussian filters have also been tested but have not produced better results. A histogram of the orientation of the gradients is then built for each cell. The histograms are then normalised over bigger sized and overlapping blocks that group adjancent cells to improve the robustness to lighting changes.

As a result of this process, we get a single vector describing features that are actually significant in describing the object, which is what we need to effectively train our classifier.

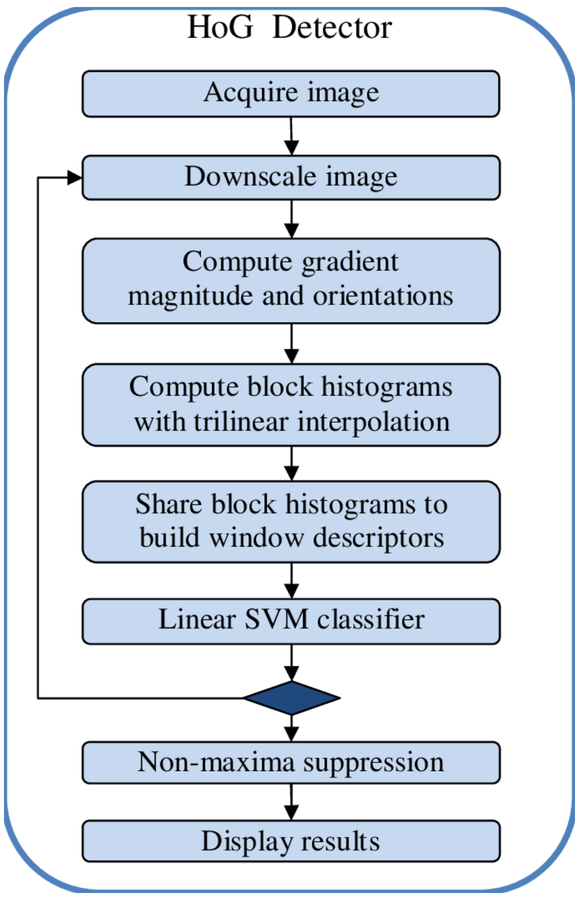

HoG Detector
Acquire image
Downscale image
Compute gradient magnitude and orientations
Compute block histograms with trilinear interpolation
Share block histograms to build window descriptors
Linear SVM classifier
Non-maxima suppression
Display results

Figure downloaded from Berkin Bilgic's paper (https://goo.gl/v934WD), Fast Human Detection with Cascaded Ensembles, 2010

The HOG descriptor

For this project, I chose to implement Open CV's HOG descriptor module. The values chosen to configure my HOG descriptor are those recommended in Dalal and Triggs paper on human detection using HOG.

In order to compute a HOG feature vector, the input image should have 1:2 aspect ratio and be resized to 64x128. This is done in the pre-processing step. The HOG feature vector is then computed and store in a list with a label indicating whether or not the input image contained a person. The HOG feature vectors and labels consists in the input data of the classifier in the following step.

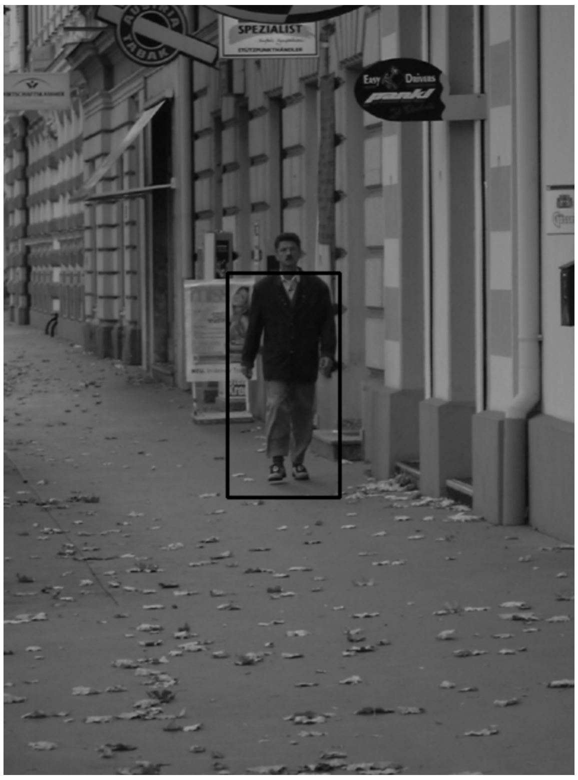
Training a Support Vector Machine classifier

Almost all machine learning libraries include SVM modules. I have first trained an SVM classifier from my favourite machine learning library, Scikit learn. I secondly trained OpenCV's built-in SVM classifier that relies on the LIBSVM library with which one can build an actual HOG detector to be applied on test images.

The input data is firsty split into a training and test set. The training set is given as input to train the model. The support vector machine model is configured to use a Radius Basis Function as kernel, as recommended in Dalal and Triggs's paper. Once the model trained, the test set is used to evaluate the performances of the models.

Both Scikit learn and Open CV's LIBSVM implementations of support vector machines produced very similar results which were not very satisfying. Barely 60% of the test images were properly classified. This is undoubtedly due the very small dataset used to train the classifiers. Had I taken more time to gather a larger and better preprocessed dataset of images, far better results should be expected.

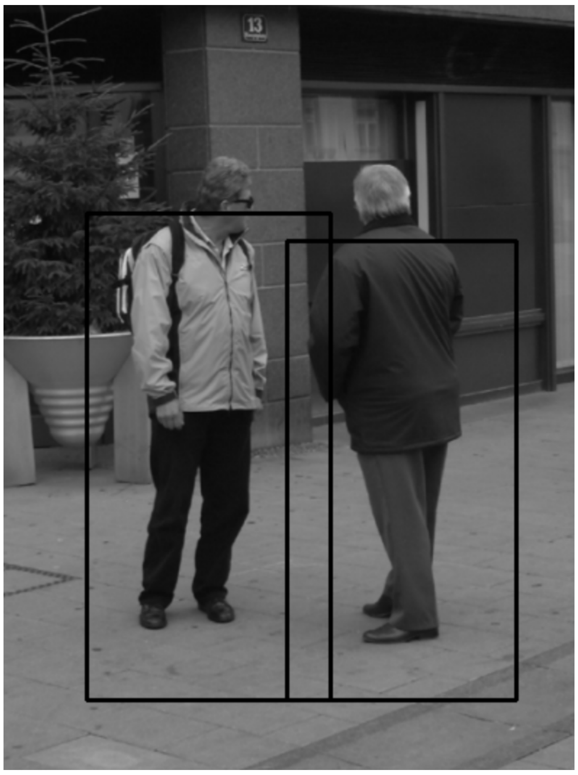
Nevertheless, as shown below, some images produced successful results:



OpenCV's pre-trained SVM classifier

The reason why I did not spend too much time gathering a larger and better dataset of images is because Open CV already offers a pre-trained SVM module for people detection. This SVM module was trained with the HOG feature vectors of a large dataset of images (apperently with the INRIA person dataset). As with the custom trained SVM module, this pre-trained SVM module can be loaded into OpenCV's HOG detector module and tested on images. The results with this pre-trained module are far more satisfying, reaching over 85% of accuracy.

Here is an example of a successfull person detection with Open CV's pre-trained SVM classifier:



Counting the detected people

Once a good HOG detector in hand, counting the people in the images consisted in counting the best fitting bounding boxes produced by the HOG detector. In order to do so, I used non-maxima suppression. This consisted in eliminating closely overlapping detected bounding boxes so only one box per actually detected person remained.

Conclusion

Throughout this project I have explored traditional computer vision techniques to build a functional person detector implementing a HOG + SVM classifier. This implementation enables me to effectively detect and count people in a given image, fulfilling the objectives I had set for this project.

References

- *P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001, 2001*
- *Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05, 2005*
- *B. Bilgic, Fast Human Detection with Cascaded Ensembles, 2010*

Ressources

- *The Learn OpenCV (https://www.learnopencv.com) website by Satya Mallick*
- *Machine Intelligence's object detector guide (http://www.hackevolve.com/create-your-own-object-detector/)*
- *The HOG person detector tutorial (http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/) by Chris McCormick*