**PEDESTRIAN DETECTION DOCUMENTATION**

**INTRODUCTION**

In recent years with the increased interest in the security and surveillance application, pedestrian detection has become an important problem in Artificial intelligence. The accurate and robust detection of pedestrians is a challenging task when it comes to AIs. The convolutional neural networks functions as a joint feature extraction and feature classification model, eliminating the need for separately designing a robust feature descriptor and training a classification model. Pedestrian detection is an essential and significant task in any intelligent video surveillance system.

In our project, we will be using the linear SVC and the HOG for the Detection and with the Localization we will be using sliding windows.

**OVERVIEW**

The pedestrian detection is formulated as a binary classification problem, where a candidate pedestrian with a box bounded is classified as a pedestrian. Our Algorithm for this project has one main component which is binary classification using trained convolutional neural network.

**METHODS**

We have dataset which comprises of Positive class and Negative class. The Positive class comprises of list of images containing human beings which is the entity of interest and the Negative class also comprises of images but without human beings.

Extraction of all the features of the positive class images and the negative class images are done using HOG. These classes need to be given an identity which is 0 and 1. For images with humans we get 1 and for images without humans we get 0 by running the prediction algorithm. =

The positive images are manually analyzed and all location of the entity of interest or recorded.

We record the positons X and Y and also the width and height of the bounding box around the entity of interest (a human being).

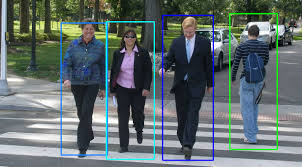
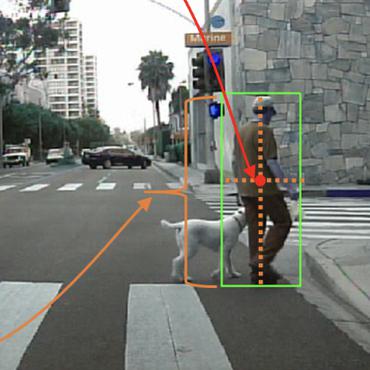
These annotations are then used in the program to localize and extract the features of the bounding box using hog features extraction.

For the negative images, we randomly select about ten sections of the image and also extract its features. We train a development model with these features for the positive and negative classes.

We then use this development model to find hard negative features for most of the images in the negative class by running prediction on the images in the negative class. For these images that are predicted as positive we record them as hard negatives and retrain the model with this new extracted features

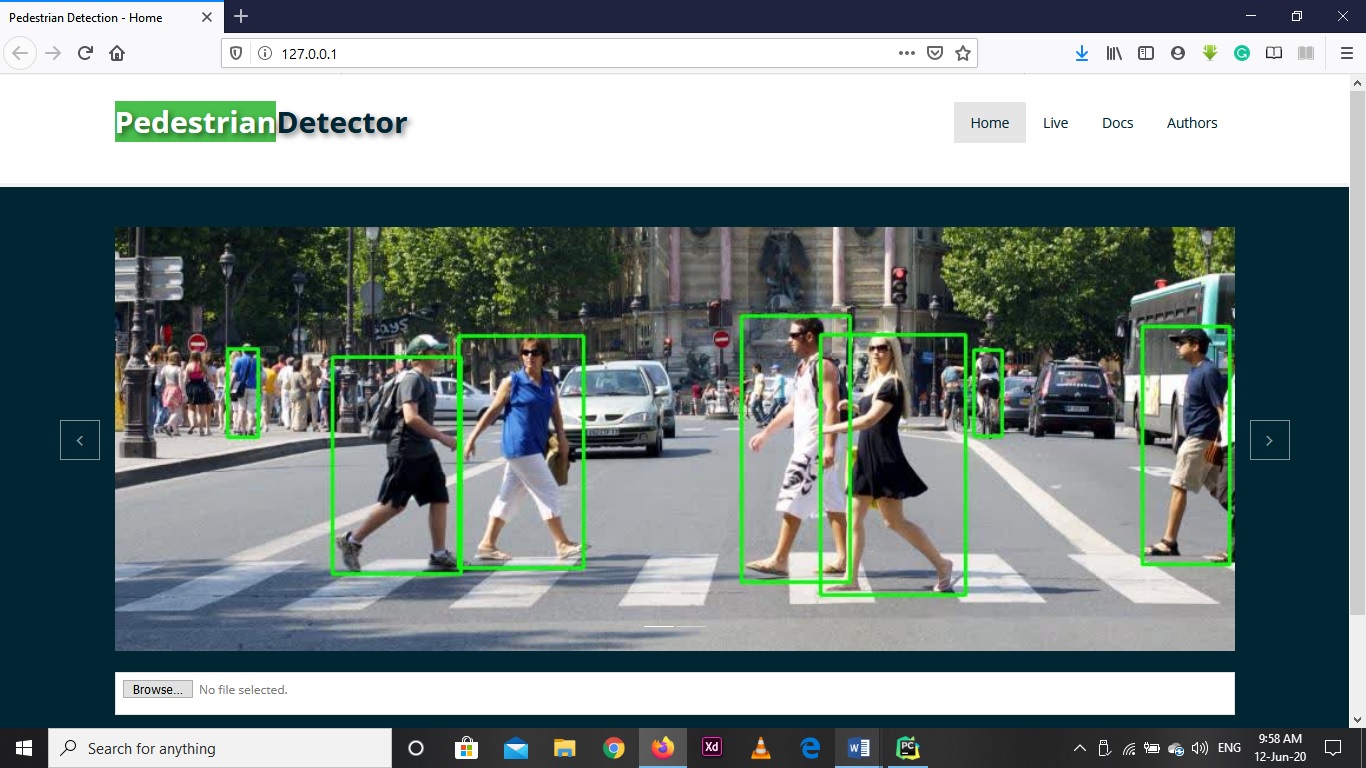
The HOG (Histogram of Oriented Gradients) is a feature descriptor that is often used to extract features from image data. It focuses on the structure or the shape of an object. These orientations are calculated in localized portions. These means that, the complete image is broken down into smaller regions and for each region, the gradients and orientations are calculated.

Sliding windows is a technique which involves using a bounding box of a certain size to slide over an image in an incremental fashion until the full size of the image has been covered.

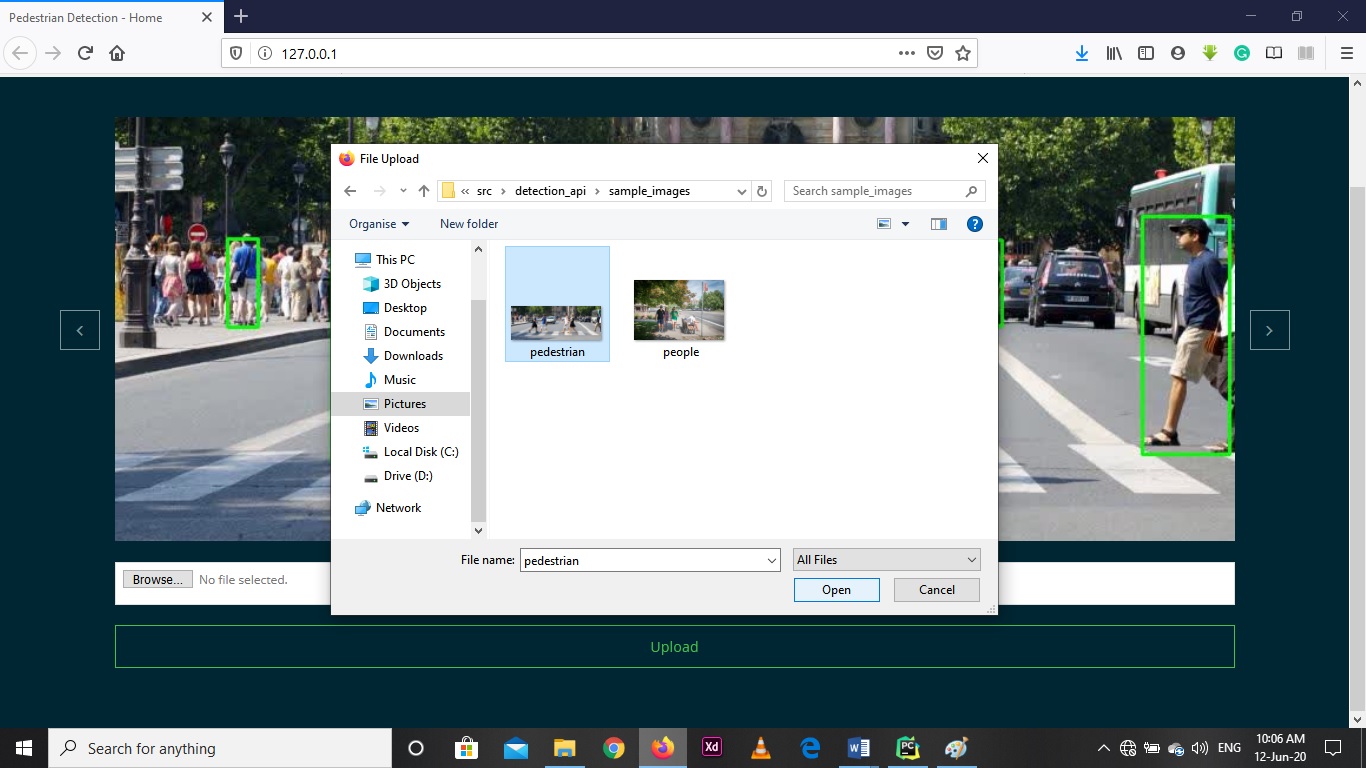
**USER MANUAL**

Home.

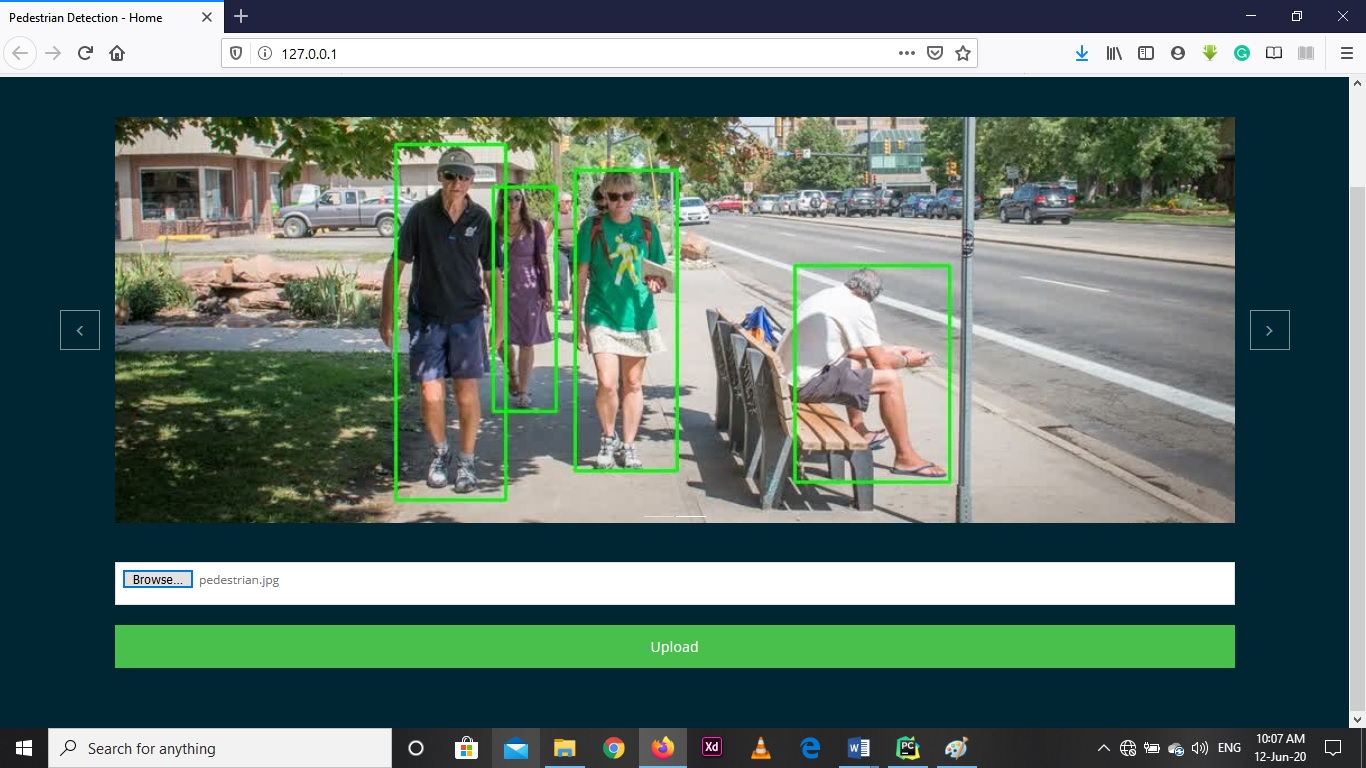


Upload

Select a file (image or video) of your choice for processing.

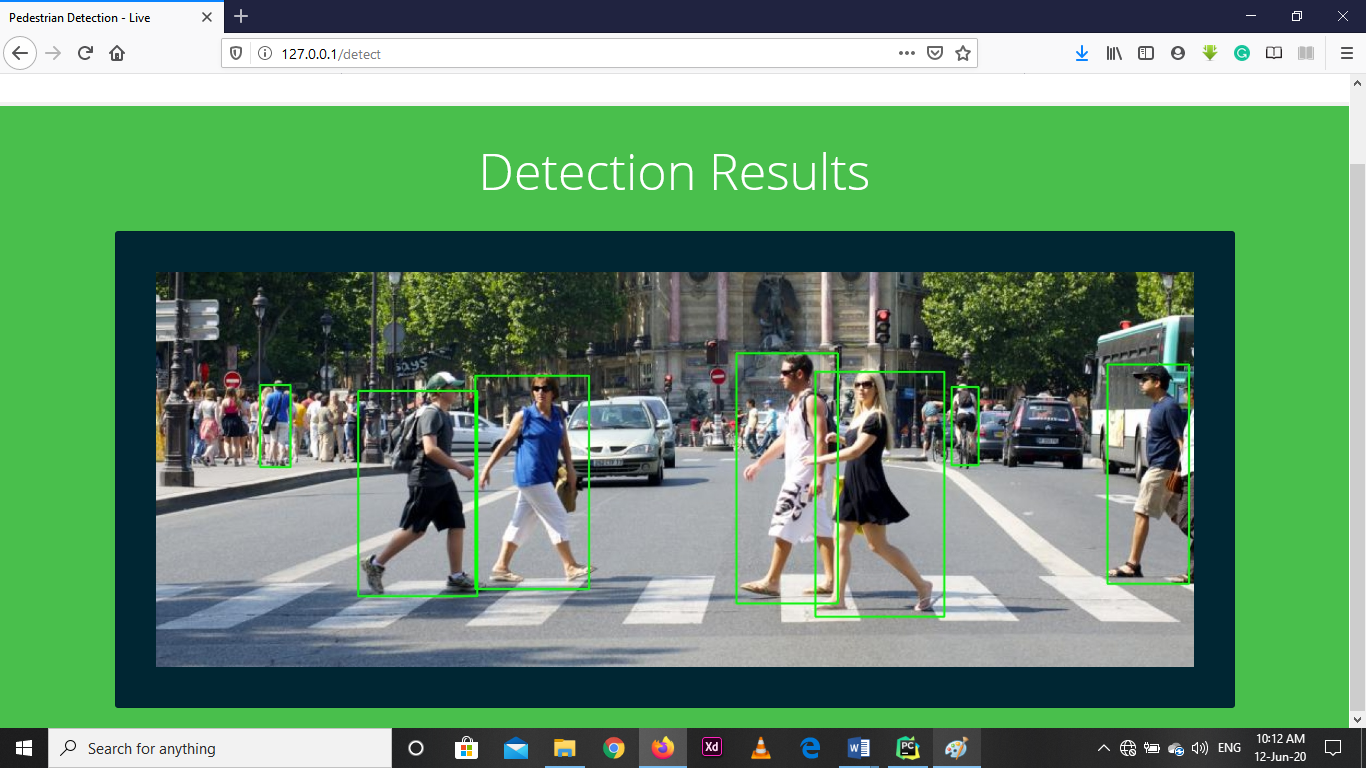


Click on upload and wait while the program process the media file



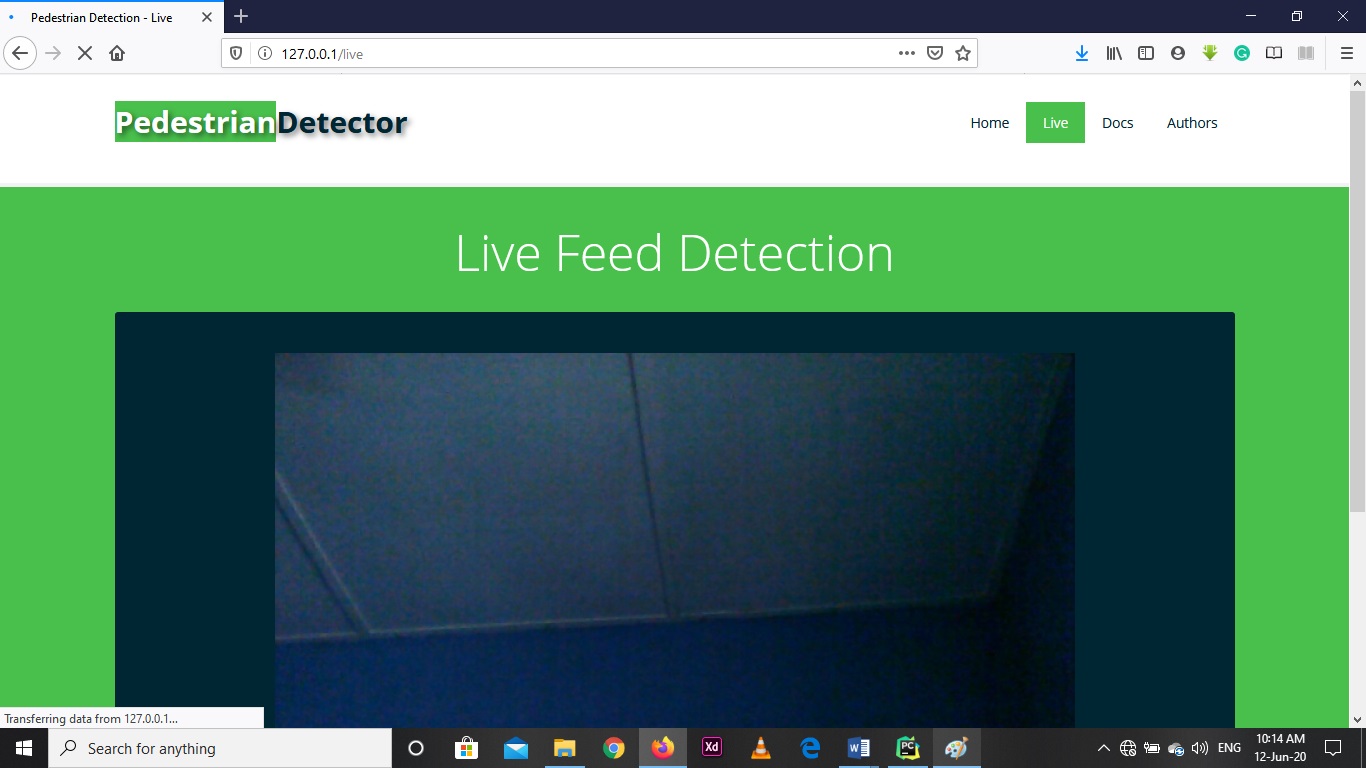
You will be taken to a new route with the processed image or video

Note: The detection program might run longer on computer with low specs.



For Live processing from the camera feed, click on live in the menu options

(This process might also run slowly on low spec computers)



Note: the camera was active in a room hence no detections were found

**DATASET LINK**

http://fivedots.coe.psu.ac.th/~kom/?p=1511

**GUIDE**

* **Libraries used**
* Flask for the web app.
* Python-open-cv for hog feature extraction
* Skimage for image processing.
* Scikit learn (sklearn) for preprocessing and building the model
* Numpy and pandas for manipulating the dataset.
* Matplotlib for basic visualizations
* Keras for the convolutional neural net.

**CONCLUTION**

Other libraries were used but for the ones listed above are the important ones.

The Convolutional neural network is used for binary classification. Moreover, compared with the sliding window framework, the candidate pedestrian detection framework significantly reduces the computational complexity. For the Evaluation on larger data sets, incorporate tracking to further improve the detection accuracy and computational complexity which will be added as time goes on in our project.

**MEMBERS**

10680673 – Godfred Twene

(Research and Documentation)

10660742 – Atinga Bernard Azumah

(Web App and Models Code Implementation)