16623 Project Proposal
Pedestrian Detection based on iOS

Team name: FlyToSky
Members: Xingchen Yu(xingche1), Yuzhang Wu(yuzhang2)

Summary:
We want to implement pedestrian detection on iOS platform so that when users take advantage of our application, they can turn on the camera and see where are pedestrians in the scene. Our goal is to design an app that can immediately recognize the pedestrians. This app can be used in many ways such for security check and unmanned vehicle.

Background:
Pedestrian detection is defined as: check pedestrians in the video frames, if there is, give location information. It is the first step of vehicle auxiliary driving, intelligent video surveillance and human behavior analysis. In recent years, pedestrian detection is also used in the emerging fields, such as aerial images, victim rescue. Pedestrians have both rigid and flexible object characteristics, and the appearance is easy to be influenced by wearing, scale, shielding, attitude and angle of view, which makes it difficult and hot spot to develop pedestrian detection[1].

Challenges:
We need to admit that there are several methods that can do pedestrian detection and the accuracy is reasonable but there are few real-time projects using camera based on iOS platform.
We think the challenges lie in first how to select feature extraction methods. Because it will cost a lot if we use a complex algorithm. When implementing on iOS, it's also challenging to find how to improve the performance on iOS using the technology from class.

Goals & Deliverables:
Datasets:
INRIA pedestrian database
The first is the well-established INRIA pedestrian database, containing 3506 64×128 images of humans cropped from a varied set of personal photos. It contains various views with a relatively large range of poses.

CVC02 pedestrian database
CVC-02 consists of 2054 images of humans. The imagery has been recorded in urban scenarios around Barcelona (Spain), using a Bumblebee color stereo camera with resolution 640x480 pixels and 6mm focal length.

CVC07 pedestrian database
This dataset contains 4824 pedestrian images. The pedestrian images have frontal view and left view, which are annotated as ‘M’ and ‘L’. You may flip the pedestrians to get right view examples.

Methods:

	low-level features	classifier	original implementation	trained on INRIA data	per-image evaluation	sec / frame	model height (in pixels)	scale stride	publication
VJ[40]	Haar	AdaBoost		✓		7.0	96	1.05	’04
HOG[5]	HOG	linear SVM	✓	✓		13.3	96	1.05	’05
FtrMine[7]	gen. Haar	AdaBoost	✓	✓		45	96	1.20	’07
Shapelet[30]	gradients	AdaBoost		✓		60.1	96	1.05	’07
MultiFtr[42]	HOG+Haar	AdaBoost	✓	✓	✓	18.9	96	1.05	’08
LatSvm[11]	HOG	latent SVM	✓		✓	6.3	80	1.05	’08
HikSvm[21]	HOG-like	HIK SVM	✓	✓		140	96	1.20	’08

This table is an overview of tested algorithms. All approaches use sliding windows and NMS (all except LatSvm use kernel density estimation for NMS). All use variants of HOG or Haar features and are trained with variations of boosting or SVMs. LatSvm was trained on the PASCAL dataset, the others on the INRIA pedestrian dataset. Only LatSvm and MultiFtr reported results using per-image measures, the rest of the algorithms were originally evaluated using per-window measures[2].
The methods in the table give us a general idea about how to extract pedestrian features from images and train classifiers. We will refer to these methods and find an adaptive one which performs best on iOS.

Schedule:

1. Select appropriate positive and negative images from datasets as our training samples;
2. Write codes and implement for each feature extraction method ;
3. Train the classifiers and test the accuracy of them;
4. Add each classifiers to iOS project and analyze the performance of them on iPad;
5. Choose the best one and try to use the knowledge which we learned from 16623 class to improve the performance of the results.

References:

[1] Dollar P, Wojek C, Schiele B, et al. Pedestrian Detection: An Evaluation Of The State Of The Art[J]. Pattern Analysis & Machine Intelligence IEEE Transactions on, 2012, 34(4):743-761.
[2] Park, D., Ramanan, D., Fowlkes, C.: Multiresolution models for object detection. ECCV, 2010.