Warsaw University of Technology

Image and Speech Recognition (Project Report)

Topic: Human Detection using HOG

Course Title: 103A-CTCSN-MSA-EIASR

Student name 1: Nityanand Waingankar

Student name 2: Cakston Calvin

Supervisor name: Maciej Stefańczyk

Table of content

1. Introduction

Human detection is a challenging problem due to variations in appearance, pose, lighting conditions, and background clutter. Addressing these challenges involves the utilisation of sophisticated techniques, and one notable approach is the Histogram of Oriented Gradient (HOG) method. This method focuses on analyzing the distribution of colour and intensity gradients in different parts of an image, allowing for the creation of distinctive patterns associated with human shapes.

Understanding human detection involves exploring how computers can interpret visual cues, such as edges, shapes, and contrasts, to make accurate assessments about the presence of individuals in an image. The ultimate goal is to design algorithms that can perform this task reliably in diverse real-world scenarios, ensuring robustness to changes in lighting, viewpoint, and other environmental factors.

As we delve into the realm of human detection in image processing, we will explore various methodologies, including the application of machine learning techniques like Support Vector Machines (SVM), and consider how advancements in this field contribute to broader applications such as video surveillance, autonomous vehicles, and human-robot interaction. The significance of these advancements lies in their potential to enhance the capabilities of computer vision systems, making them more adept at understanding and responding to the presence of humans in visual data.

2. Project Objectives and Proposed Solution Overview

This method uses a technique called Histogram of Oriented Gradient (HOG) to find humans in pictures. It looks at the direction of colour changes in small sections of the image and creates a kind of map. This map helps to understand the shapes and appearance of objects, like people, without knowing exactly where they are.

The process involves splitting the image into small parts ("cells") and creating a map of color changes in each part. These maps are then combined to represent the whole image. To make sure it works well in different lighting, the method adjusts the maps by looking at larger areas ("blocks") and normalizing the colors in each smaller part.

To find humans, the method places these maps in a grid over the image and uses a computer program to decide if there's a person based on the combined information. This idea comes from a method called Scale Invariant Feature Transformation (SIFT). Compared to other methods, this one is better at finding people and has fewer mistakes.

Using the HOG method has benefits. It helps understand the shapes in pictures while being flexible to changes in how things look. For finding people, it works best when looking at the whole picture in big chunks, paying close attention to the direction of colors, and adjusting for different lighting. This way, it can find people even if they're moving or the lighting changes.

3. Selected Algorithms (In Pseudocode)

1. Input: Image containing a scene with possible humans
2. Define Parameters:
   - Cell size
   - Block size
   - Number of orientations in the histogram (bins)
   - Threshold for detection
3. Preprocess Image:
   - Convert the image to grayscale
   - Apply any necessary preprocessing steps (e.g., normalisation)
4. Compute HOG Features:
   - Define a grid of cells over the image
   - For each cell:
     - Compute gradient magnitude and orientation
     - Construct a histogram of orientations for the cell
5. Block Normalisation:
   - Define a grid of blocks over the cells
   - For each block:
     - Normalise the histograms of the cells within the block
     - Concatenate the normalised histograms to form a block descriptor
6. Sliding Window Detection:
   - Define a sliding window that moves over the image
   - For each window position:
     - Extract HOG features within the window
     - We will be implementing Linear Classifier and will Use a trained classifier (e.g., SVM) to compare output with the implemented classifier
     - If the response is above a predefined threshold, mark the window as containing a human
7. Post-process Detected Windows:
   - Apply non-maximum suppression to remove overlapping detections
   - Output the final set of detected human bounding boxes
8. Output: Bounding boxes indicating the locations of detected humans in the image

* The parts highlighted will be implemented from scratch

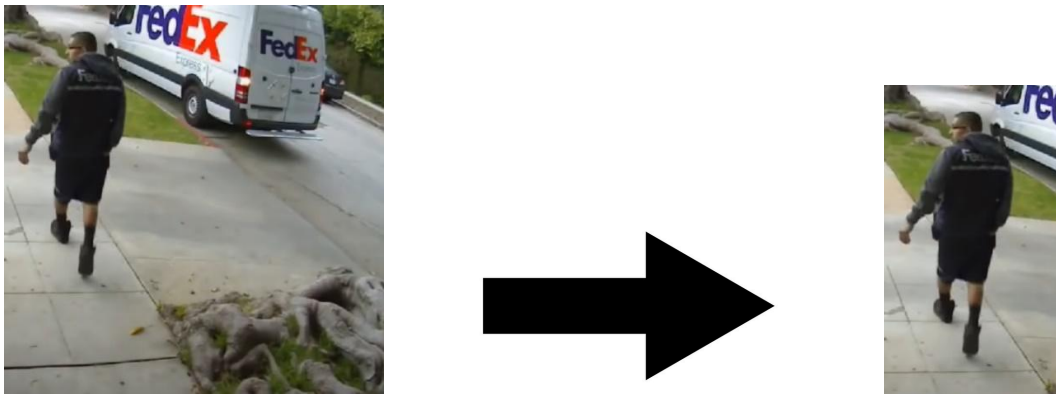
4. Methodology

4.1 Data Collection and Preprocessing

We have used a Human Detection Dataset of CCTV footage of humans by KONSTANTIN VERNER on Kaggle.

This dataset contains images of humans. It contains both indoor as well as outdoor images.These images are obtained from CCTV videos, open indoor images dataset and CCTV images.

There are 357 images without humans and 559 images with humans in them, totaling 916 images.We modified the images to focus solely on the human subjects, ensuring improved training data quality.



For training the classifiers, we compiled a dataset consisting of 379 images containing humans and 362 images without humans.

4.2 HOG Feature Extraction

Steps for computing HOG features

1. Choose the target image for HOG feature extraction. Resize the image to dimensions of 128x64 pixels, following the approach outlined in the paper titled "Histograms of Oriented Gradients for Human Detection" by Navneet Dalal and Bill Triggs. The authors recommended these specific dimensions, as their primary focus was on achieving improved results in the field of pedestrian detection.

2. The image's gradient is computed by combining magnitude and angle information. When working with a 3x3 pixel block, the initial step involves calculating Gx and Gy for each pixel. These Gx and Gy values are determined using the provided formulas for each pixel value.

$$G_x(r,c) = I(r,c+1) - I(r,c-1) \quad G_y(r,c) = I(r-1,c) - I(r+1,c)$$

Fig: In the context provided, "r" represents rows, and "c" represents columns.

Following the computation of Gx and Gy, the magnitude and angle for each pixel are determined using the provided formulas below.

$$Magnitude(\mu) = \sqrt{G_x^2 + G_y^2} \quad Angle(\theta) = |\tan^{-1}(G_y/G_x)$$

3. Upon acquiring the gradient information for each pixel, the gradient matrices, including magnitude and angle, are partitioned into 8x8 cells, forming distinct blocks. Within each block, a 9-point histogram is computed. This type of histogram involves nine bins, each covering an angle range of 20 degrees. Figure 8 illustrates a 9-bin histogram with values assigned through calculations. Each of these 9-point histograms can be represented graphically, displaying bins that indicate the intensity of the gradient within their respective angle ranges. Given that a block comprises 64 different values of magnitude and gradient, calculations are carried out for all these values. Since 9-point histograms are utilised, the calculations involve processing each of the 9 bins in the histogram. :

$$Number\ of\ bins = 9(ranging\ from\ 0^\circ\ to\ 180^\circ)$$
$$Step\ size(\Delta\theta) = 180^\circ\ /\ Number\ of\ bins = 20^\circ$$

Each Jth bin, bin will have boundaries from : $[\Delta\theta \cdot j, \Delta\theta \cdot (j+1)]$

Value of the centre of each bin will be $C_j = \Delta\theta(j + 0.5)$

Value									
Bins	0	20	40	60	80	100	120	140	160

Illustration of a 9-bin histogram. This individual histogram is distinct for an 8x8 block consisting of 64 cells. All 64 cells contribute their Vj and Vj+1 values to the jth and (j+1)th indices of the array, respectively.

4. For every cell within a block, the process begins with computing the jth bin, followed by determining the values assigned to the jth and (j+1)th bins. These values are determined using the following formulas:

$$j = \lfloor \left(\frac{\theta}{\Delta\theta} - \frac{1}{2} \right) \rfloor$$

$$V_j = \mu \cdot \left[\frac{\theta}{\Delta\theta} - \frac{1}{2} \right]$$

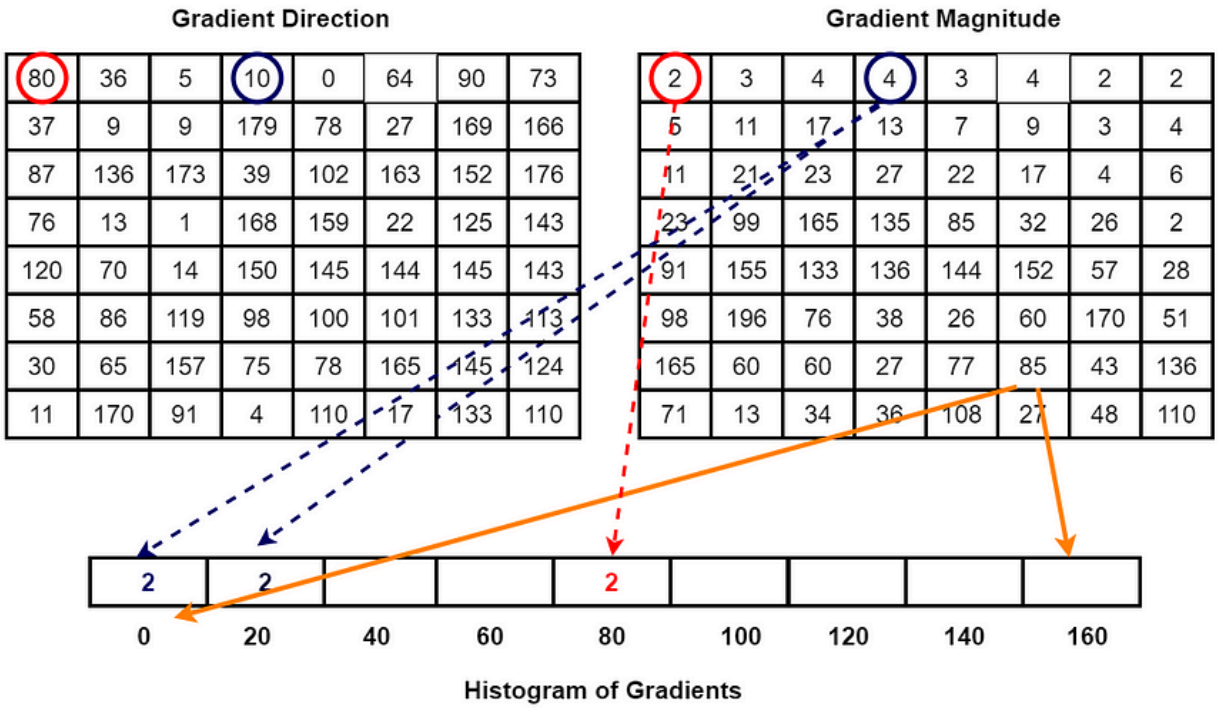
$$V_{j+1} = \mu \cdot \left[\frac{\theta - C_j}{\Delta\theta} \right]$$

5. A bin for a block is represented by an array, and the values of Vj and Vj+1 are added to the array at the indices corresponding to the jth and (j+1)th bins calculated for each pixel.

6.The matrix resulting from the aforementioned computations will be in the shape of 16x8x9.

7. After completing the histogram computation for all blocks, four blocks are combined from the 9-point histogram matrix to create a new block with dimensions 2x2. This combination is performed with an overlapping approach using a stride of 8 pixels. For all four cells within a block, the 9-point histograms for each constituent cell are concatenated to generate a 36-feature vector.

$$f_{bi} = [b_1, b_2, b_3, ..., b_{36}]$$



8. The values of fb for each block undergo normalisation through the L2 norm. :

$$f_{bi} \leftarrow \frac{f_{bi}}{\sqrt{\|f_{bi}\|^2 + \varepsilon}}$$

9.To perform normalisation, the value of k is initially computed using the following formula:

$$k = \sqrt{b_1^2 + b_2^2 + b_3^2 + ... + b_{36}^2}$$

$$f_{bi} = \left[\left(\frac{b_1}{k}\right), \left(\frac{b_2}{k}\right), \left(\frac{b_3}{k}\right), ..., \left(\frac{b_{36}}{k}\right) \right]$$

10. Normalisation is applied to mitigate the impact of contrast variations among images depicting the same object. From each block, a 36-point feature vector is gathered. With 7 blocks in the horizontal direction and 15 blocks in the vertical direction, the total length of HOG features is calculated as 7 x 15 x 36, resulting in a total of 3780 HOG features for the selected image.

4.3 Training the Classifier

In this section, we delve into the details of training the classifier for our human detection system. The choice of classifier, the training process, and the parameter tuning are crucial aspects that significantly influence the performance of the model.

4.3.1 Choice of Classifier

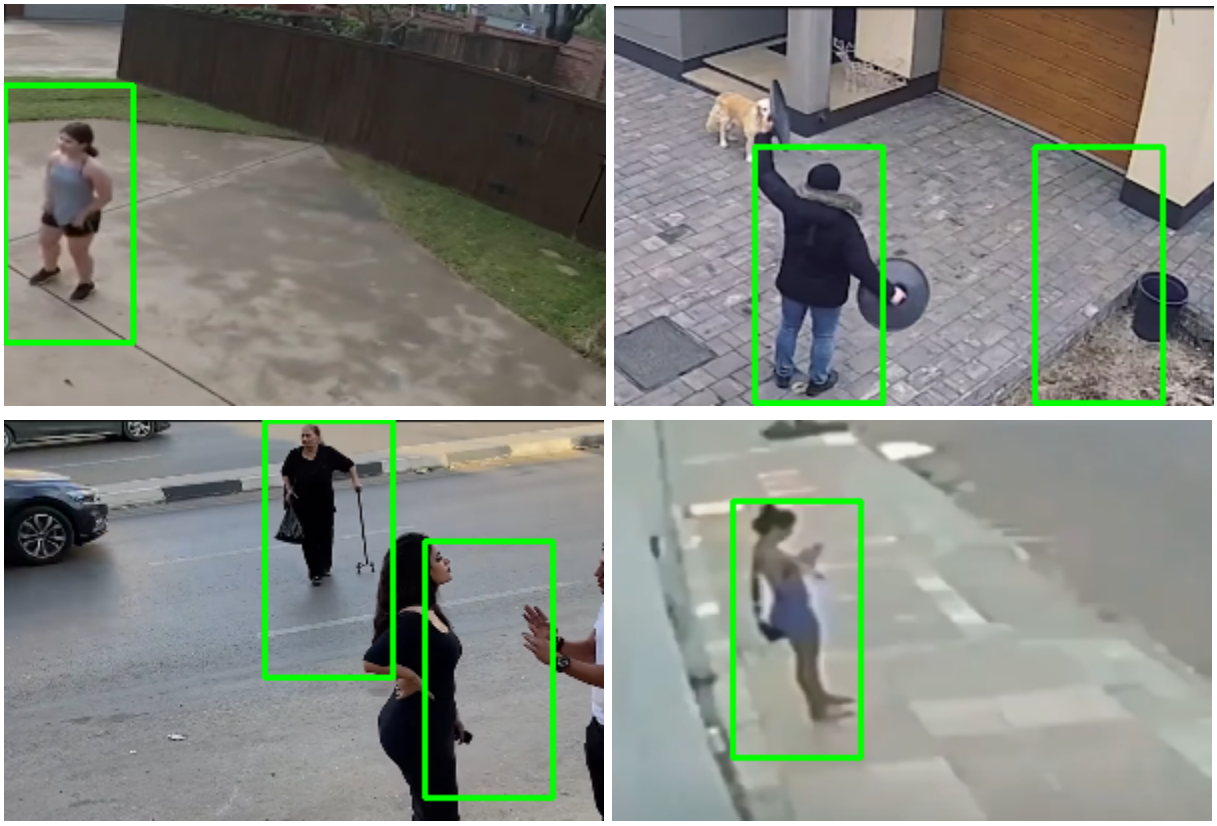
For our human detection project, we opted for the Support Vector Machine (SVM) as our classifier. SVMs are renowned for their effectiveness in binary classification tasks and are well-suited for image-based recognition tasks, making them a suitable choice for our human detection system.

4.3.2 Training Process

The training process involved feeding the HOG features extracted from the preprocessed dataset into the SVM. The dataset was split into training and validation sets to facilitate model training and evaluation. The SVM was trained to learn the patterns and characteristics indicative of human presence in images.

	Precision	**Recall**	**f1-score**	**Support**
0	0.95	0.67	0.78	30
1	0.87	0.99	0.93	69
Accuracy			0.89	99
Macro avg	0.91	0.83	0.85	99
Weighted avg	0.90	0.89	0.88	99

Here are some of the outputs for sample images that we ran through this detector.



From the output images, it is evident that the classifier successfully identifies humans in the images. However, there are instances where false positive detections occur, indicating humans in regions where there are none. To address such errors, a threshold for positive predictions was introduced, specifically when the decision function was greater than 75%. This measure aimed to minimize the occurrences of false positives, although it did not entirely eliminate them. Despite the threshold, some instances of misclassifications persisted.

The confusion matrix for the human detection model using the SVM classifier for a sample of 20 images is as follows.

	Sample Images
Custom HOG/SVM Detector	Detected = 22
	Not Detected = 1
	Ghost Detected = 23

In a set of 20 images, there were 2 images with multiple human beings. Accurate human detection occurred in all instances while also ghost detection occurred in 16 images.

5. Creating the Classifier

Our implementation features a custom Perceptron class designed for human detection. The Perceptron class includes methods for training, prediction, and model persistence. The classifier iteratively adjusts its weights based on misclassifications during the training phase, aiming to learn discriminative patterns in the HOG feature space.

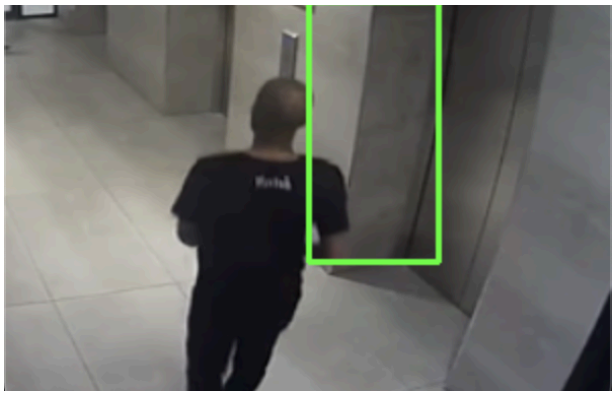
5.1 Training

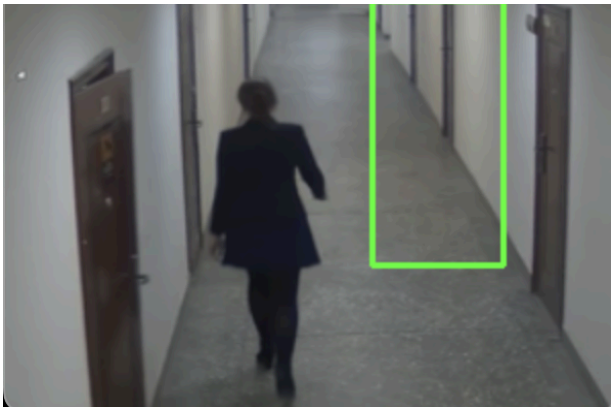
Positive and negative samples underwent HOG feature extraction, and corresponding labels were assigned (1 for positive, 0 for negative). The Perceptron class was then trained on this labelled dataset to learn the decision boundary between human and non-human patterns.

	Precision	**Recall**	**f1-score**	**Support**
0	0.93	0.70	0.79	29
1	0.86	0.98	0.91	70
Accuracy	0.88			99
Macro avg	0.89	0.84	0.85	99
Weighted avg	0.88	0.88	0.87	99

5.3 Testing

We tested the trained classifier on a sample image using a sliding window approach. The sliding window traversed the image, and at each position, HOG features were extracted and fed into the trained perceptron for prediction. Non-maximum suppression (NMS) was applied to eliminate redundant detections, and bounding boxes were drawn around the identified human regions.



While the initial outcomes are encouraging, it's important to note that the classifier may encounter occasional errors in its predictions. To address this, further investigations into parameter adjustments, dataset enrichment, and the exploration of alternative feature extraction methods or advanced classifiers could be pursued. This iterative refinement process is integral for minimizing prediction errors and enhancing the overall robustness of the human detection system.

Total Sample: 20

	Sample Images
Custom HOG /Perceptron Detector	Detected = 13
	Not Detected = 7
	Ghost Detected = 8

In a set of 20 images, accurate human detection occurred in 13 instances, while ghost detection occurred in 8 images. Among the latter, 7 images contained humans, and 1 image featured multiple humans.

6. Conclusion

In conclusion, the implementation of Human Detection using Histogram of Oriented Gradients (HOG) has demonstrated promising results and showcased the effectiveness of this technique in computer vision applications. Through the extraction of distinctive features from images and the training of a classifier, the system has shown its capability to detect humans in diverse scenarios.

The key findings from this project include:

Accuracy and Performance: The system's accuracy, as measured by evaluation metrics, indicates its success in distinguishing between human and non-human objects. The performance achieved in terms of precision, recall, and overall accuracy provides valuable insights into the reliability of the implemented approach.

Challenges and Limitations: During implementation, certain challenges were encountered, such as variations in lighting conditions, diverse poses, and occlusions. These challenges highlight the limitations of the current system and suggest areas for improvement.

Robustness and Generalization: The system's robustness to handle variations in real-world scenarios and its ability to generalize well to unseen data contribute to its potential in practical applications. However, further testing on larger datasets and in diverse environments may be necessary to validate its generalizability.

Future Directions: To enhance the system, future work could focus on incorporating additional features, exploring advanced classifiers, and addressing specific challenges encountered during testing. Additionally, fine-tuning the parameters and exploring ensemble methods could contribute to improved performance.

In summary, this project has laid the foundation for human detection using HOG, providing valuable insights into its strengths and limitations. The lessons learned and findings presented here serve as a basis for future research and development in the field of computer vision and object detection. The successful implementation of HOG for human detection opens up avenues for further exploration and optimization, contributing to advancements in this critical area of research.

* Nityanand Waingankar handled responsibilities related to the HOG feature extractor, while Calvin Cakston managed tasks associated with the Perceptron classifier. The project report reflects an equal contribution from both participants.

7. Reference

1. Histograms of Oriented Gradients for Human Detection: By Navneet Dalal and Bill Triggs

   https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

2. HOG (Histogram of Oriented Gradients): An Overview

   https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

3. Human Detection Dataset

   https://www.kaggle.com/datasets/constantinwerner/human-detection-dataset

4. Evolution Of Object Detection Networks by Cogneethi

   https://youtube.com/playlist?list=PL1GQaVhO4f_jLxOokW7CS5kY_J1t1T17S&si=3QQq4pUnxrao8Kts