Kadie Clancy
Independent Study
Pedestrian Detection Techniques

Previous Work on Pedestrian Detection

There has been high demand for technology to be able to detect and distinguish certain objects from their surroundings in images and video. The detection of humans has become a problem at the forefront of object detection, namely due to the application to autonomous driving, intelligent surveillance systems, and robotics. Google currently employs radar, lidar and cameras in their autonomous vehicles to avoid to detect and avoid human beings [16]. These systems can be very expensive. In fact, the spinning lidar roof unit equipped on these vehicles costs over 10,000 dollars alone [16]. If computer science can reliably develop a technique to detect humans in real time using relatively cheap cameras, applications will become more affordable and more accessible. This will contribute to the widespread use of robotics and autonomous systems.

In the last decade, tremendous progress has been made in the field of pedestrian detection, which is the detection of humans specifically in upright positions. In fact, there have been over forty proposed methods to solve this problem [4]. The problem has been approached in several ways and these methods have improved performance, computational speed, or both at each stage of evolution. Pedestrian detection is a particularly challenging form of object detection due to the variance in factors of human beings such as pose and clothing, on top of challenges for object detection in general including shadowing, lighting variation, and occlusion. Due to the complexity of the problem, all approaches borrow computer vision and machine learning techniques.

Computer vision is a subfield of computer science that deals with methods to reconstruct, interpret or understand a three dimensional scene from a two dimensional digital image. Machine learning is a subfield of computer science that uses pattern recognition to give computers the ability to “learn” and predict without being explicitly programmed. When used together, techniques from these two subfields provide the tools for systems to intelligently perceive their surroundings.

Pedestrian detection systems are evaluated against well-established benchmark datasets including (from most to least challenging) the MIT pedestrian dataset, INRIA Person dataset, and the Caltech Pedestrian Database [4, 8, 7]. The MIT pedestrian dataset contains 509 training images and 200 test images of pedestrians in city scenes plus left-right reflections of these [9]. This is considered the least challenging of the three sets as it contains only front or back views with a relatively limited range of poses. The INRIA Person data set contains 1,805 images of usually upright humans cropped from original photos [8]. The Caltech Pedestrian Benchmark is approximately 10 hours of video taken from a vehicle driving through regular traffic in an urban environment that included 2,300 uniquely annotated pedestrians [7]. This dataset is the most challenging yet realistic dataset for application mentioned.

Early detectors employ a similar “sliding window” approach, with the main difference lying in the extraction of different features or the use of different machine learning algorithms being as classifiers. Each detection approach will have testing and training data. This data may be in the form of still images or video segments. Certain features will be extracted from this training and testing data. Feature extraction in general means reducing or converting the pixel values of an input image into something meaningful. In the case of pedestrian detection, what is "meaningful" in an image can be edges, color channels, or something unintuitive determined by a learning algorithm. These features then train the machine learning algorithm of choice for each detector. The testing image is then segmented into a number or windows where the machine learning algorithm will iteratively be run over the image in different scale windows and predict whether or not the window in question is a pedestrian or not. These approaches can be placed into three main categories: Histogram of Oriented Gradient (HOG) features + SVM, Deformable Parts Models (DPM), and Deep Neural Networks [4].

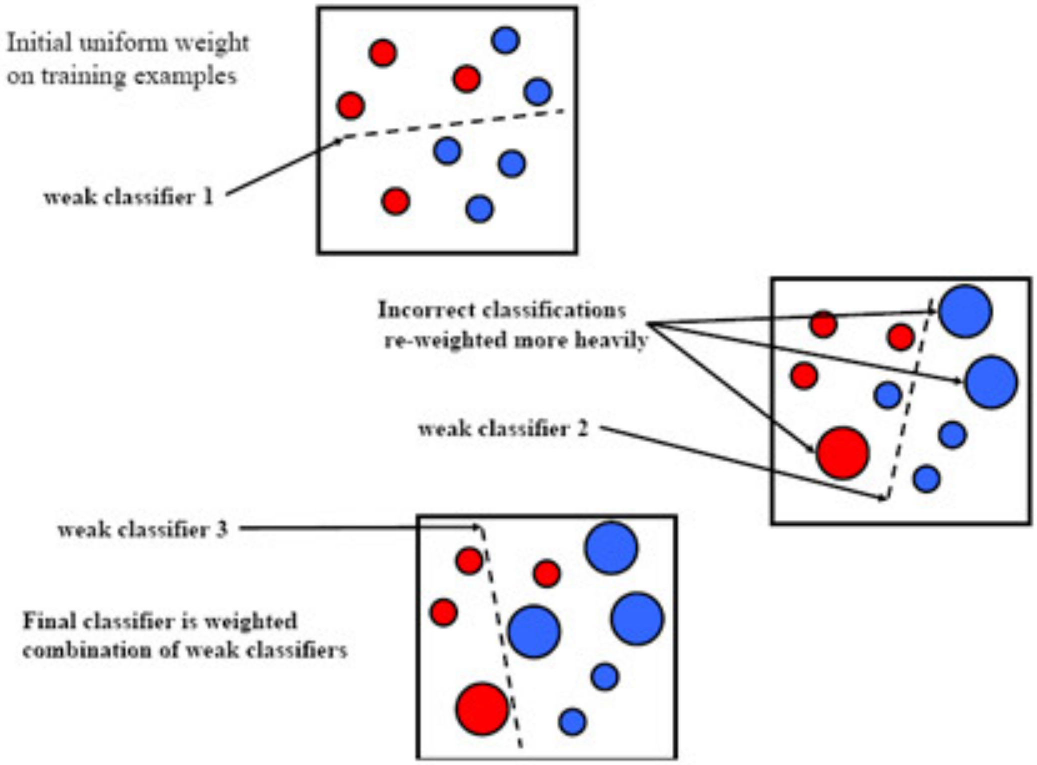
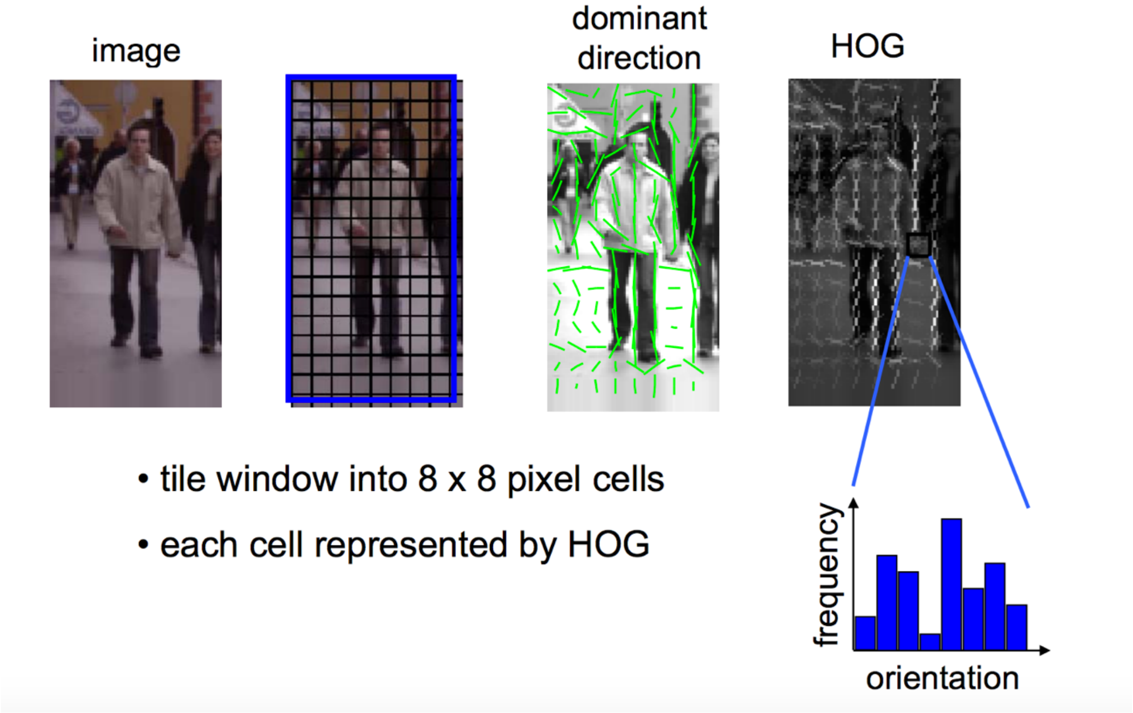


Figure 1. Kim, Kihwan. Description of Adaboost. Digital image. Computer Vision Final Project. Web. 1 Mar. 2016.

Pedestrian detection began in 2003, when Paul Viola and Michael J. Jones applied their VJ detector coupled with the learning algorithm AdaBoost to the problem [18]. AdaBoost, short for Adaptive Boosting, is a learning algorithm from the family of boosting algorithms. Boosting algorithms produce accurate prediction rules by combining several relatively weak rules (see figure 1) [11]. Specifically, AdaBoost calls a given weak learning algorithm repeatedly in a series of rounds where the initial weight of each example is equal. Through subsequent rounds, the weights of misclassified examples are increased while the weights of correctly classified examples are increased. This forces the algorithm to concentrate on difficult examples [13]. The VJ detector used image gradient information and motion information to detect walking pedestrians even under difficult conditions like rain and snow [18]. This detector is used only on video as the motion information is an integral part of the algorithm.



- tile window into 8 x 8 pixel cells
- each cell represented by HOG

Figure 2. Zisserman, Andrew. Feature: Histogram of Oriented Gradients. Web.

In 2005, Navneet Dalal and Bill Triggs developed the Histogram of Oriented Gradient (HOG) feature detector. The idea behind these particular features is that the appearance of a local object can usually be well characterized by the edge direction or orientation of the gradient. HOG features are computed in the following way [9]. First, the entire image is divided into 8 by 8 pixel "cells." For each cell, a one dimensional histogram of gradient directions is computed over the pixels in each cell. These histograms are contrast normalized to account for variance due to illumination or shadowing over several cells called "blocks." These histogram values are then converted into a vector for each testing or training image. Dalal and Triggs coupled HOG features with linear SVM due to speed and simplicity of the learning algorithm [9]. Linear SVM (Support Vector Machine) is an algorithm that is able to distinguish and therefore predict between two classes of data. In this case, SVM is predicting between pedestrian and non-pedestrian image segments. The "linear" portion of linear SVM comes from the fact that the data can be linearly separated when the classifier is trained [11]. SVMs are well suited for high dimensional classification problems utilizing only a small training data set [10].

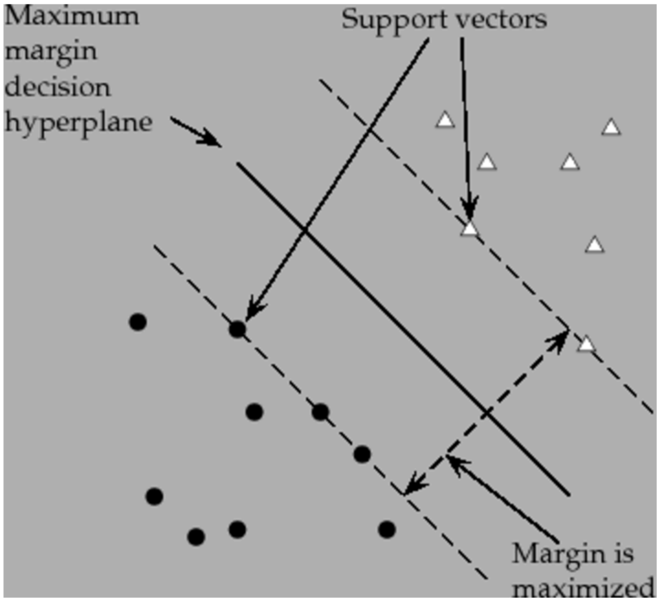


Figure 3. Manning, Christopher, Raghavan, Prabhakar and Schütze, Hinrich. The Support Vectors Are the 5 Points Right up against the Margin of the Classifier. Digital image. Introduction to Information Retrevial. Cambridge University Press, 2008. Web.

Support vectors are the data points that lie closest to the decision surface and these are the only points that contribute to the line separating the two classes. SVMs maximize the margin around the line separating the two classes. While there may be any number of lines effectively separating the classes, SVMs search for a linear separator that is maximally far away from any point [5]. Now consider a data set like figure 4 left. Although there is no way to separate this data using a line, SVMs have the ability to project data into a higher-dimensional space using a kernel [17]. When projected into a higher dimension, the data is clearly separable with a hyper plane (see figure 4 right). While there are many kernels, a common and effective one is the dot product. The dot product kernel is simply the inner product applied to the actual vectors of data [21].

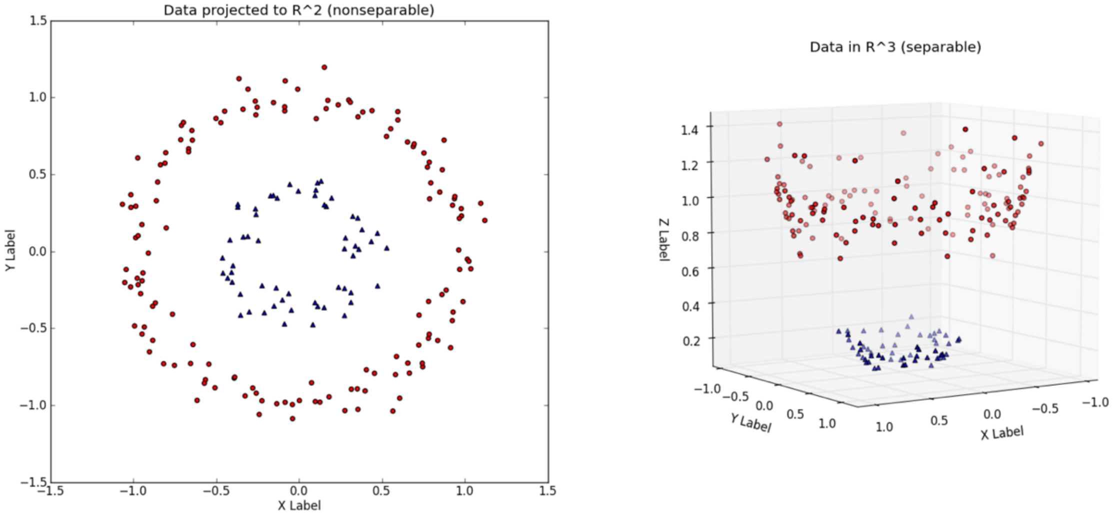


Figure 4. Kim, Eric. (Left) A Dataset, Not Linearly Separable. (Right) The Same Dataset Transformed. Digital image. Everything You Wanted to Know about the Kernel Trick. 9 Jan. 2013. Web.

After the initial proposal of the HOG+SVM detector, both linear and non-linear kernels have been tested for pedestrian detection. However, there has been no empirical evidence that a non-linear kernel provides any gains over a linear kernel when using non-trivial features [18].

----------------- RUNTIME OF SVM ---------------

Color features have been implemented alongside HOG features in a number of different ways. Shanshan Zhang, Rodrigo Beneson, and Bernt Schiele have created a detector using HOG+LUV features combined with a boosted decision forest [20]. A boosted decision forest is an AdaBoost classifier run with decision trees as the weak classifier. A decision tree is a predictive learning algorithm that maps observations about an example to its predicted value using a graph-like structure [11]. This detector focuses on a sliding window approach using only the histogram of oriented gradient and CIE LUV color channel information [12]. CIE is a color space that links between pure physical colors and physiological perceived colors in human vision [15]. Color has been shown to be a good feature for pedestrian detection as certain areas like background or human skin have a certain color bias. Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele introduce self similarity on color channels combined with HOG to improve performance on both still images and video segments [19]. The new feature on self similarity, termed CSS, is particularly color histograms from different sub-regions within the testing images. CSS does not concern the actual color value of the image's pixel which will vary from pedestrian to pedestrian, but rather can represent symmetry of color on a pedestrian's left and right sides. This seems to be a successful and descriptive feature when coupled with the edge information captured by HOG features [19].

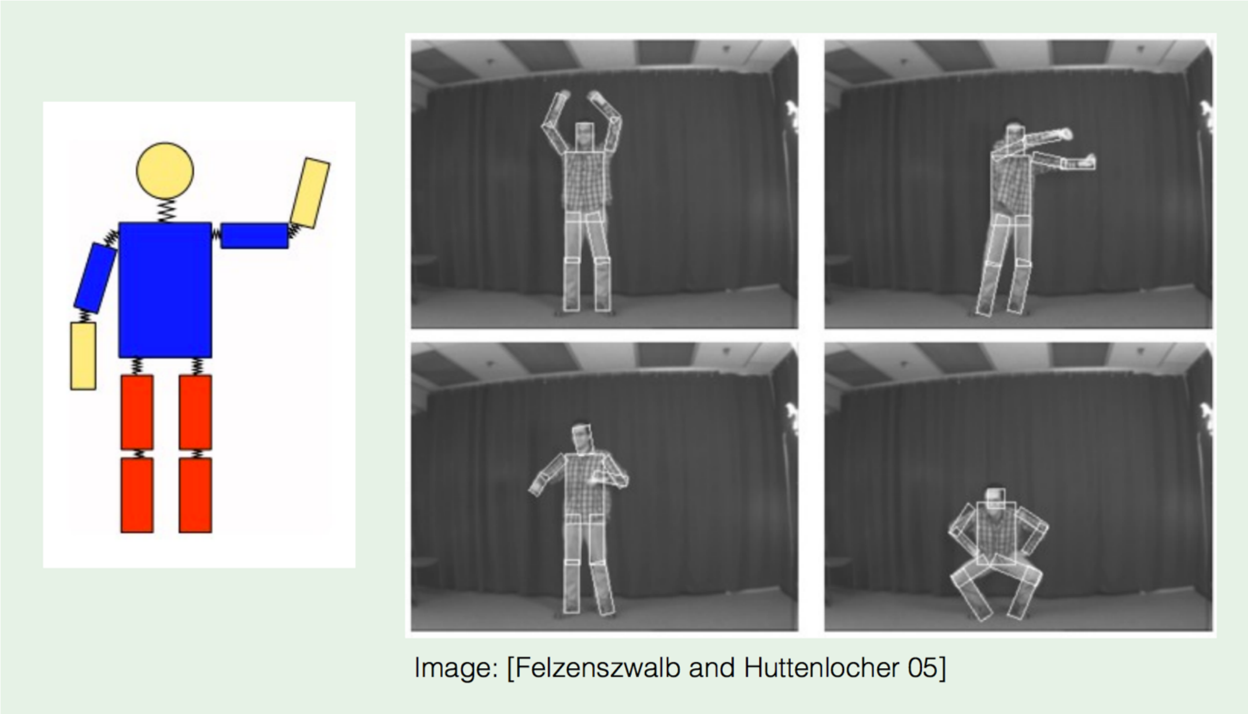


Image: [Felzenszwalb and Huttenlocher 05]

Figure 5. Girshick, Ross. Parts Based Model. Digital image. Deformable Part Models. UC Berkeley, 16 Apr. 2013. Web.

The Deformable Parts Model detector was originally designed to address the problem of pedestrian detection and has since become a very popular solution with many variants on the original [4]. The DPM detector built upon HOG features coupled with the SVM classifier with the main difference being the “deformable parts” step [12]. “Deformable parts” refers to the representation of an object model using a lower-resolution root template and a set of spatially flexible high-resolution “part” template [6]. Each of these “part” templates captures a local property of a human and the deformations are linking the separate parts together to detect a single object [6].

Another popular method of pedestrian detection that has achieved high accuracy uses deep neural networks. Deep neural networks are learning algorithms modeled after the human brain and nervous system [11]. A neural network runs based on hidden layers of decisions, which in the case of pedestrian detection may be features that are not as intuitive as edge orientation or color rather and are generated specially for the data provided [11]. Deep networks have the advantage of being able to operate on raw pixel input rather than specially developed features. While deep neural network approaches achieve high accuracy, they are very slow computationally. ------RUNTIME OF NEURAL NETS ------------Hiroshi Fukui, Takayoshi Yamashita, Yuji Yamauchi, Hironobu Fujiyoshi, and Hiroshi Murase created a pedestrian detection system based on deep convolution neural networks (CNN) that have achieved high accuracy [14]. The CNN in this detector takes in raw pixel input, edge features and normalized data [14]. Approaches using neural networks are comparable to state-of-the-art results, but are computationally slow and therefore unusable in real time system applications.

Fairly recently, Google researchers have been able to not only achieve a high detection accuracy, but have significantly sped up the detection process using deep neural networks (DNNs) coupled with cascade classifiers [2]. Unlike the previous pedestrian detection systems mentioned, this system runs in real-time at fifteen frames per second. This is a major breakthrough as real-time systems like autonomous vehicles and robots require in-the-moment detection to be useful. The detector combines a fast cascade with a cascade of deep neural networks. To gain a speed advantage, the depth of some of the convolutional layers have been reduced [2]. To further speed up the detector, this system utilizes a small convolution network. This network has only three hidden layers that are designed for speed. When the network is run in a cascade, it first processes all image patches and then will return only to the patches that have high confidence values, further reducing computation time of the detector [2]. This is a significant improvement in computation time needed to evaluate all locations and scales in the previous sliding window approach. The combination of these two neural networks and the cascade classifiers allows the detector to work 80 times faster than the original sliding window with a deep neural network alone [2]. An interesting component of this system is the fact that it is pre-trained on the ImageNet dataset. ImageNet is an ongoing research effort to provide researches with an extensive image database [1]. The data is organized according to meaningful concepts and their "synonym sets." There are currently more than 100,000 synonym sets with an average of 1,000 images per set [1]. The Google researchers have found that pre-training is minimally helpful to the overall performance, but does aid in the elimination of false positives [2]. Training on 1,000 unique object categories found in ImageNet allows the detector to better discriminate between what is and what is not a pedestrian. Eliminating false positives is especially important to autonomous vehicular applications as detection of an obstacle may cause the car to partake in unwarranted and unsafe behavior in order to avoid the perceived obstacle. Therefore, only forcing the vehicle to perform in such a way when a pedestrian is actually present is crucial. This detector is able to perform competitively on the Caltech dataset [2].

Pedestrian detection has drastically improved both in terms of accuracy and computational speed over the past ten years. Beginning with Viola and Jones VJ detector and cumulating with a real-time Google pedestrian detection system that is usable in autonomous driving and robotics applications, the field has drastically evolved to better address this complex problem in the field of object detection. There is still much work to be done on this problem, however. It is imperative to better understand why certain features are better descriptors for pedestrian detection than others, continue developing new and stronger features, and continue improving accuracy and speed of the system. Better features seem to be what drives progress forward as opposed to different classification algorithms. Pedestrian detection is a challenging problem that is currently being propelled forward by potential applications. While research continues to advance, the problem may never fully be solved for applications using computational methods alone.

Works Cited

[1] "About ImageNet." *ImageNet*. Stanford Vision Lab, Stanford University, Princeton University, Web

[2] A. Anelia, A. Krizhevsky, V. Vanhoucke, A. Ogale, D. Ferguson. “Real-Time Pedestrian Detction with Deep Network Cascades.” Proceedings of BMVC 2015 (to appear), 2015.

[3] Benenson, Rodrigo, et al. "Seeking the strongest rigid detector." Computer Vision *and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013.

[4] Benenson, Rodrigo, et al. "Ten years of pedestrian detection, what have we learned?" *Computer Vision-ECCV 2014 Workshops*. Springer International Publishing, 2014.

[5] Berwick, R. "An Idiot's Guide to Support Vector Machines (SVMs)." (2003): n. page. Web.

[6] Divvala, Santosh K., Alexei A. Efros, and Martial Hebert. "How important are “Deformable Parts” in the Deformable Parts Model?." *Computer Vision–ECCV 2012. Workshops and Demonstrations.* Springer Berlin Heidelberg, 2012.

[7] Dollár, P. "Caltech Pedestrian Detection Benchmark." Web.

[8] Dalal, Navneet. "INRIA Person Dataset." N.p., n.d. Web.

[9] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

[10] De Poortere, Vincent, et al. "Efficient pedestrian detection: a test case for svm based categorization." *Workshop on Cognitive Vision*. 2002.

[11] Duda, Richard O., Peter E. Hart, and David G. Stork. *Pattern Classification*. New York: Wiley, 2001. Print.

[12] Felzenszwalb, Pedro, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model." *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008.

[13] Freund, Yoav, Robert Schapire, and N. Abe. "A short introduction to boosting." Journal-Japanese Society For Artificial Intelligence 14.771-780 (1999): 1612.

[14] Fukui, Hiroshi, et al. "Pedestrian detection based on deep convolutional neural network with ensemble inference network." Intelligent Vehicles Symposium (IV), 2015 IEEE. IEEE, 2015.

[15] Gonzalez, Rafael C., and Paul A. Wintz. Digital Image Processing. Reading, MA: Addison-Wesley, 1987. Print.

[16] Harris, Mark. "New Pedestrian Detector from Google Could Make Self-Driving Cars Cheaper." *IEEE Spectrum*. N.p., 28 May 2015. Web. 25 Feb. 2016.

[17] Russell, Stuart J., and Peter Norvig. Artificial Intelligence: A Modern Approach. Upper Saddle River: Prentice-Hall, 2010. Print.

[18] Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: CVPR. (2003)

[19] Walk, Stefan, et al. "New features and insights for pedestrian detection."*Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*. IEEE, 2010.

[20] Zhang, Shanshan, Rodrigo Benenson, and Bernt Schiele. "Filtered channel features for pedestrian detection." *arXiv preprint arXiv:1501.05759* (2015).

[21] Zumel, Nina, and John Mount. Practical Data Science with R. Manning, 2014. Print.