



**Master thesis of Paris-Saclay University**

**Thesis supervision**

**Committee**

Arnaud BOURNEL Chairman

*Université Paris-Saclay*

Pierre DUHAMEL Reporter

*CNRS, Laboratory of Signals and Systems*

NGUYEN Linh Trung Examiner

*VNU University of Engineering and Technology*

BUI Duy Hieu Examiner

*VNU Information Technology Institute*

Erwan LIBESSARTExaminer

*CentraleSupélec*

BUI Duy Hieu Thesis supervisor

*VNU Information Technology Institute*

TRAN Thi Thuy Quynh Co-supervisor of the thesis

*VNU University of Engineering and Technology*

Erwan LIBESSARTCo-supervisor of the thesis

*CentraleSupélec*

DEVELOPING A REAL-TIME OBJECT DETECTION SYSTEM ON FPGA

Specialization: M2 Communication and Data Engineering

**Master student: NGUYEN Trung Kien**

# ABSTRACT

# AUTHORSHIP

|  |  |  |
| --- | --- | --- |
|  | | *Hanoi*, January 22nd, 2024 |
|  | | Author |
|  | |  |
|  |

# ACKNOWLEDGEMENT

Contents

[ABSTRACT i](#_Toc201044025)

[AUTHORSHIP ii](#_Toc201044026)

[ACKNOWLEDGEMENT iii](#_Toc201044027)

[List of Abbreviations vi](#_Toc201044028)

[List of Figures viii](#_Toc201044029)

[List of Tables ix](#_Toc201044030)

[Introduction x](#_Toc201044031)

[Chapter 1. Real-time Object Detection System 1](#_Toc201044032)

[1.1. Introduction to the Internet of Things (IoT) and Its Applications for Fire Monitoring Alert Systems 1](#_Toc201044033)

[1.2. Current situation and Limitations of the monitoring fire alert system 2](#_Toc201044034)

[1.3. Related Works 3](#_Toc201044035)

[1.4. Conclusions 6](#_Toc201044036)

[Chapter 2. Proposed Real-time Object Detection System 7](#_Toc201044037)

[2.1. Proposed System using Zifian Estimation Technique and HOG SVM Algorithm 7](#_Toc201044038)

[2.1.1. Dataflow 8](#_Toc201044039)

[2.1.2. Proposed System 10](#_Toc201044040)

[2.2. Lightweight Motion Detection Algorithm used on a stationary camera 11](#_Toc201044041)

[2.3. Object Detection using the HOG SVM Algorithm 13](#_Toc201044042)

[2.3.1. State of the Art 13](#_Toc201044043)

[2.3.2. Object Detection Block 16](#_Toc201044044)

[2.4. Conclusions 22](#_Toc201044045)

[Chapter 3. Implementation and Evaluations 23](#_Toc201044046)

[3.1. Experiment setup environment 23](#_Toc201044047)

[3.1.1. Mean Average Precision 24](#_Toc201044048)

[3.1.2. Frame rate 26](#_Toc201044049)

[3.2. Experimental results 26](#_Toc201044050)

[3.2.1. Input: PETS09-S2L1 [39] 26](#_Toc201044051)

[3.2.2. Input: TUD-Stadtmitte [40] 28](#_Toc201044052)

[3.2.3. Input: TUD-Campus [41] 29](#_Toc201044053)

[3.3. Conclusions 31](#_Toc201044054)

[Conclusions and Perspective 32](#_Toc201044055)

[References 34](#_Toc201044056)

# List of Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Definition |
| 2D | Two Dimensional |
| ADAS | Advanced Driver Assistance Systems |
| AI | Artificial Intelligence |
| AMD | Advanced Micro Devices |
| mAP | Mean Average Precision |
| ASIC | Application Specific Integrated Circuit |
| AVC | Advanced Video Coding |
| CBAM | Convolutional Block Attention Module |
| CNN | Convolutional Neural Network |
| COCO | Common Objects in Context |
| CPU | Central Processing Unit |
| DMA | Direct Memory Access |
| FPGA | Field Programmable Gate Array |
| FPS | Frames Per Second |
| GB | Gigabyte |
| GE | Gate equivalents |
| GPU | Graphics Processing Unit |
| HD | High Definition |
| HOG | Histogram of Oriented Gradients |
| LTS | Long Term Support |
| MJPEG | Motion JPEG |
| MOT15 | Multiple Object Tracking Benchmark 2015 |
| MOTC | Multiple Object Tracking Challenge |
| MPSoC | MultiProcessor System On Chip |
| NMS | Non-Maximum Suppression |
| OS | Operating System |
| PASCAL | Pattern Analysis Statistical Modelling and Computational Learning |
| PETS09 | Performance Evaluation of Tracking and Surveillance 2009 |
| PTZ | Pan-Tilt-Zoom |
| RAM | Random Access Memory |
| RCNN | Regional Convolutional Neural Network |
| RGB | Red Green Blue |
| SIFT | Scale-Invariant Feature Transform |
| SPP | Spatial Pyramid Pooling |
| SRAM | Static Random Access Memory |
| SSD | Single Shot Detector |
| SVM | Support Vector Machine |
| TSMC | Taiwan Semiconductor Manufacturing Company |
| UHD | Ultra High Definition |
| VNU | Vietnam National University |
| VOC | Visual Object Classes |
| YOLO | You Only Look Once |
| ZCU106 | Zynq UltraScale+ ZCU106 Evaluation Kit |

# List of Figures

[Figure 1. IoT system with interconnected smart device. 2](#_Toc200984189)

# List of Tables

[Table 1 - Sine and Cosine quantized value. 19](#_Toc157527576)

[Table 2 - Truth Table Confusion Matrix. 24](#_Toc157527577)

# Introduction

# Real-time Object Detection System

This chapter provides an overview of the application of Internet of Things (IoT) technology in monitoring and alerting fire detection systems, primarily in the context of increasingly serious fire problems that require critical attention. It begins with an introduction to IoT and the importance of its role in improving monitoring skills and its value in fire situations. In addition, this chapter also focuses on the main objective of the thesis: to explore and employ the full potential of IoT-connected hardware devices in both IoT applications and real-world scenarios. This chapter is organized into three sections. Section 1.1 provides an overview of the Internet of Things (IoT) and its applications in fire monitoring systems. [*Section 1.2*](#Section1_2) discusses the current situation and limitations of existing fire monitoring systems. [*Section 1.3*](#Section1_3) presents modern fire detection methods using current technologies, along with their limitations.

## Introduction to the Internet of Things (IoT) and Its Applications for Fire Monitoring Alert Systems

In recent years, fire accidents have become increasingly dangerous, causing critical damage to both human life and property. Traditional fire monitoring systems, which often rely on manual checks or local alerts, are limited in their ability to respond quickly and effectively, especially in complex or remote environments. To handle this issue, we need smart and effective solutions to improve it in time.

The Internet of Things (IoT) is one exciting technical development that resolves these issues. As its name suggests, IoT refers to a network of smart devices that can communicate and connect with each other, exchanging data. These devices can collect, transmit, and process data in real-time, thereby enabling automation and making personal decisions.

Unlike standalone devices that work independently, a true IoT system is a scalable and integrated system where multiple devices work together for a common goal. In the context of fire detection, this goal is always monitoring environmental conditions, identifying early signs of fire, and sending real-time alerts, even when no one is present at that location.

By providing features such as remote monitoring, automated notifications, and log data, IoT-based systems significantly enhance responsiveness and overall safety. Therefore, using IoT for fire monitoring systems is both useful and essential for enhancing early detection and minimizing damage.



Figure . An IoT system with interconnected smart devices.

## Current situation and Limitations of the monitoring fire alert system

Fire accidents are now common and can occur in various types of environments, ranging from crowded residential areas to industrial zones, forests, etc.... In an industrial environment, the use of flammable materials, high electricity consumption, and poor ventilation are common reasons that increase the risk of fire. In a residential area, the complex and overloaded electrical network increases the chance of short circuits and electrical fires. In a forest environment, rising global temperatures make it more flammable, and also human activities, such as camping or purposeful burning of the forest, increase the risk of fire. As mentioned, the fire can occur anywhere, any time, whether in crowded urban areas or isolated natural areas.

Therefore, deploying fire monitoring systems with sensors and data-collecting modules in large areas is expensive and challenging. Traditional systems, which rely on wired connections or require continuous human monitoring, are not realistic, especially in large areas or remote and harsh environments. These systems will face problems such as processing complex data and false fire alerts. This is really a big challenge for the fire warning system. The solution using IoT with sensor data can help provide more accurate and useful results.

Currently, traditional fire monitoring and alert systems are still widely used and have not been fully replaced by IoT-based systems. Although some improvements have been made, many issues and limitations still exist, such as detection delays, limited coverage range, and high maintenance requirements. As a result, developing a fire detection and monitoring system that uses fire, temperature, and humidity sensors combined with wireless communication technologies is a creative, possible, worthy solution to consider.

## Related Works

The development of fire monitoring and alert systems based on IoT has gained increasing attention in recent years. This section presents several existing works and discusses their limitations, highlighting the gap that this project aims to address.

Nowadays, IoT systems typically use various sensors, ranging from tiny microcontrollers to large ones, and are on the way to merging machine learning in the new era of AI, which is now very popular. I plan to use AI in my future work. Again, in this section “Related Works”, I believe that anyone who reads this thesis someday will find it easy to refer to the paper I will share below:

In the first paper [1] (2023), the authors proposed an IoT-based fire detection and monitoring system using temperature, flame, and smoke sensors with an ESP32 microcontroller, and implemented a LoRa Network to detect and alert on forest and farm. In paper [2] (2013), the authors have used an Arduino-based home fire alarm system with a GSM module and temperature sensor to detect fire and send SMS alerts, enhancing user safety and property protection. Next paper [3] (2025), the authors developed a smart IoT-based system using same list sensors like paper 1 to detect fire, but addition uisng Arduino UNO and gas sensor. The system was implemented on the Blynk platform and uses GSM to send alerts.

Paper [4] (2017), the authors designed an STM32-based wireless fire detection system using multi-sensor fusion and embedded wireless communication technologies like Wi-Fi to overcome the shortcomings of conventional alarm systems. Next research project [5] (2019) about the application of robots, with pages 267–278 focusing on how robots are used in fire monitoring and alert systems. The project uses STM32 and STC89C51 microcontrollers, drones, to connect between air and ground monitoring, and communication through ZigBee.

Bài báo này thì tập trung nhiều hơn về alert [6] (2021), this system uses ESP32 and PIR sensors to detect fire and movement, sending alerts through a Telegram bot via smartphone, with alarms triggered by image and temperature recognition.

Các năm gần đây khi AI phát triển thì việc áp dụng AI vào IoT rất phổ biến. Như bài báo này [7] (2023), the authors combined IoT devices with YOLOv5 to enable early forest fire detection and real-time reporting, aiming to reduce false alarms and improve safety in dry seasons. Bài báo tiếp tơi này khi ở năm 2025 mang tính gợi ý, cải thiện về hệ thống monitoring và alert fire. This paper [8] (2025) present a very good and feasible improved idea for a fire detection system by leveraging IoT and machine learning, as well as combining multiple types of sensors to enhance accuracy and enable remote monitoring. Hướng đến việc để áp dụng AI vào trong 1 hệ thống IoT, quay lại quá khứ năm 2013, bài báo này [9] về real-time fire alarm system was developed using Raspberry Pi and Arduino Uno, with functions including smoke detection, room image capture, and alert transmission via SMS and a web interface. The system only sends alerts to firefighters after user confirmation, aiming to reduce false alarms while saving power and memory. Although this paper dates back to 2013, its implementation was already quite complete—features such as login were already included. However, the web platform at that time had not yet adopted modern design practices, so some features now appear outdated. Still, in terms of implementation, it was a great success, and the authors’ idea was truly bold.

Thanks to this paper and its forward-looking vision—particularly in the direction of integrating AI—it became a source of inspiration for me to develop an extended version based on my professor’s and seniors’ ideas. From that, I created a research project that applies Raspberry Pi and STM32, communicates via HC-12, and uses fire-detecting sensor modules such as flame, temperature, and humidity sensors. Tuy chỉ có cải tiến 1 chút về wireless khi truyền qua vô tuyến, nhưng do với thời gian hạn chế, tôi vẫn chưa khai thác được hết, như kết hơp AI vào dự án. Nhưng đó sẽ nằm sâu trogn mục Fureture work.. Once again, I am grateful to these “giants” whose pioneering work has allowed this small idea of a thesis to be realized.

**In project** [**[1]**](https://github.com/trandinhnguyen/fire-alarm-embedded/)**,** the author developed a fire and smoke detection system using ESP32, connected through Wi-Fi. Although the system successfully detects smoke and transmits data over a network, it does not include user alert notifications or AI-based analysis. Moreover, Wi-Fi-based ESP32 devices are not suitable for long-range communication, which limits deployment in remote or large areas.

**In a second project** [**[2]**](https://github.com/Mohamed-Sherif-Ali/fire-alert-esp32-whatsapp)**,** the author extended the ESP32 system by integrating WhatsApp messaging for fire alerts. This is a practical improvement, enabling real-time user notifications. However, it still relies on Wi-Fi, which is sensitive to interference and has limited scalability in environments with physical obstructions.

[**In project**](https://pccchungthinh.vn/nghien-cuu-ung-dung-mang-cam-bien-khong-day-va-giao-thuc-truyen-thong-zigbee-trong-canh-bao-chay-runghttps:/pccchungthinh.vn/nghien-cuu-ung-dung-mang-cam-bien-khong-day-va-giao-thuc-truyen-thong-zigbee-trong-canh-bao-chay-rung)[**[3]**](https://github.com/simranbajaj27/Forest-fire-detector)**,** the use of Zigbee technology was proposed for fire monitoring in forest environments, utilizing a frequency range from 868 MHz to 2.4 GHz. This approach leverages Zigbee’s strengths, including low power consumption, mesh networking, and flexible frequency operation, which enhances its ability to penetrate obstacles in complex environments, making more usefull compared to Wi-Fi and cellular in such contexts. However, Zigbee also has the disadvantages such as: low data transmission rates (~250 kbps), limited range (10–100 meters depending on the environment), and the requirement for careful configuration, which can be a barrier to large-scale deployment.

More advanced systems such as [**[4]**](https://huggingface.co/prithivMLmods/Fire-Detection-Engine) employ detect fire and smoke image using real-time machine learning models. These systems offer high detection accuracy and precise localization, making them suitable for modern smart monitoring. However, they often lack integration with enviroment sensors and require significant computational resources, which may not be practical for embedded or battery-powered environments. However, with the availability of high-performance embedded platforms like singele board computer Raspberry Pi which featuring up to 8GB RAM and SD storage of 64GB or more, implementing AI in fire monitoring and alert systems is becoming increasingly possible and realistic.

In addition, several works have implemented alert systems using various microcontrollers, integrated mobile applications, and SMS-based notifications. However, web-based platforms remain relatively rare, and wireless transmission technologies beyond Wi-Fi and Zigbee are still underexplored.

To overcome these limitations, this project proposes a fire monitoring and alert system using STM32 microcontrollers in the sensor nodes, combined with the HC-12 wireless module for long-range, low-cost data transmission. Compared to Zigbee, HC-12 offers superior penetration through obstacles and a wider communication range, making it more suitable for harsh or remote environments where short-range or wired solutions are impractical. HC-12 supports a communication distance of up to 1 kilometer in open areas, with selectable power levels, and works stably in various environments. Its transparent serial communication design makes it easy to integrate with most microcontrollers using UART, reducing system complexity and development time. Furthermore, HC-12 is relatively easy to configure using AT commands, which can be set up through popular serial communication tools such as Hercules, Putty. Developers can easily modify parameters like baud rate, transmission power, and communication channel, ip of each HC-12 without complex network setup. As a result, HC-12 is a practical solution for quick development and implementation in fire monitoring systems using IoT. The proposed system in addition to using the HC-12 module, will also include multiple environmental sensors (temperature, humidity, flame), and implement with a web-based interface, allowing users to remotely visualize and manage fire alerts in real time.

## Conclusions

Real-time object detection is an essential technology with applications across many areas. However, achieving real-time speeds, high accuracy, and hardware constraints presents significant challenges.

Both deep learning and traditional computer vision methods have been used to solve these problems. However, there are tradeoffs between accuracy, speed, performance, and hardware configuration. Deep learning methods such as YOLO, R-CNN, or SSD can provide state-of-the-art accuracy but require powerful GPUs or specialized hardware. On the other hand, traditional algorithms like HOG-SVM are more lightweight and hardware-efficient. However, they rely on feature engineering more heavily and may not match the accuracy of deep learning methods.

This thesis proposes a system that can detect objects in real time by utilizing the HOG-SVM algorithm as its core. Various techniques will be employed to enhance its speed and efficiency, including using background motion using the Zipfian Estimation method. The objective is to implement the complete system on an FPGA platform to leverage the hardware design optimization flexibility and parallelism.

# Proposed Real-time Object Detection System

The first chapter of this thesis provides an introduction to object detection and real-time object detection, and it also outlines the specific goals of the thesis. To create a system that detects objects in real-time on embedded edge devices using a stationary camera, it's recommended to use motion detection based on background subtraction using Zipfian Estimation Techniques. This technique helps identify the moving components in the frame, which can then be passed on to HOG-SVM calculation blocks to identify the objects accurately.

In the second chapter of the thesis, a system architecture will be proposed based on the idea above. The chapter will go into the different components of the system, including the motion detection block. The motion detection block works as the pre-processing part of the system. The object detection block uses HOG-SVM as its core and other components, such as the Bilinear Interpolation Scale Generator, which detects objects in multiscale, and the NMS, responsible for collecting the final results. Finally, the post-processing component is responsible for bringing the location of the recognized object back into the frame.

The chapter is divided into four sections. Section 2.1 outlines a system architecture that utilizes lightweight algorithms to detect humans. Section 2.2 describes a background subtraction algorithm, which helps detect motion in image frames from a stationary camera, thereby reducing computation compared to running object detection on every frame. Section 2.3 explains the use of Histogram of Oriented Gradients features with a Support Vector Machine classifier to detect humans in video frames after motion detection. Lastly, Chapter Conclusions are discussed.

## Proposed System using Zifian Estimation Technique and HOG SVM Algorithm

This section presents a system architecture designed for detecting humans in  
video images. The system comprises several components, including motion detection,  
human detection through parallel processing, non-maximum suppression, and merging  
of detection results. The proposed approach offers optimized algorithms and a parallel architecture to improve throughput. The following sections will discuss the Dataflow.  
and System Architecture in detail

### Dataflow

#### Pre-processing

***Input***: RGB Image data.

***Output***: Detected motion areas.

To process two image data, it is necessary first to convert them into grayscale images. This is achieved by computing the average pixel value of the Red, Green, and Blue channels, with each pixel being represented by an 8-bit number ranging from 0 to 255. Next, the Zipfian Estimation Techniques [10] are employed to obtain a list of contours that detect motion areas. This list of contours is then utilized to crop the input grayscale image and put them in human detection processing.

As shown in Figure 2, in the image with a resolution of *1300x982* (used 1.7 scales up) using *77202* sliding windows, we only push three images with resolutions *176x180*, *158x161*, and *182x190* into the object detection block, a total using *556* sliding windows, equivalent to *0.7%.*

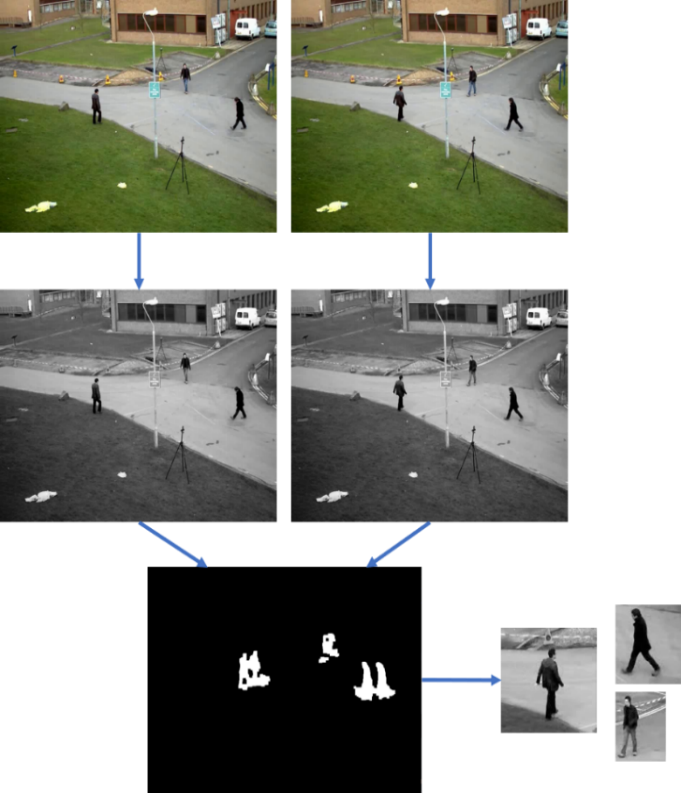


Figure 2 - Pre-processing data flow.

#### Processing

***Input***: Detected motion areas

***Output***: Detected human regions.

For each motion detected area, scale it using the Bilinear Interpolation Scale Generator algorithm, each step of 1.05, decreasing until the height is less than *130* and the width is less than *66* (this is the minimum size of the *64x128* sliding window plus a 1-pixel outer border to implement the HOG-SVM algorithm). Divide these scaled images into six parallel HOG-SVM modules. The result will be pushed into the combined module and finally pushed through the NMS module to filter out the sliding window. The final result will be the area where the system determines to identify the object, precisely, in this case, a person.

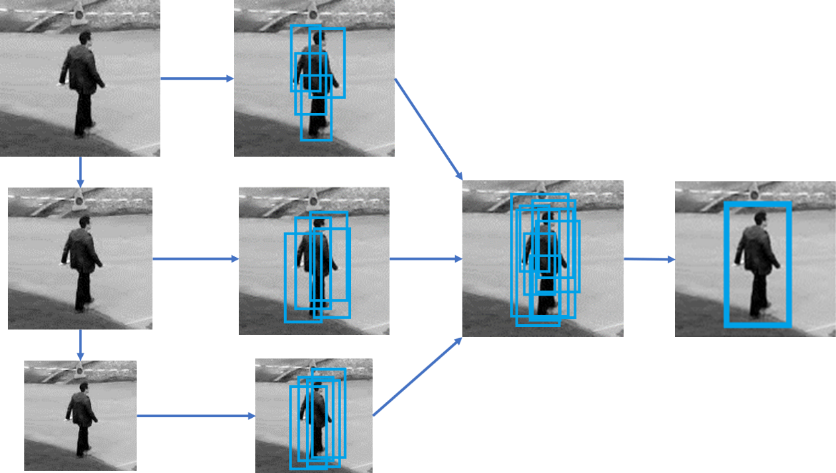


Figure 3 – Processing data flow.

#### Post-processing

***Input***: Detected human areas and raw input image

***Output***: Detected human regions.

The contour result list is a crucial component that merges the output of the human detection process with the saved raw RGB image. Combining these two pieces of information, the contour result list generates the final results, accurately identifying and outlining the detected objects or individuals. The result can be shown on a monitor, saved to a file, or sent to the database for future applications.

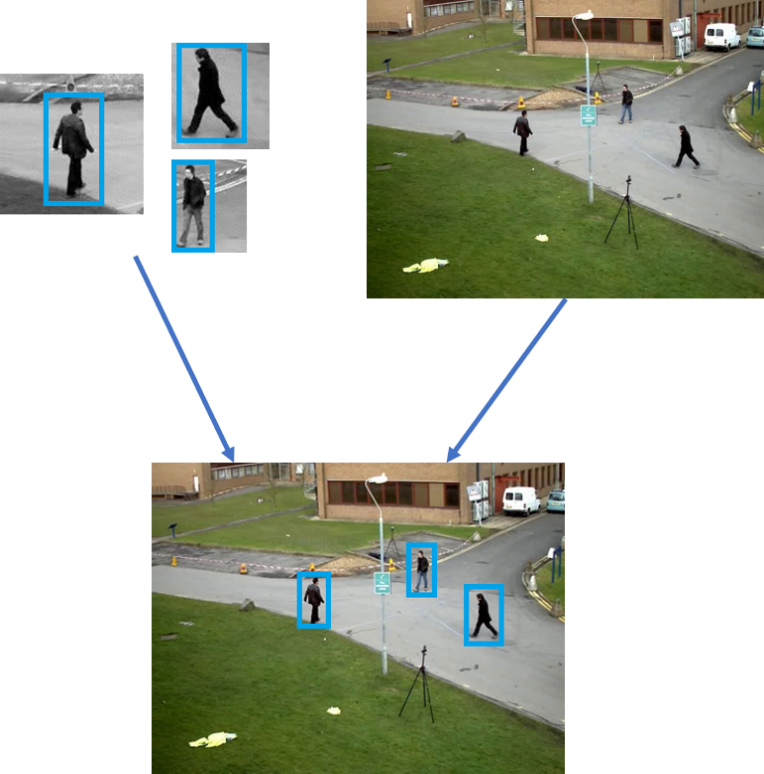


Figure 4 – Post-processing data flow.

### Proposed System

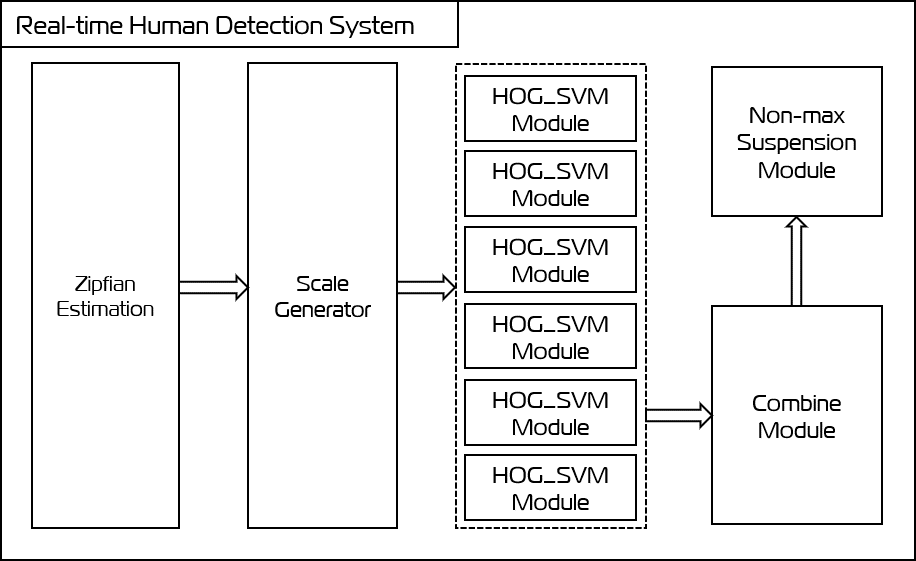


Figure 5 – Proposed system architecture.

The proposed system architecture involves using a sequence of images as input, which will be pre-processed with the help of Zipfian Estimation Techniques. This technique is proper when dealing with dynamic objects such as humans because it only calculates those in the frame's hold. The motion-detected areas are then directed to the Object Detection block, which comprises a Scale Generator module, six HOG-SVM computation modules that run in parallel in a multithread architecture, and an NMS algorithm module. The Scale Generator and multithread HOG-SVM computation modules are required to reduce processing time when scaling the input image to detect all humans in multiscale. The results are merged with the original image for viewing, storage, or transmission to a server for further processing.

Overall, the proposed system uses lightweight algorithms executed in parallel to detect humans in real-time in both video and image sequences. Pre-processing is done using motion detection before applying the HOG-SVM algorithm to reduce the search space. The system then outputs bounding boxes that outline the detected humans, which can be used in various applications. An efficient and accurate human detector is crucial in many areas, such as automated surveillance, advanced driver assistance systems, and human-computer interaction.

## Lightweight Motion Detection Algorithm used on a stationary camera

The stationary camera is widely used for surveillance and monitoring in public areas. It is highly flexible, cost-effective, and easily deployed in large numbers due to its compact size and affordability. Its multifunctional nature makes it an ideal choice for deployment in public spaces. Stationary cameras are non-PTZ cameras installed in a fixed position with a predetermined field of view. These cameras may have adjustable zoom capabilities. As they make the background almost constant during operation, and people or objects appearing in the frame are dynamic, they only calculate those within the frame. This reduces the system's number of calculations, making it faster and enabling it to solve the input image with a higher resolution.

In 2014, The Zipfian Estimation [10] used MJPEG to extract moving and stationary blocks separately, resulting in a compression ratio twice as high as conventional MJPEG. This method encodes only the residuals of the moving blocks, producing a similar quality and bit rate to H.264/AVC. It requires fewer operations than conventional MJPEG if the static scene is equal to or greater than 60%. Zipfian estimation was one of the fastest motion detection algorithms [11] in 2007, [12] in 2009, and the basic Σ-Δ background subtraction algorithm [13] in 2004. Overall, the Zipfian Estimation demonstrates how selective encoding of only dynamic regions can realize substantial compression gains without sacrificing visual quality or requiring extensive computations.

As mentioned above, the Zifian Estimation Module is a lightweight computing technique based on the Sigma-Delta algorithm to detect motion in the frame, as shown in Algorithm 1.

Algorithm 1 - Zipfian estimation [13].



To begin with, the algorithm calculates a threshold based on the frame index *t* and frame index *t-1*. The value of the index is divided by *2m* (where *m* is the number of bits representing a pixel; in our case, *m = 8* for grayscale image), and the remainder is denoted as *p*. Then, the threshold *σ* is equal to *2m* divided by *2p.* The background *Mt* is updated whenever variance *Vt-1* is more significant than *σ*. Next, *Ot* refers to the absolute difference between *It* and *Mt*. To avoid self-referencing, the variance *Vt* is updated using a fixed period *Tv*. Usually, *a Tv* has a power of 2 within the range of 1 to 64. In our case, we have set *Tv* to 1. *N* is typically an amplification factor for the variance *Vt*, ranging from 1 to 4. In our case, *N = 2*. Finally, the movement or stillness of a pixel is determined by comparing its absolute difference with the variance.

Overall, the Zipfian Estimation shows how particular encoding of only dynamic regions can realize significant compression gains without sacrificing visual quality or requiring extensive computations. Additionally, these are lightweight techniques suitable for this thesis's purpose.

## Object Detection using the HOG SVM Algorithm

As mentioned above, many works have been used and developed on the HOG-SVM algorithm. This algorithm was introduced in 2005 for human detection [14]. Histogram of Oriented Gradients (HOG) is a widely used feature for object detection that analyzes edge distribution to balance detection accuracy and complexity [15]. It is the standard baseline for object detection [16]. A Support vector machine (SVM) [17] is also used for classification based on extracted features.

### State of the Art

#### Gradient calculation optimizes

The HOG algorithm generates histograms with complex cells, including inverse tangent, square, square root, and floating point multiplication. Optimizing this step is crucial to improving the HOG-SVM algorithm module.

Pixel derivatives concerning *x*,*y*:



Gradient magnitude and angle calculation:



In 2017, a new technique [18] was proposed for generating cell histograms that simplify the process while producing accurate results. This approach involves bypassing the calculation of pixel gradients. The outcome of this research demonstrates that the reconstruction error is less than 2% with an 8-bit fractional part length. Manipulating the precision of the gradient magnitude is simple using pre-defined sine and cosine values of quantized angles.

The work mentioned above presented a method where the primary gradient vector was split into two vectors, as shown in Figure 6. The values of these two vectors for pixel I(x,y) were calculated using, which takes into account the values of the four surrounding pixels I(x+1,y), I(x-1,y), I(x,y+1), and I(x,y-1).

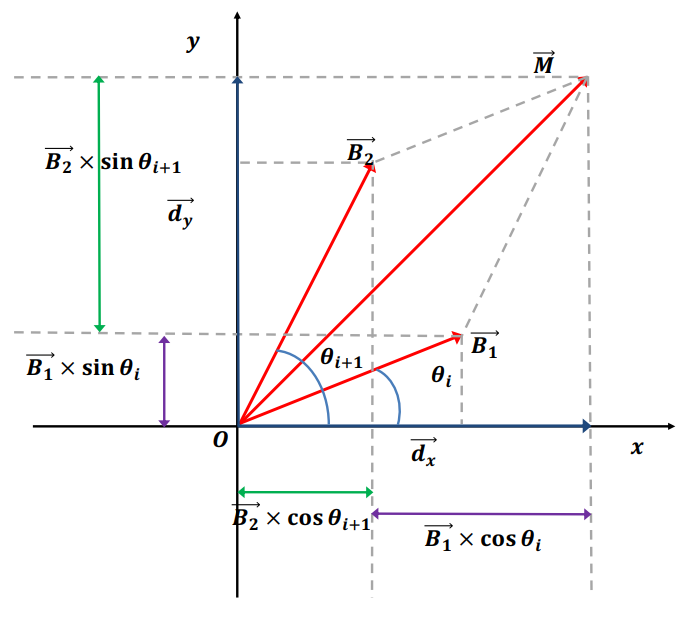


Figure 6 - Decomposition of a vector into the form of two vectors [18].

Two magnitudes are the solutions of the two equations:



This thesis uses this technique to improve the HOG-SVM algorithm and decrease the number of calculations needed to process each sliding window.

#### Implement HOG-SVM into hardware

In 2015, a

frame rates and low latency. HOG-SVM is widely used for object detection but needs optimizations for real-time applications with power constraints.

### Object Detection Block

As mentioned above, the Object Detection Block will include Bilinear Interpolation Scale Generator Module, HOG-SVM module, combine module, and NMS module.

#### Bilinear Interpolation Scale Generator Module

Bilinear

output pixel.

#### HOG-SVM Module

As

to form a block, and the cell histograms are normalized using block data.

##### Gradient calculation and Histogram generation

This

).

Algorithm 3 – Gradient calculation.

will generate all the histograms for the input image used for the next step.

Algorithm 4 – Histogram generation.

##### Descriptor generation and Result

After

generation.

#### Combine Module and NMS Module

The combining

remove redundancies. This block is crucial to this thesis's proposed object detection system.

## Conclusions

In this

architecture offers a promising pipeline for efficient and accurate human detection for real-world applications.

# Implementation and Evaluations

In the previous

## Experiment setup environment

### Mean Average Precision

#### Mean Average Precision (mAP)

an object

#### Mean Average Precision (mAP)

Average

Here is a summary of the steps to calculate the AP:

1. *Generate the prediction scores using the model.*
2. *Convert the prediction scores to class labels.*
3. *Calculate the confusion matrix—TP, FP, TN, FN.*
4. *Calculate the precision and recall metrics.*
5. *Calculate the area under the precision-recall curve.*
6. *Measure the average precision.*

The mAP is calculated by finding Average Precision(AP) for each class and then averaging over several classes.



### Frame rate

## Experimental results

### Input: PETS09-S2L1 [19]

#### Input description

-S2L.

#### Result

and mAP graphs per frame of input PETS209-S2L.

#### Analysis

low mAP score.

### Input: TUD-Stadtmitte [20]

#### Input description

Input racteristic in the photo is that the moving object is only human moves in groups.

#### Result

In of 51

.

#### Analysis

The system can process images in near-real-time but faces significant challenges in accurately identifying and tracking human groups in urban pedestrian settings. The moderate precision combined with low recall and mAP scores suggests that the system might misidentify groups or miss them altogether.

### Input: TUD-Campus [21]

#### Input description

TUD-Campus.

#### Results

In the

D-Campus.

#### Analysis

The system

fast.

## Conclusions

# Conclusions and Perspective

# References

|  |  |
| --- | --- |
| [1] | Pradeep, Aneesh; Latifov, Akmaljon; Yodgorov, Ahborkhuja; Mahkamjonkhojizoda, Nurislombek, "Hazard Detection using custom ESP32 Microcontroller and LoRa," *IEEE,* no. 26 May 2023, 2023. |
| [2] | N. I. M. E. N. M. Z. a. K. S. S. K. M. N. N N Mahzan, "Design of an Arduino-based home fire alarm system with GSM module," *Journal of Physics: Conference Series,* vol. 1019, 2917. |
| [3] | T. Panda, R. Banerjee, A. Pal, S. K. Bishnu and A. Chakraborty, "IOT-Based Home Automation for LPG Gas and Fire Detection System With Automated Safety Measures," *IEEE,* no. 16 April 2025, 2025. |
| [4] | N. I. M. E. N. M. Z. a. K. S. S. K. M. N. N N Mahzan, "Design of an Arduino-based home fire alarm system with GSM module," *Journal of Physics: Conference Series,* vol. 1019, 2017. |
| [5] | H. Y. ·. J. L. ·. L. L. ·. Z. J. ·. Y. L. ·. D. Zhou, "Development of Four Rotor Fire Extinguishing System for Synchronized Monitoring of Air and Ground for Fire Fighting," *ICIRA,* p. 267–278, 2019. |
| [6] | N. Komalapati, V. C. Yarra, L. A. Vyas, Kancharla and T. N. Shankar, "Smart Fire Detection and Surveillance System Using IOT," *IEEE,* 2021. |
| [7] | A. E. H. 1. A. S. S. 1. K. 2. B. A. 1. a. I. C. y Kuldoshbay Avazov 1, "Forest Fire Detection and Notification Method Based on AI and IoT Approaches," *Future Internet (ISSN: 1999 - 5903),* vol. 15, no. 2, 2023. |
| [8] | L. WHITE and R. AJAX, "Improved Fire Detection and Alarm Systems.," 2025. |
| [9] | M. S. B. Bahrudin, "Development of Fire Alarm System using Raspberry Pi and Arduino Uno," *2013 International Conference on Electrical, Electronics and System Engineering (ICEESE),* 2013. |
| [10] | Nguyen, N. S., Bui, D. H., & Tran, X. T., "Reducing temporal redundancy in MJPEG using Zipfian estimation techniques," *2014 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). IEEE,* pp. 65-68, 2014. |
| [11] | A. Manzanera, "σ-δ background subtraction and the Zipf law," *Progress in Pattern Recognition, Image Analysis and Applications: 12th Iberoamericann Congress on Pattern Recognition, CIARP 2007,* pp. 42-51, 2007. |
| [12] | Lacassagne, L., Manzanera, A., & Dupret, A, "Motion detection: Fast and robust algorithms for embedded systems," *IEEE international conference on image processing (ICIP),* pp. 3265-3268, 2009. |
| [13] | Richefeu, A. M. J., "A robust and computationally efficient motion detection algorithm based on σ-δ background estimation," *Indian Conference on Computer Vision, Graphics & amp; Image Processing,* vol. 9, pp. 16-18, 2004. |
| [14] | Dalal, N., & Triggs, B., "Histograms of oriented gradients for human detection," *IEEE computer society conference on computer vision and pattern recognition,* 2005. |
| [15] | Dollár, P., Belongie, S., & Perona, P., "The fastest pedestrian detector in the west," 2010. |
| [16] | Dollar, P., Wojek, C., Schiele, B., & Perona, P., "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence, 34(4),* pp. 743-761, 2011. |
| [17] | Cristianini, N., & Shawe-Taylor, J., An introduction to support vector machines and other kernel-based learning methods, Cambridge university press, 2000. |
| [18] | Ho, H. H., Nguyen, N. S., Bui, D. H., & Tran, X. T, "Accurate and low complex cell histogram generation by bypass the gradient of pixel computation," *4th NAFOSTED Conference on Information and Computer Science,* pp. 201-206, 2017. |
| [19] | Ferryman, J. & Shahrokni, A., "PETS2009: Dataset and challenge.," *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance,* 2009. |
| [20] | Andriluka, M., Roth, S. & Schiele, B. , "Monocular 3D Pose Estimation and Tracking by Detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2010. |
| [21] | Andriluka, M., Roth, S. & Schiele, B., "People-Tracking-by-Detection and People-Detection-by-Tracking," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2008. |
| [22] | A. Suleiman, "An energy-efficient hardware implementation of HOG-based object detection at 1080HD 60 fps with multi-scale support.," 2016. |
| [23] | Nguyen, N. D., Bui, D. H., & Tran, X. T., "A novel hardware architecture for human detection using HOG-SVM co-optimization," *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05).,* vol. 1, pp. 886-893, 2019. |
| [24] | Nguyen, T. A., Tran-Thi, T. Q., Bui, D. H., & Tran, X. T, "FPGA-Based Human Detection System using HOG-SVM Algorithm," *International Conference on Advanced Technologies for Communications (ATC),* pp. 72-77, 2023. |
| [25] | [Online]. Available: https://learnopencv.com/non-maximum-suppression-theory-and-implementation-in-pytorch/. |
| [26] | [Online]. Available: https://www.v7labs.com/blog/mean-average-precision. |
| [27] | Shanghai Xian Dai Architecture,Engineering&Consulting Co. ,China , "Information fusion technology based on wireless fire detection and alarm system," *Journal of Physics: Conference Series,* pp. 883-887, 21-11-2013. |