# Experiments on Pedestrian and Object Detection Report

1st Shengyi Zhang
*1097063*
*Lakehead University*
zhangs@lakeheadu.ca

2nd Beili Yin
*1148875*
*Lakehead University*
byin@lakeheadu.ca

***Abstract*—As pedestrian fatalities have increased at an alarming rate over the past decade, pedestrian detection and monitoring in a surveillance system are crucial for advanced applications such as AI auto-driving, real-time monitoring, pedestrian heatmap analytics during the COVID-19 pandemic.**

**In this paper, we review and evaluate several currently available algorithms for feature extraction and detection. In addition, we introduce several existing high-quality datasets for pedestrian images and highlight the dataset of our choice. Finally, we outline a brief introduction for our research methods.**

***Index Terms*—Pedestrian detection, Pattern recognition, Machine learning**

## I. Introduction

Pedestrian detection has been studied widely in the context of video surveillance with fixed cameras and stationery backgrounds.The research of pedestrian detection began in the mid-1990s, which is a branch of object detection [19]. It has been studied widely in the context of video surveillance with fixed cameras and stationery backgrounds. Furthermore it is also a key problem in computer vision, with several applications including robotics, surveillance and automotive safety [20]. Nowadays U.S. pedestrian fatalities have risen recently, even as vehicles are equipped with increasingly sophisticated safety and crash avoidance technology. Many experts expect that advances in automated vehicle technology will reduce pedestrian fatalities substantially through eliminating crashes caused by human error [21]. Pedestrian detection is a classification problem, and it has various real-time applications such as video surveillance, intelligent driver-less cars, robotics, person tracking, etc.The specific topic of pedestrian detection is extensively studied because of its widespread applications. The studies are mostly aided by deep learning. Some of the major challenges that are being faced in pedestrian detection are complex occlusions, localisation, varying scales, etc.

The core concept of pedestrian detection consists of analyzing captured statistic image sequences and videos for possible pedestrians and generating the final set of bounding boxes. Until 2002, researchers gradually introduced more well-developed methods from pattern recognition, which focus on the available pedestrian feature vectors and singular classification algorithms.

Since 2005, the training datasets for pedestrian detection have been quantified. The detection research accuracy and speed began to accommodate the margin of errors so that pragmatic and real-time surveillance systems blossomed. With continuous research by universities, research institutes, and automobile manufacturers, pedestrian detection technology has developed rapidly.

However, there are major issues and challenges involved in either implementing singular features extracting algorithms or innovative combinations of detecting methods. Due to various situations such as video acquisition, illumination variation, abrupt motion, complex background, shadows, object deformation and crowd density, pedestrian detection remains a challenging and ubiquitously discussed topic among both research areas, and real-world applications of computer vision.

Nowadays Various approaches have been proposed, including background elimination and analysis, periodic motion, symmetry, silhouette shape analysis of the foreground, spatial-temporal silhouette analysis of human parts, annealed particle filtering for articulated pedestrian, as well as maximum posterior detection, plane and parallax decomposition, wavelet template representation, and support vector machine (SVM) classifiers etc. [22]

## II. Literature Review

### A. Feature extraction

Feature extraction branched into two research approaches: 1) focusing on pedestrians' appearance and motion characteristics by separating images of body parts into instinct key portions 2) introducing a mixed combination of feature vectors, such as computing multiple Haar-like features for optimized performance. In addition, local features have been implemented as supplements. Alternative methods are able to improve the detection rate and accommodate the error margin in order to resolve the issue of obstacle occlusion partially.

From 2001 to 2005, the early feature sets are mainly composed of grey layers and detected silhouettes, with a few exceptions, which included RGB three-dimensional layers. Since each feature presents various pertinence, obtaining better detection performance with a single feature proved to be fruitless at the time.

The period between 2005 and 2011 witnessed a leaping step towards a superior performance for feature extraction. Inspired by Scale-invariant feature transform (SIFT) proposed by [1], [2] introduced Histogram of Gradient (HOG) to achieve extracting compact pedestrian motion expression, a

novel histogram of gradient feature descriptor, which proved to be more informative than grey layered feature capture. Later, many methods that combine HOG with other feature descriptors have been proposed to solve different problems, for example, HOG+LBP [5] construct an occlusion likelihood map for occlusion handling, a new feature by self-similarity of low-level features, HOG+CSS [6] proposed a new feature by self-similarity of low-level features

### B. Detection methods

Handcrafted based pedestrian detection methods can be roughly divided into two different categories: decision forests based methods and deformable part based approaches [7].

- **Decision forests based methods**
  Decision forests based methods usually use multi-dimensional handcrafted features to represent each patch (positive or negative) in an image, followed by boosting techniques together with decision forests to select a set of most discriminative features, which are later used to train a pedestrian detector. [8] first extracts the candidate Haar features for each detection window (patch) and then utilizes the cascade AdaBoost [9] to learn the detector. ChnFtrs [10], improve the performation by using more channels such as gradient histograms, gradient magnitude and LUV color channels, with the cascade adaboost. Some variants of ChnFtrs have been proposed, such as [11], [12] focus on extracting better local features from HOG+LUV channels.
- **Deformable part based methods**
  [14] First introduced the deformable part based methods(DPM) to capture the deformation of pedestrians better. DPM is a star-structured part based model and consists of a coarse root model and a set of higher-resolution parts deformation models. HOG is used to extract the local features in each model. Based on HOG, the part based model is used to capture the deformations in objects.

In recent years, deep convolutional neural networks (CNN) have achieved great success in many computer vision tasks such as object detection [17]. Researchers also try to use CNN based methods to learn the deep features in order to solve pedestrian detection, We just introduce a few pure CNN based methods here for completeness.

Since pedestrian detection develops as a sub-category of object detection, Faster R-CNN [15] has achieved great success in object detection. However, directly apply the Faster R-CNN without any modification seems to yield unsatisfying result as shown in [18]. [16] introduced several modifications (anchor scale, feature stride, and ignored region handling etc.) to improve the performance. Furthermore, abundant state-of-the-art deep feature-based models continuously spring up, such as part-based method, attention-based methods, feature-fused methods, cascade-based methods, anchor-free methods, data-augmentation based methods, loss-driven methods, post-processing methods and multi-task methods.

## III. Dataset Overview

### A. MIT pedestrian dataset

MIT [13] is one of the first pedestrian dataset. The training set contains 924 positive samples and 11,361 negative samples, where the resolution of samples is of 128x64 pixels. There are 123 images for testing in this dataset.

### B. INRIA Person Dataset

This dataset [2] collected in 2005 as part of research work on detection of upright people in images and video, is one of the most popular pedestrian datasets and is widely used by the handcrafted features based methods.The INRIA dataset is biased toward large, mostly unoccluded pedestrians.This dataset consists of personal digital images. The training set contains 614 positive images with 1,208 pedestrians and 1,218 negative images. The test set has 288 images.

In our work we will mainly focus on the INRIA dataset as training and testing examples for detection and learning algorithms.

### C. Daimler Pedestrian Detection Benchmark Dataset

The benchmark involves a large training and testing set [4] collected in 2009. The training set contains 15,560 pedestrian samples (image cut-outs at 48x96 resolution) and 6744 additional full images not containing pedestrians for extracting negative samples. The testing set contains an independent sequence with more than 21,790 images and 56,492 pedestrian labels. It provides the baseline performance of three state-of-the-art methods (wavelet-based AdaBoost cascade, HOG/linSVM and a convolutional network NN/LRF) on the specified training and test set.

### D. Caltech Pedestrian Detection Benchmark

The Caltech Pedestrian Dataset [3], proposed at 2014, consists of approximately 10 hours of 640x480 30Hz video taken from a vehicle driving through regular traffic in an urban environment. The training data consists of six training sets, each with 6-13 one-minute long seq files, along with all annotation information. The testing consists of five sets. In addition, the annotations for the entire dataset are included in the package.

The following figure 1 is just a summary of the datasets related to pedestrian detection.

| name | publication | #images | #pedestrians | resolutions | annotations | time | description |
|---|---|---|---|---|---|---|---|
| MIT [151] | IJCV2000 | - | 924 | 64×128 | full | day | one of earliest pedestrian datasets |
| INRIA [40] | CVPR2005 | 2120 | 1774 | 640×480 | full | day | one of earliest popular pedestrian datasets |
| ETH [48] | ICCV2007 | 1803 | 12$k$ | 640×480 | full | day | a pair of images in busy shopping streets |
| TUD-Brussels [201] | CVPR2009 | 508 | 1326 | 640×480 | full | day | pedestrians in the inner city of Brussels |
| Daimler [47] | PAMI2009 | 29$k$ | 72$k$ | 640×480 | full | day | gray-color images in urban traffic |
| Caltech [45] | PAMI2010 | 250$k$ | 289$k$ | 640×480 | full, visible | day | a standard and complete pedestrian datasts |
| KITTI [55] | CVPR2012 | 15$k$ | 9$k$ | 1240×376 | full | day | a real-world computer vision benchmarks |
| Citypersons [242] | CVPR2017 | 5$k$ | 32$k$ | 2048×1024 | full, visible | day | extensions on top of the Cityscapes [35] |
| CrowdHuman [169] | arXiv2018 | 24$k$ | 552$k$ | - | full, visible, head | day | humans in crowded scenes from website |
| EuroCity [8] | PAMI2019 | 47$k$ | 219$k$ | 1920×1024 | full | day, night | images in multiple European Cities |
| NightOwls [133] | ACCV2019 | 281$k$ | 56$k$ | 1024×640 | full | night | pedestrians at night in three countries |
| WIDER Pedestrian | Challenge | 97$k$ | 307$k$ | - | full | day | pedestrians in traffic and surveillance scenes |
| WiderPerson [245] | TMM2019 | 13$k$ | 39$k$ | - | full | day | persons in the wild, not only traffic |
| KAIST [82] | CVPR2015 | 95$k$ | 103$k$ | 640×480 | full | day, night | color-thermal image pairs in traffic scene |
| CVC-14 [60] | Sensors2016 | 5051 | 7795 | 640×512 | full | day, night | multimodal (FIR+visible) videosequences |

[1] the top part is early pedestrian datasets, the middle part is modern pedestrian datasets, and the bottom part is multispectral pedestrian datasets.
[2] 'full' means the fully-body bounding-box, 'visible' means the visible-body bounding-box, and 'head' means the head bounding-box.

Fig. 1: Pedestrian dataset

## IV. Project introduction

The traditional processing pipeline for object detection is input image – proposal generation – feature extraction – classification(regression) – post processing. First, we generate candidate proposals from an input image. Second, we perform the feature extraction, which is applying different descriptors to extract different features from the proposal. Third, the proposals are assigned to either the positive class(pedestrians) or the negative class(background). In the end, some post processing methods aimed to suppress duplicate bounding boxes belonging to the same pedestrian are used, such as NMS (Non-maximum Suppression) and its variants.

We will perform basic data preprocessing, which consists of normalizing images for sampling and feature selection, reducing feature dimension by performing Principal Component Analysis (PCA) to attain optimized feature vectors for classification.

Combining single classifiers (KNN, SVM, Naïve Bayes, MLP, etc.) through ensemble learning to improve outpu stability and reliability. Multi-classifier System (bagging ar adaboosting) will be implemented for evaluating detectic performance (as in the figure 2). The fundamental workflo

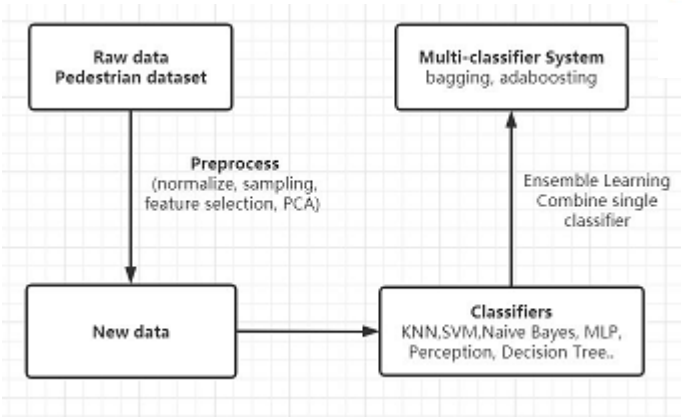



Fig. 2: Research workflow

of our experiment will proceed as follows: 1) Collecting raw data for pre-processing; 2) implementing multiple classifiers on the processed feature vectors and annotations for evaluation and comparison.

## V. Methodology and Justification

### A. Objective

To develop classifiers, which take feature vectors as input and predict the labels and boxes. With different feature extraction methods, combined with a hard sample mining post-processing approach, we will analyze and evaluate the advantages and disadvantages of various learning models.

### B. Pre-processing

Demonstrate the reason why the specific approaches were applied in this experiment. Several problems may arise if working directly working on the raw images.

- **Image Size** The dimensionality of extracted feature vectors depends on the image size, and it is not scalable. It is better to use fixed-size image patches.
- **Redundant Information** Without pre-processing, useful information and features can be overlooked, which will result in a low accuracy rate.

As computer vision technology advances, developers and researchers have found many efficient features that can proficiently depict the characteristics of images. Such as Haar-like features, HOG features, LBP features, SIFT features, SURF features, etc.

After extracting features, the local dimensions are deduced. For example, the dimensionality of HOGfeature vectors is 3780 (over a window size of 64x128 pixels, block size 16x16, cell size 8x8 and stride 8). And these feature vectors serve as input for classifiers such as SVM, MLP, Naïve Bayes, etc. to actively learn the models. Before deep learning, many

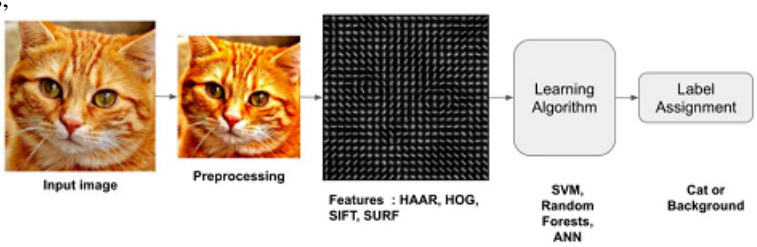



Fig. 3: Preprocessing

traditional computer vision image classification algorithms follow this pipeline, while deep-learning algorithms completely bypass the feature extraction step.

Often an input image is preprocessed to remove contrast, noise, and brightness effects. A very common practice is to subtract the mean of image intensities and divide by the standard deviation. Sometimes, gamma correction will be applied to get better results. For RGB images, sometimes a color space transformation can contribute to better performance, such as convert the image from RGB to a gray image. In most cases, the benefit of preprocessing is unpredictable. Through trial and error, we are able to decern what to expect from each approach.

For feature extraction, the size of input images should be fixed. Input images need to be cropped and resized into the same shape and color space, which is another reason that the preprocessing step is essential before shoving any input data into classifiers.

### C. Feature Extraction

The raw input images have too much redundant information that does not contribute to the classifiers even after the detailed preprocessing mentioned above. The significant changes in image pixels' properties, carry more weights, such as the edges, represented by gradients in different directions.

The next step is to extract such important information in the image and record their attributes and frequencies for the specific features or shapes. For example, no matter what color

and type of clothes people are wearing, their rough body shape remains the same for all human beings. Actually, there are many theories about this topic, pose or keypoint is used to describe the body of the person, and usually, we will use 14 poses. The adjacent two poses form into a skeleton. Many models use pose/skeleton such as Hourglass, OpenPose, CPM, and AlphaPose. To conclude, the person's specific shapes can

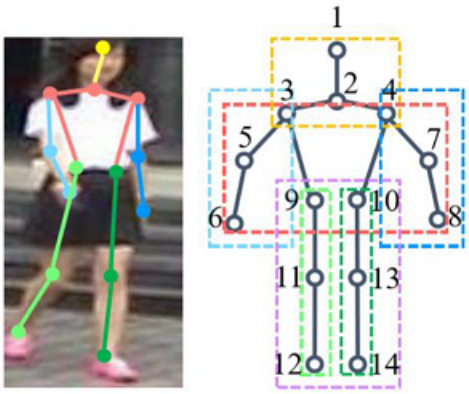



Fig. 4: Pose

be easily discerned in the edge or other feature maps, which retains the essential information while throwing away non-essential information.Designing suitable features are crucial to specific tasks. A good feature will capture a person's desired shape and information about how such a person is different from other entities, such as cars, trees, and buildings.

Some well-know features used are Haar-like features introduced by Viola and Jones, Histogram of Oriented Gradients(HOG) proposed by Dalal, Scale-Invariant Feature Transform(SIFT), Local Binary Patterns(LBP), and Speeded UP Robust Feature(SURF), Oriented FAST and Rotated BRIEF(ORB).

There are abundant feature extraction methods available. The goal is to decide whether the approach is more advantageous and how well the feature extraction can solve specific problems. Usually, a good feature should have these properties: Scale invariance, Brightness invariance and Rotational invariance.

For the scale invariance, we can build pyramids as shown in Fig 5. For the rotational invariance, SIFT and ORB have good performance and there are some variants of LBP that have rotational invariance.

Let's take a look at HOG first. The central concept of this approach is that the appearance of a pedestrian can be distinctively characterized by the distribution of the local intensity gradients and directions. Pixels in detection windows are grouped into different blocks, and each block is divided into four 8x8 cells, and the gradient of each cell is calculated, with the magnitude and direction, as in figure 6b.

Thereafter, store the magnitude of each cell-gradient voting into the appropriate bins such as nine bins, as in figure 7b, thus reducing the feature descriptor's dimensionality significantly.

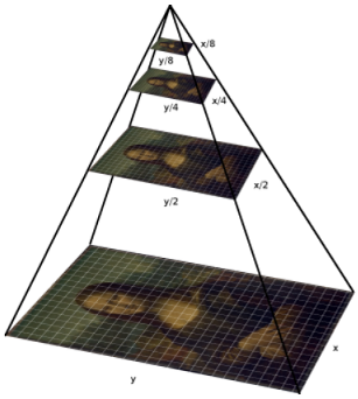
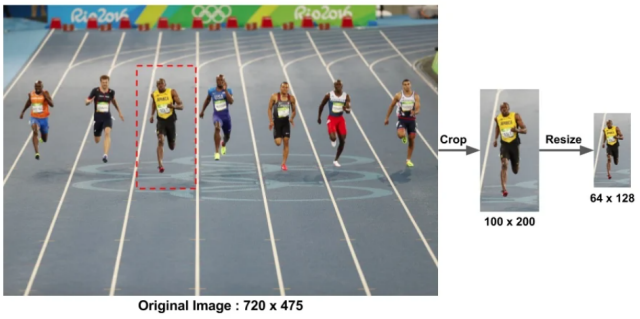


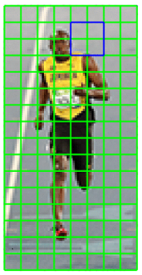

Fig. 5: Pyramid

With enough labeled training pedestrian images and negative samples, HOG features are collected from the process above. And these feature vectors are fed into classifiers such as SVM, MLP, Naïve Bayes, etc., to learn the model effectively.

In our experiments, we get different 64x128 image patches as samples, and then use a window that is the same size as the patches to detect whether a person in that patch or not. First, we need to divide the windows into 7x15 blocks, the block size is 16x16, and we calculate the block in horizontal and vertical directions. Furthermore, in order to detect small features, we again divide the blocks into 2x2 cells, the cells size is 8x8. As in Fig 6b, each block has four cells, and we



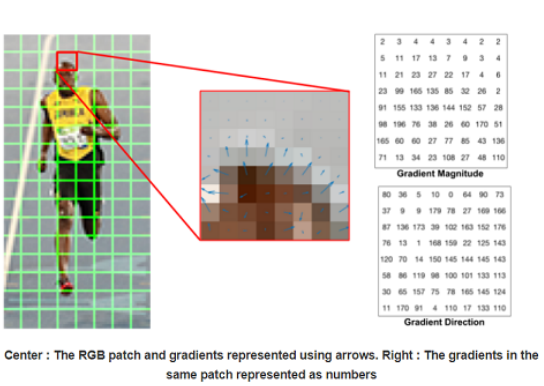

(a) Windows



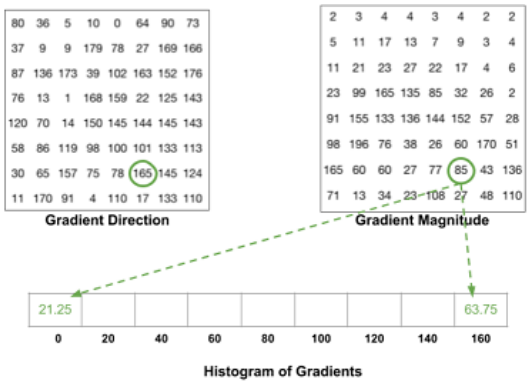
(b) Blocks

Fig. 6: Windows and Blocks

calculate the gradient magnitude and direction in each cell as in Fig 7. After we get the histogram of the gradient of each




(a) Computing Gradient




(b) Put Gradients into bins

Fig. 7: Compute gradients

cell, we need to normalize it to make it brightness invariant. Usually, a 16x16 block is normalized as a whole, the overall process as in Fig 8.

With enough labeled training pedestrian images and negative samples, HOG features are collected from the process above. And these feature vectors are fed into classifiers such as SVM, MLP, Naïve Bayes, etc., to train the model effectively.

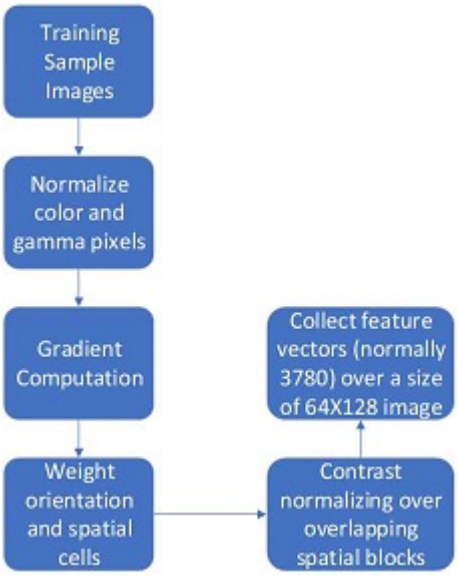



Fig. 8: Process overview

### *D. Training Models*

Five models will be used in the learning steps:

- K-Nearest Neighbor: need to learn which k is the most suitable, test the performance of different k by cross validation.
- Support Vector Machine: use linear kernel, need to set different parameters.
- Multilayer-Perception: need to decide how many hidden neurons.
- Decision Trees: need to set different parameters.
- Naïve Bayes

In addition, two ensemble learning methods will be included:

- Adaboost: with Decision Trees as base classifier.
- Bagging: with KNN as base classifier.

In this experiment, we will compare the performance for different models, calculate the accuracy and F1 score for each model.

With the INRIA datasets, which consists of 2416 images as positive training samples, actually, there are only 1208 images contain people, but each image is flipped to get more positive samples. There are 1218 images as negative training samples, 75 percent of them are used for training, 2436 images sampled from these images as negative training samples by random cropping and resizing. For the testing phase, we will use 1126 images as positive testing samples; specifically, there are only 563 images contains pedestrians. Each image is flipped to get more positive samples. 25 percent of the previous negative training samples will be used for testing, there will be 1218 negative samples for testing in total.

Additionally, first-time trained results will be evaluated, and all false-positive samples will be labeled as negative samples for the hard sample mining process. In theory, this process will significantly increase performance for classifiers.

One more step is considered in our experiment: Investigate the influence of PCA on the HOG features, the result comparison of the performance with dimension reduction, and the original samples will be included in the report.

The Haar-like features with cascading weak classifiers using the Adaboost is a successful practice in face detection. In this experiment, Haar-like features with ensemble learning will be applied as a comparison with HOG-feature-based approaches to discern whether it is viable for pedestrian detection. All performances with both feature-based detections to evaluate their effects, benefits, and shortcomings.

## VI. Experimentations

Two experiments was conducted based on HOG features and Haar-like features both generated from INRIA Person dataset, of which can be used to detect the pedestrian with

five classifiers (K-Nearest Neighbor, Multi-Layer Perceptron, Decision Tree, Support Vector Machine, Naïve Bayesian). In addition, one more experiment was conducted to explore the performance difference between using HOG features with SVM and original features with SVM on medium-sized dataset - CIFAR10. Last but not least, convolutional neural network was implemented as deep learning methods to train and test the CIFAR10 dataset, serving as a side-by side comparison.

The classic pipeline for pattern recognition task is as Fig 9: The very first step would be acquiring and exploring the

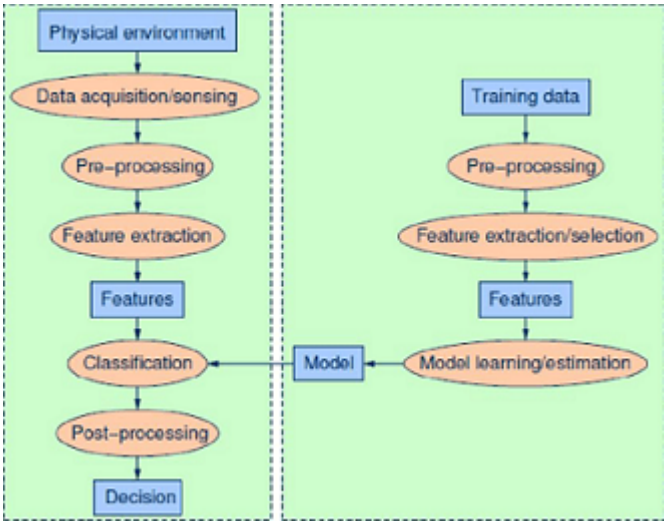



Fig. 9: Experiment Pipeline

dataset. If no suitable public dataset suitable for the task we would like to solve, we will need to collect the dataset by ourselves. For the images, they can be collected via free-licensed or permitted videos and online pictures, then labels for the images are added with third-party tools such as Label Image. In the pre-processing stage, it is essential to remove the noise from the data, and deploy cleaning and formatting steps for the data.

For the pedestrian detection problem, INRIA Person dataset is our first choice. After exploring the dataset, several invalid images are removed. Since all the positive and negative samples are clearly classified by the authors of the dataset, all positive samples (which have pedestrians inside) are labeled as 1 and negatives (which don't include any pedestrians) as -1. By extracting HOG features, gradient features, LUV features, and Haar-like features, and performing Principal Component analysis on HOG gradient features as decomposed data, all of these are converted into feature matrix as the input for the classifiers chose for this experiment.

### *A. Experiment - HOG Features*

Based on our experiment methodology, the bounding box size (detection window) is set to 64*128, and 3780 dimension HOG features are generated as a result. Furthermore, they are decomposed into 128 and 512 dimensions for performance comparison. Seven classifiers are chosen for training and testing - Support Vector Machine (linear), Naïve Bayes (Gaussian), Multilayer-Perceptron, Decision Tree and KNN as simple classifiers, Bagging (KNN) and Adaboost (Decision Tree) as ensemble learning classifiers. The complete result output for each classifier includes training and testing accuracy, F1 score, recall, precision, and confusion matrix. Among

| **Features** | HOG | | | | | | Haar | |
|---|---|---|---|---|---|---|---|---|
| **Demension** | 3780 | | 128 | | 512 | | 128 | |
| **Classifier** | TA | F1 | TA | F1 | TA | F1 | TA | F1 |
| SVM | 0.96 | 0.96 | 0.97 | 0.98 | 0.97 | 0.97 | 0.84 | 0.87 |
| Naïve Bayes | 0.90 | 0.91 | 0.90 | 0.90 | 0.77 | 0.77 | 0.83 | 0.86 |
| MLP | 0.98 | 0.98 | 0.98 | 0.98 | 0.97 | 0.97 | 0.76 | 0.82 |
| Decision Tree | 0.88 | 0.89 | 0.86 | 0.87 | 0.86 | 0.87 | 0.85 | 0.86 |
| KNN | 0.71 | 0.65 | 0.45 | 0.00 | 0.45 | 0.00 | 0.82 | 0.83 |
| Bagging-KNN | 0.70 | 0.62 | 0.45 | 0.00 | 0.45 | 0.00 | 0.84 | 0.85 |
| Adaboost | 0.98 | 0.98 | 0.96 | 0.97 | 0.96 | 0.96 | 0.94 | 0.95 |

TA = Test Accuracy
F1 = F1 Score on positive samples

Fig. 10: Experiment results

all the classifiers with 3780 dimension HOG features, MLP, Adaboost and SVM have best performance, with test accuracy and F1 score more than 95 percent.

To further compare the performance between the original and decomposed feature sets, the feature dimensions are decomposed by PCA from 3780 to 128 and 512.

Again using the same classifiers with the same parameters as for the original feature sets. MLP, Adaboost and SVM still attain better performance than the others.

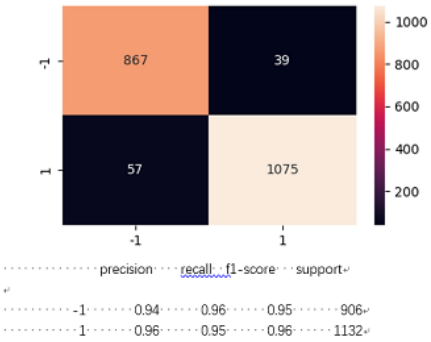
Interestingly, for Naïve Bayes, and KNN classifiers, the performance dropped significantly after applying the dimension deduction. To figure out the reason behind the result, we compared the confusion matrix between the SVM and KNN as in Fig 11. When HOG features was decomposed from 3780 to 128, the precision and recall for SVM didn't drop, even increased.

In contrast, the confusion matrix of KNN below indicates that all the positive samples are falsely classified as negative samples. The negative accuracy along contributed to the overall accuracy. As shown in Fig 12

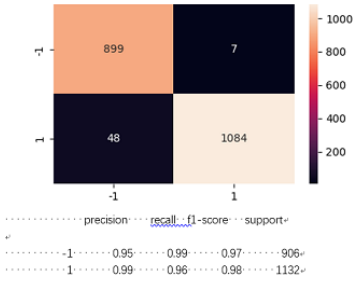
Since KNN calculates the feature distances between samples, and HOG features represents pedestrian shape directions which are too close for KNN to differentiate them from each class. On the other hand SVM which finds boundaries with margin based on sparse data (support vectors) is more robust against close distances.

Ensemble learning such as bagging is implemented to improve the performance for KNN; And it didn't yield any improvement or enhancement. Before experiment started, cross-validation was tested for K, which in turn would be 3, so that KNN could attain the highest accuracy.

For the Adaboost, decision tree was chosen as the base classifiers. In comparison for a single decision tree, Adaboost notably boosted performance for both original and decomposed data (all by 10 percent).

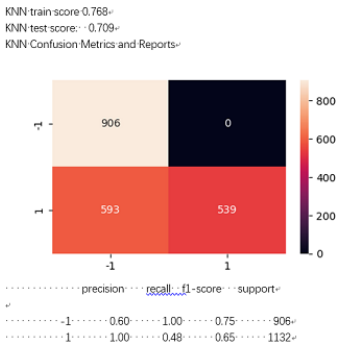



(a) SVM On Original Gradient

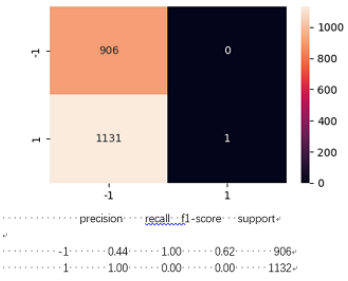



(b) SVM on Decomposed 128 Features

Fig. 11: SVM Performance




(a) KNN On Original Gradient



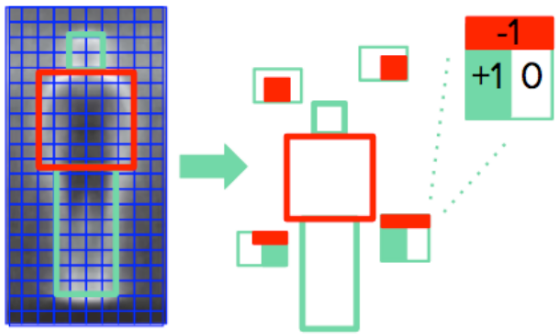

(b) KNN on Decomposed 128 Features

Fig. 12: KNN Performance

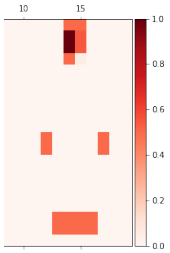
The detailed experiment report can be found in the classifier report document.

### *B. Experiment - Informed Haar-like Features*

**??** proposed informed Haar-like features combined with HOG, LUV, gradient features to enhance the performance for pedestrian detection. They discovered that the head and shoulder features were applicable to detect a person. With well-captured Haar features,more weights were assigned to the head and shoulder region. For this experiment we use the same Haar-like features and multiply them with HOG, LUV, and gradient features to generate the informed features as the input feature vectors. As in table from the Fig 13, with




(a) Informed Haar




(b) Feature Weights

Fig. 13: Haar Features Combined with HOG, cited from Informed Haar-like features for Pedestrian Detection

these informed Haar-like features, which only include the top 128 weighted features, the performance of KNN is improved conspicuously. For the Haar features, the Adaboost, Decision Tree and SVM resulted in better performance. It is reasonable since the Haar templates are all tiny features. Decision Tree and AdaBoost, which can take full advantage of these features, can achieve superior results than simply HOG features in the same context.

After training and saving models of each classifiers, we proceed to the prediction stage. NMS algorithms were implemented to restrict the number of bounding boxes.

We used slide windows to detect if a person was inside the detection range by sending the features in that region to the classifiers and get the predicted result. If a person was detected, the particular region was added to a list.

After applying the NMS, using the informed Haar-like features, different parameters were tested which indicated that with the scaling factor 1.1-1.2, the threshold for NMS less 0.1, we could attain sustainable detection results. In Fig 14 shows one of the result. Still there are rooms for improvement and will require further experiment.

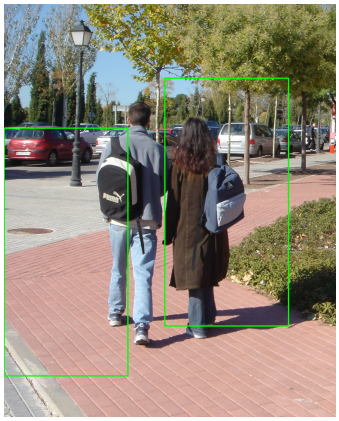


Fig. 14: Detection Result

### C. Experiment - HOG Features in Object Detection

From the first two experiments, we can conclude that HOG features are very powerful in representing principal characteristics of an object silhouette in an image. So we would like to compare the performance of traditional image classification problems both with and without HOG features.

Here we choose a medium size dataset - CIFAR 10, and use SVM as our base classifier.

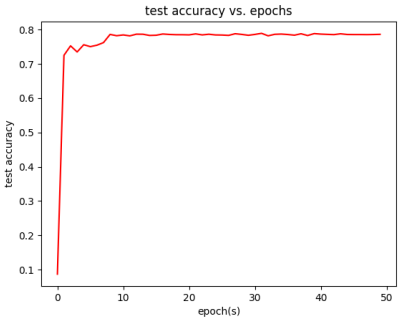
There are 50,000 training samples and 10,000 testing samples in a total 10 classes. Training samples are decomposed by PCA before sending into SVM to train. The test accuracy was 30.14 percent. With HOG features, SVM with the same parameters could achieve the test accuracy of 46 percent, which improved remarkably compared to the raw features.

Although the HOG features were first proposed for pedestrian detection, it also can be used in other computer vision domains such as object detection.

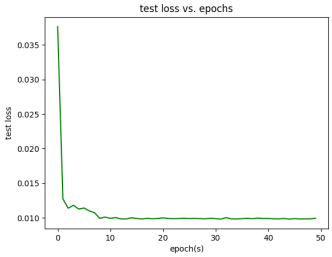
### D. Experiment - SVM vs. Convolutional Neural Network

The HOG+ SVM method has reached 46 percent test accuracy on the CIFAR10 dataset, what about the performance of deep learning methods? The experiment on SVM vs. deep learning methods was conducted in order to satisfy our curiosity.

Here Resnet50 pre-trained model was our choice of CNN. First all the weights were frozen for the feature layers and the last fully connected layer could be trained. Then The classes of output were converted from 1000 to 10. Finally initiate PyTorch version Resnet50 for training on the CIFAR10 training samples. The total epochs was 50. Since the model is pre-trained by the ImageNet, within 1-2 epochs the test accuracy had already reached around 75 percent (as shown in Fig 15). Eventually, Resnet50 had the average test accuracy of 78 percent, which was better performed compared to HOG + SVM.




(a) Test Accuracy




(b) Test Loss

Fig. 15: Resnet50 Test Results

## VII. Discussion and Conclusion

In these experiments, we implement various classification methods from both pattern recognition and machine learning on 2 ubiquitously applied feature extraction measures, such as K-Nearest Neighbor, Multilayer Perceptron, Decision Tree, Support Vector Machine, Naïve Bayes to compare their performance in resolving pedestrian detection problems. In addition, ensemble learning, such as Adaboost and Bagging, were included as side-by-side comparison for different feature extraction process. The features do carry significant weight in representing key characteristics for an object.

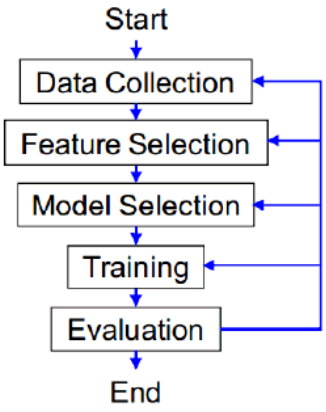



Fig. 16: Process

We also came into conclusion that, without deep learning methods, raw images need to be well explored and features extracted before sending them into classifiers in order to attain better results.

Data pre-processing and result evaluation stages are the major time consumer for us in above experiments. And we believe there are ways for improvement awaiting to be discovered.

Image pre-processing methods are equally important as powerful classifiers. Although deep learning has dominated various areas of the computer vision field, the traditional methods still hold a firm position since they are applicable, effective and efficient even on small embedded devices without GPUs.

## REFERENCES


[1] Lowe, D.G. "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision 60, 91–110 (2004).

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1.

[3] P. Dollar, C. Wojek, B. Schiele and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 4, pp. 743-761, April 2012.

[4] M. Enzweiler and D. M. Gavrila, "Monocular Pedestrian Detection: Survey and Experiments," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 12, pp. 2179-2195, Dec. 2009.

[5] X. Wang, T. X. Han and S. Yan, "An HOG-LBP human detector with partial occlusion handling," 2009 IEEE 12th International Conference on Computer Vision, Kyoto, 2009, pp. 32-39.

[6] S. Walk, N. Majer, K. Schindler and B. Schiele, "New features and insights for pedestrian detection," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 1030-1037.

[7] Jiale Cao, Yanwei Pang, Jin Xie, Fahad Shahbaz Khan, Ling Shao, "From Handcrafted to Deep Features for Pedestrian Detection: A Survey,",1,Oct,2020 Cornell University.

[8] S. Walk, N. Majer, K. Schindler and B. Schiele, "New features and insights for pedestrian detection," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 1030-1037.

[9] Yoav Freund, Robert E Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," Journal of Computer and System Sciences, Volume 55, Issue 1, 1997, Pages 119-139, ISSN 0022-0000.

[10] P Dollár, Z Tu, P. Perona, S Belongie, "Integral channel features.", (2009)91-1

[11] P. Dollár, R. Appel, S. Belongie and P. Perona, "Fast Feature Pyramids for Object Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 8, pp. 1532-1545, Aug. 2014.

[12] Benenson, Rodrigo, et al. "Seeking the strongest rigid detector." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013. Conference on Computer Vision and Pattern Recognition, 2013. 4, 5, 6

[13] Papageorgiou, Constantine, Tomaso Poggio. "A trainable system for object detection." International journal of computer vision 38.1 (2000): 15-33.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627-1645, Sept. 2010.

[15] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.

[16] Zhang, Shanshan, Rodrigo Benenson, and Bernt Schiele. "Citypersons: A diverse dataset for pedestrian detection." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[17] Z. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection With Deep Learning: A Review," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, Nov. 2019.

[18] Zhang L, Lin L, Liang X, He K, "Is faster R-CNN doing well for pedestrian detection?" European conference on computer vision. Springer, Cham, 2016: 443-457.

[19] N. J. Karthika and S. Chandran, "Recent Developments in Pedestrian Detection Using Deep Learning," 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2019, pp. 353-358.

[20] P. Dollar, C. Wojek, B. Schiele and P. Perona, "Pedestrian detection: A benchmark," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 304-311.

[21] Tabitha S. Combs, Laura S. Sandt, Michael P. Clamann, Noreen C. McDonald"Automated Vehicles and Pedestrian Safety: Exploring the Promise and Limits of Pedestrian Detection,"American Journal of Preventive Medicine, Volume 56, Issue 1, 2019, Pages 1-7, ISSN 0749-3797.

[22] Fengliang Xu, Xia Liu and K. Fujimura, "Pedestrian detection and tracking with night vision," in IEEE Transactions on Intelligent Transportation Systems, vol. 6, no. 1, pp. 63-71, March 2005.

[23] Zhang, Shanshan, Christian Bauckhage, and Armin B. Cremers. "Informed haar-like features improve pedestrian detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.