universite PARIS-SACLAY

ĐẠI HỌC CÔNG NGHỆ

ĐHQGHN

Since 1906 VNU ĐẠI HỌC QUỐC GIA HÀ NỘI Vietnam National University, Hanoi

DEVELOPING A REAL-TIME OBJECT DETECTION SYSTEM ON FPGA

Master thesis of Paris-Saclay University

Specialization: M2 Communication and Data Engineering

Master student: NGUYEN Trung Kien

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 LIBESSART	Examiner
CentraleSupélec	

Thesis supervision

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 LIBESSART	Co-supervisor of the thesis
CentraleSupélec	

ABSTRACT

AUTHORSHIP

Hanoi, January 22nd, 2024

Author

ACKNOWLEDGEMENT

Contents

List of Abbreviations

Abbreviation	Definition
IoT	Internet of Things
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

List of Tables

Introduction

Chapter 1. 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 IoT and its applications in fire monitoring systems. *Section 1.2* discusses the current situation and limitations of existing fire monitoring systems. *Section 1.3* presents modern fire detection methods using current technologies, along with their limitations.

1.1. 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 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.

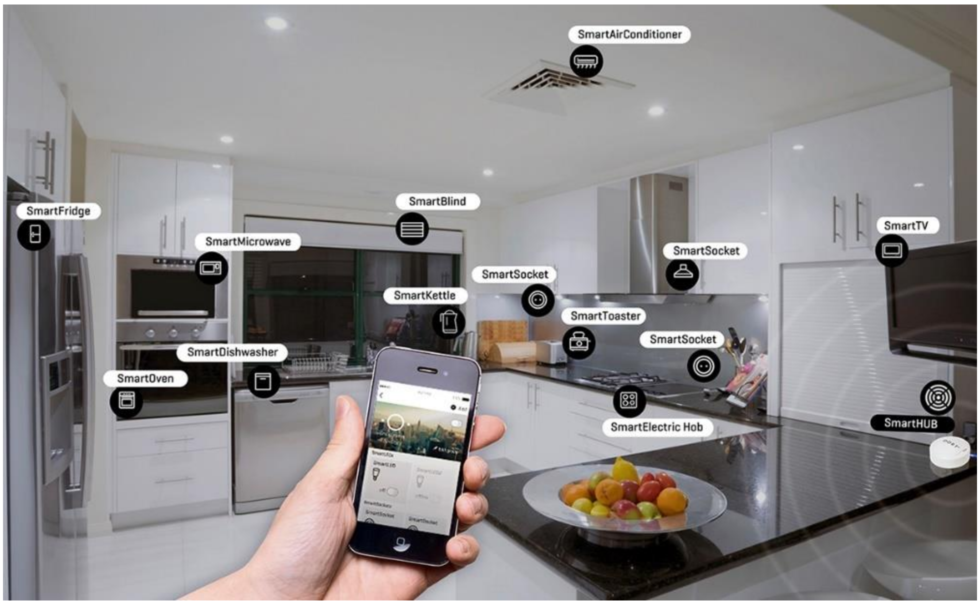

Figure 1. An IoT system with interconnected smart devices.

1.2. 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.

1.3. 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. Back to 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. In the next paper [3] (2025), the authors developed a smart IoT-based system using the same list of sensors as paper 1 to detect fire, but in addition, using Arduino UNO and a 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 a block of sensors and embedded wireless communication technologies like Wi-Fi to detect fire. Next paper [5] (2019) shows a big improvement in the application of robots. Pages 267–278 of the paper focus on the use of robots in fire monitoring and alert systems. The project utilizes STM32 and STC89C51 microcontrollers, drones, to connect air and ground monitoring systems and facilitate communication through ZigBee. This paper [6] (2021) focuses more on the alerting part. The authors use ESP32 and PIR sensors to detect fire and movement, sending alerts through a Telegram bot to a smartphone, with feature alarms by image and temperature parameters. When AI becomes popular, some papers nowadays are applying it in IoT systems to detect and alert fire. For example, paper [7] (2023) presents a system that combines IoT devices with YOLOv5 to enable early forest fire and real-time detection, aiming to reduce false alarms and enhance safety during dry seasons. While the approach in [7] is already technically advanced, [8](2025) adds a suggestive improvement to fire monitoring and alert systems by using IoT and machine learning, and also combining various sensor types, to improve detection accuracy and support remote monitoring. Although this recent work represents the development, the core concept of fire detection systems has already been explored in earlier studies.. Indeed, a paper from 2013 [9] discusses a real-time fire alarm system developed using Raspberry Pi and Arduino Uno. This system features include smoke detection, room image capture, and alert through SMS and a web interface. It only sends alerts to firefighters after user confirmation to reduce false alarms. Although this paper was from 2013, its implementation was quite complete – features like login were already added. However, the web design is still basic and a little bit outdated. However, about implementation, it was a meaningful success, and the authors' idea was forward–thinking.

As mentioned at the beginning of Section 1.3, my thesis does not apply AI in the IoT system. However, this will be explored in future work. To recognize and expand on the ideas presented from the science papers, many of which use wireless technologies such as Wi-Fi or GSM. This thesis proposes enhancing current approaches by applying radio communication to an IoT-based fire detection and alert system. Once again, I am grateful to these “giants” whose pioneering work has allowed this small idea of a thesis to be developed.

1.4. Conclusions

An IoT system is an essential technology with applications in monitoring and fire detection. Achieving real-time monitoring, high accuracy, processing all scenarios, and overcoming hardware limitations creates serious difficulties. The use of IoT brings a promising solution, enabling real-time monitoring, remote access, and automated alert notifications. This project aims to inherit and integrate the ideas from those previous works, while also contributing further improvements to develop a more responsive and reliable fire alert system. It also introduces new innovations to improve the system’s flexibility and performance in real-life situations. In Chapter 2*In System Architecture Design*, a proposal will be presented, focusing on the integration of hardware and software components, including sensor modules, communication systems, and control logic. All of which will be implemented through software to build the proposed IoT-based fire monitoring solution.

Chapter 2. System Architecture Design

The first chapter of this thesis provided an overview of IoT technology and its practical applications in fire detection and monitoring. It also discussed the current state of research and proposed a new direction by enhancing existing ideas with the use of radio communication. While many previous systems relied on Wi-Fi or GSM, this thesis focuses on integrating HC-12 wireless modules for long-range, low-cost, low-power communication. The core goal is to build an IoT-based system that monitors, detects fire, and alerts in real time.

Building on this ideal, *Chapter 2* presents the proposed system architecture, which consists of both hardware and software components designed to work together for efficient fire monitoring. The system includes hardware devices: Raspberry Pi and STM32 microcontrollers, HC-12 modules for communication, and a variety of sensors to collect fire-related data.

This chapter is organized into three main sections:

- *Section 2.1 Hardware System Architecture*: This section introduces the hardware components used at both the sensor node and the central processing unit. It explains what each component is, its role in the system, and the reasons behind choosing it. Subsections cover specific components, including fire sensors, temperature and humidity modules, STM32 for edge processing, and the Raspberry Pi for central control and web server hosting.

- *Section 2.2: Software & Firmware System Architecture*: This part details how the software & firmware work across the system. It begins with sensor-side firmware, which includes collecting data, encoding, and transmission. On the server side, it explains how data is received, decoded, stored, and visualized through a Flask-based web application. This section also includes user authentication, account management, and how software sends alerts (using Fuzzy Logic and Threshold Evaluation.

- *Section 2.3 Conclusions:* This final section summarizes all components introduced in the *Chapter 2* and prepares for the system implementation and evaluation that will follow in later chapters.

.

2.1. Hardware System Architecture

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

2.1.1. Dataflow

(1) 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%.*

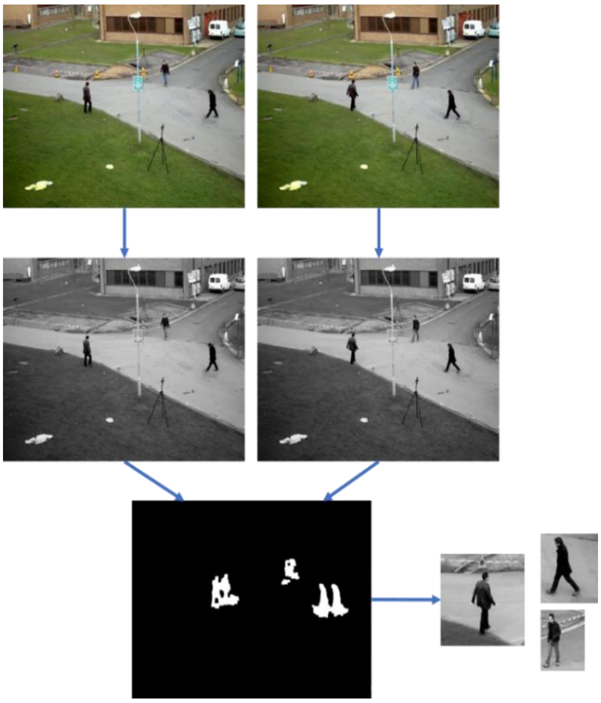

Figure 2 - Pre-processing data flow.

(2) 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.

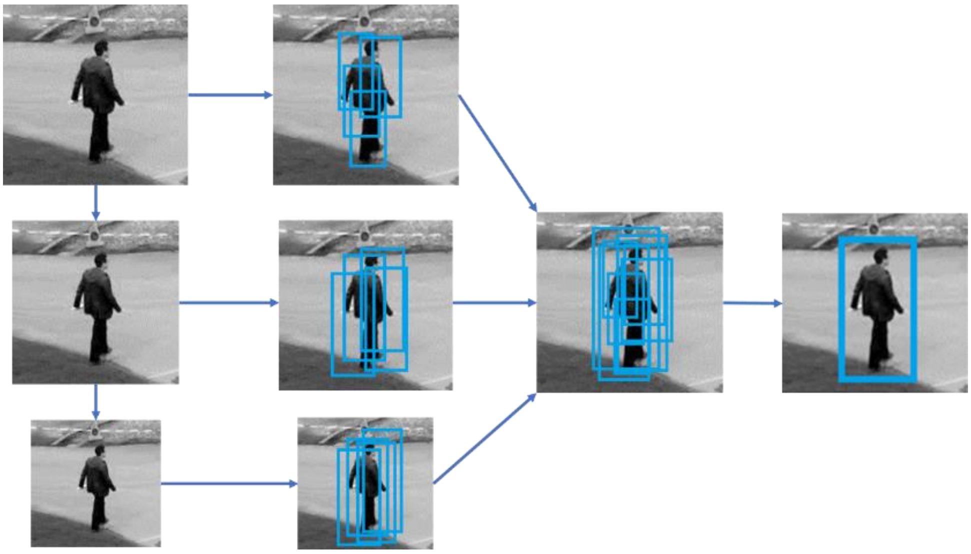

Figure 3 – Processing data flow.

(3) 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.

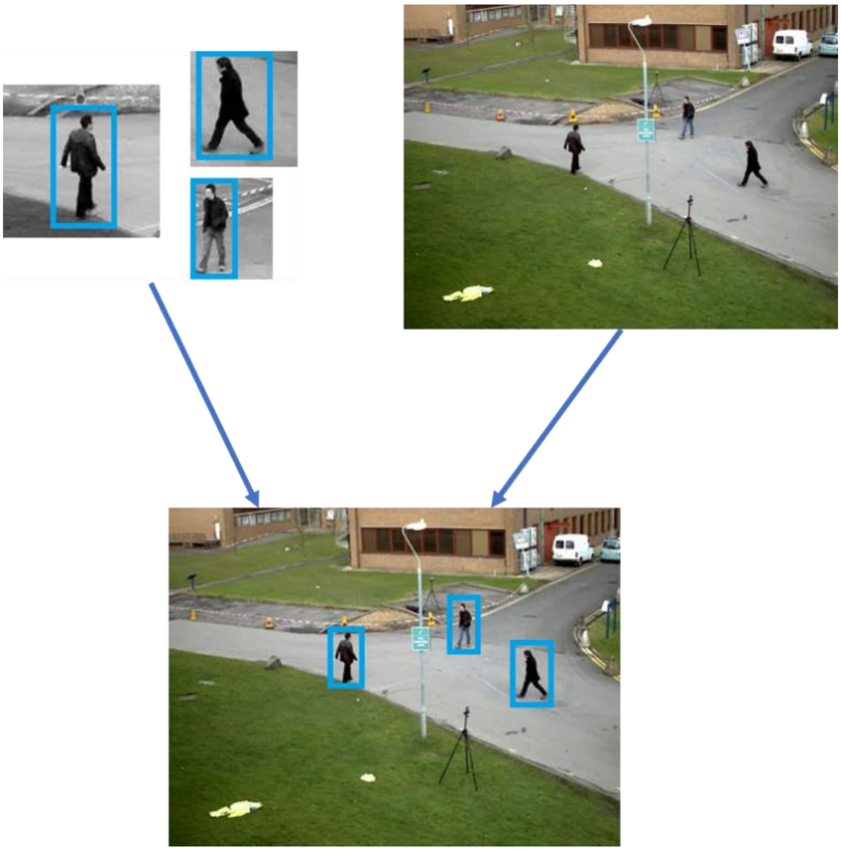

Figure 4 – Post-processing data flow.

2.1.2. Proposed System

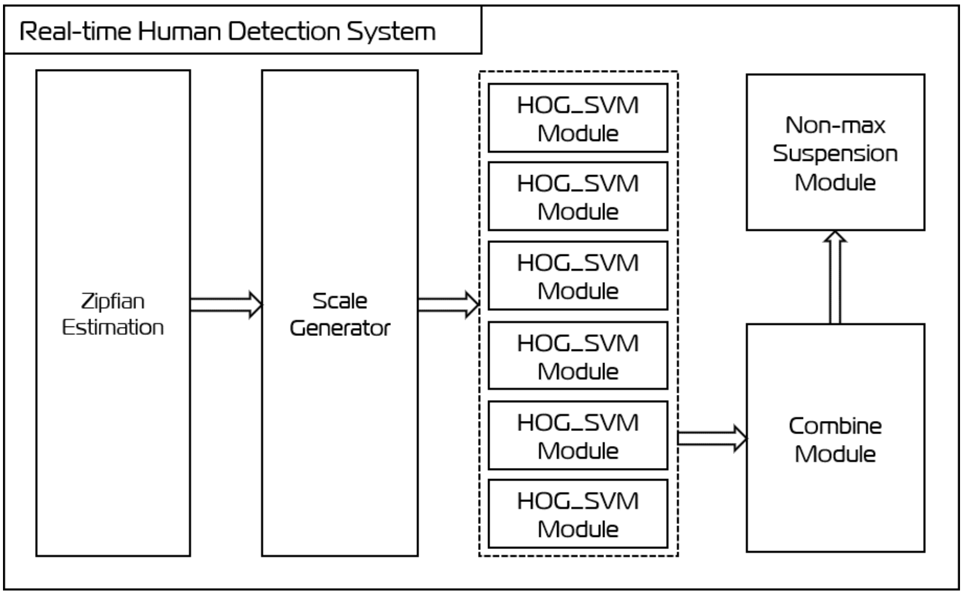

Figure 5 – Proposed system architecture.

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.

2.2. 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].

```
find the greatest $2^p$ that divides ($t \bmod 2^m$)
set $\sigma = \frac{2^m}{2^p}$
foreach pixel x do
   if $V_{t-1}(x) > \sigma$ then
      if $M_{t-1}(x) < I_t(x)$ then $M_t(x) \leftarrow M_{t-1} + 1$
      if $M_{t-1}(x) > I_t(x)$ then $M_t(x) \leftarrow M_{t-1} - 1$
foreach pixel x do
   $O_t(x) = |M_t(x) - I_t(x)|$
foreach pixel x do
   if $t \bmod TV = 0$ then
      if $V_{t-1}(x) < N \times O_t(x)$ then $V_t(x) \leftarrow V_{t-1}(x) + 1$
      if $V_{t-1}(x) > N \times O_t(x)$ then $V_t(x) \leftarrow V_{t-1}(x) - 1$
foreach pixel x do
   if $O_t(x) < V_t(x)$ then $E_t(x) \leftarrow 0$
```

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 2^m (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 2^m divided by 2^p. The background M_t is updated whenever variance V_{t-1} is more significant than σ. Next, O_t refers to the absolute difference between I_t and M_t. To avoid self-referencing, the variance V_t is updated using a fixed period T_v. Usually, *a* T_v has a power of 2 within the range of 1 to 64. In our case, we have set T_v to 1. N is typically an amplification factor for the variance V_t, 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.

2.3. 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.

2.3.1. State of the Art

(1) 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:

$$dx = I(x, y+1) - I(x, y-1);\ dy = I(x+1, y) - I(x-1, y)$$

Gradient magnitude and angle calculation:

$$\left\|\overrightarrow{M(x,y)}\right\| = \sqrt{d_x^2 + d_y^2};\ \alpha = \arctan\left(\frac{dx}{dy}\right)$$

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).

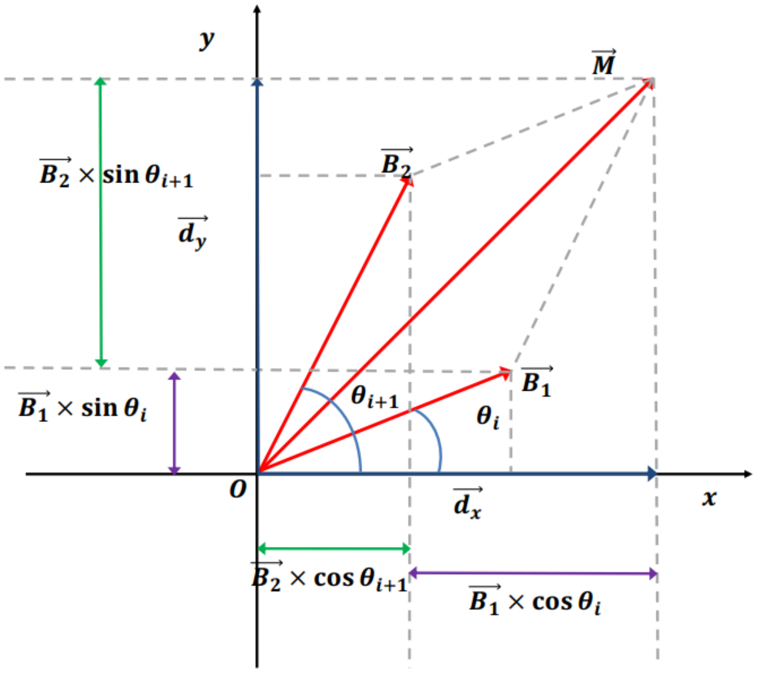
y
$\overrightarrow{M}$
$\overrightarrow{B_2} \times \sin\theta_{i+1}$
$\overrightarrow{B_2}$
$\overrightarrow{d_y}$
$\overrightarrow{B_1}$
$\overrightarrow{B_1} \times \sin\theta_i$
θ_{i+1}
θ_i
O
$\overrightarrow{d_x}$
x
$\overrightarrow{B_2} \times \cos\theta_{i+1}$
$\overrightarrow{B_1} \times \cos\theta_i$

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

Two magnitudes are the solutions of the two equations:

$$\left|\left|\overrightarrow{B_{\theta_i}}\right|\right| = \frac{\sin(\theta_{i+1})\, d_x - \cos(\theta_{i+1})\, d_y}{\sin(20)};\ ||\overrightarrow{B_{\theta_{i+1}}}|| = \frac{\cos(\theta_i)\, d_y - \sin(\theta_i)\, d_x}{\sin(20)}$$

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

(2) 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.

2.3.2. 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.

(1) Bilinear Interpolation Scale Generator Module

Bilinear

output pixel.

(2) HOG-SVM Module

As

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

(a) <u>Gradient calculation and Histogram generation</u>

This

).

Algorithm 2 – Gradient calculation.

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

Algorithm 3 – Histogram generation.

(b) <u>Descriptor generation and Result</u>

After

generation.

(3) Combine Module and NMS Module

The combining

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

2.4. Conclusions

In this

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

Chapter 3. Implementation and Evaluations

In the previous

3.1. Experiment setup environment

3.1.1. Mean Average Precision

(1) Mean Average Precision (mAP)

an object

(2) 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.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} APi$$

3.1.2. Frame rate

3.2. Experimental results

3.2.1. Input: PETS09-S2L1 [19]

(1) Input description

-S2L.

(2) Result

and mAP graphs per frame of input PETS209-S2L.

(3) Analysis

low mAP score.

3.2.2. Input: TUD-Stadtmitte [20]

(1) Input description

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

(2) Result

In of 51

.

(3) 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.

3.2.3. Input: TUD-Campus [21]

(1) Input description

TUD-Campus.

(2) Results

In the

D-Campus.

(3) Analysis

The system

fast.

3.3. 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.