**CHAPTER II**

**PROBLEM DEFINITION**

Facilities with a medium footfall of 50-500 visitors per day suffer greatly to implement proper surveillance solutions that balance between the security needs and practical constraints. Traditional CCTV systems are very common, but suffer from a few key limitations:

* Resource Constraints:
  + Small to medium-sized businesses cannot afford costly server infrastructure or pricey, cloud-based analytics platforms
  + Monthly fees on cloud subscription and bandwidth requirements drive high-end surveillance features out of reach
  + Fewer IT personnel and scarce skillset to manage complex systems
* Operational Inefficiencies:
  + Manual monitoring of multiple video feeds tends to be labor-intensive and error-prone
  + Monitoring an individual movement requires reviewing multiple, distinct video footages coming from different cameras
  + Real-time threat detection is dependent on human vigilance and reaction
  + Does not have automated alerts and incident reporting features
* Technical Limitations:
  + Basic CCTV systems do not offer intelligent monitoring features with just a simple recording function
  + Cannot detect anomalies or patterns without automating the system
  + Constant surveillance cannot be maintained at peak hours
  + Restricted ability to derive insights from collected video
* Privacy and Compliance
  + Security mandates needs to be balanced against privacy regulation
  + Data needs to be stored and processed in accordance with local privacy laws
  + Risk of transmitting sensitive video to cloud servers

**CHAPTER III**

**LITERATURE SURVEY**

**INTRODUCTION**

Doing an extensive literature survey, helped us to recognize the models and prerequisites that will be required for completion for this project.  
  
Some of them required include:

* [yolo](https://docs.ultralytics.com/)
* [facenet](https://github.com/davidsandberg/facenet)
* frame subtraction
* another custom built algorithm

We have also identified previously done work and their advantages/disadvantages, helping us narrow down our thoughts to the parts of the project that would require more attention and how we can work on improving it.

* [**Multicamera Edge-Computing System for Persons Indoor Location and Tracking**](https://www.sciencedirect.com/science/article/pii/S2542660523002639)**" (2023)**
* **Authors: Ángel Carro-Lagoa, Valentín Barral, Miguel González-López, Carlos Escudero, Luis Castedo**
* **Dataset Utilized:**
* **Substantiates the role of sound datasets such as** [**CITIC-MTMC**](https://github.com/GTEC-UDC/citic-mtmc-dataset) **in algorithm design and evaluation of tracking methodologies.**
* **Summary:**
* **The paper discusses how multimodal AI supports the advance towards Artificial General Intelligence and relates the implementation to educational contexts and it engages multimodal data-visual, auditory, and linguistic- in order to create dynamic analytics driven by AI that can happen in real-time with adaptive decision making. This is applicable to your project, as it is aimed at exploiting AI for making real-time surveillance tasks. Strengths**
* **1.Multimodal Inputs Integration**
* **o The paper emphasizes combining text, audio, video, and kinesthetic data to enhance learning, reflecting human-like cognitive processes.**
* **o This approach ensures adaptive learning tailored to various sensory and cognitive preferences.**
* **2.Alignment with Educational Theories:**
* **o Frameworks like** [**Dual Coding Theory**](https://en.wikipedia.org/wiki/Dual-coding_theory) **and VARK are utilized to validate the need for multimodal learning, adding credibility to the research.**
* **3.Future-Oriented Approach:**
* **o In relation to education, this paper explains the long-term potential of AI by setting the basis for scalable and dynamic educational tools.**
* **4. Practical Applications:**
* **o This research gives emphasis to practical applications, such as virtual tutors, real-time feedback systems, and personalized learning pathways, which make the research actionable and relevant in reality.**
* **Weaknesses**
* **1. Lack of Implementation Details:**
* **The paper is more or less theoretical in its approach but lacks extensive experimental validation or real-world deployment case studies.**
* **2. Highly Advanced AI Models**
* **The paper relies on models such as** [**GPT-4 Vision**](https://help.openai.com/en/articles/8555496-gpt-4-vision-api) **and** [**KOSMOS-2**](https://github.com/microsoft/unilm/tree/master/kosmos-2)**, which assume a high level of computational resources that might be inappropriate for low-resource setups.**
* **3. Scalability:**
* **In cases where multimodality is effective in small-scale or controlled environments, the approach's scalability for diverse educational settings is not well addressed.**
* **Key Contributions of the Paper:**
* **Multicamera Edge-Computing System:**
* **Instantiates a multi-camera indoor tracking scenario with AI accelerators such as Google Coral and NVIDIA Jetson Nano.**
* **It processes images locally with privacy in mind as only non-sensitive information, such as positions and appearance features, is sent to a central server.**
* **Presenting a multi-camera indoor dataset with real-world coordinates and annotations for multiple individuals.**
* **Analyzes tracking algorithms in complex indoor environments, both overlapping and non-overlapping FoVs.**
* **Tracking Accuracy and Scalability**
* **Multiple individuals can be tracked in real-time with low power consumption, and thus it is perfectly scalable for medium to large setups.**
* **It negates issues like occlusion and blind spots, ensuring robust accuracy in complex environments involving large camera networks.**
* **Conclusion**
* **Multicamera edge-computing systems serve as a very viable, affordable solution for privacy-preserving tracking in small businesses and institutional facilities.**
* **Studies show that the edge devices are indeed capable of real-time tracking, although such limitations persist on reduced accuracy in low FPS or complex environment.**

1. [**A Modified Frame Difference Method Using Correlation Coefficient for Background Subtraction**](https://www.researchgate.net/publication/306067947_A_Modified_Frame_Difference_Method_Using_Correlation_Coefficient_for_Background_Subtraction) **A Modified Frame Difference Method Using Correlation Coefficient for Background Subtraction**
2. **Introduction**
3. **Authors: P. Ramya, R. Rajeswari**
4. **Overview: In video surveillance, foreground object identification has always been a challenge in background subtraction with increasing complexity and dynamics.**
5. **Strengths:**
6. **New contribution: Authors put forward a new amendment of frame difference, introducing correlation coefficient-based background subtraction, in order to be used for better detection accuracy and speed.**
7. **Results and Discussion: The approach is tested on difficult data sets such as Wallflower and I2R; the various performance metrics are clear to prove that the proposed methodology is much better than the conventional ones in terms of precision, recall, and F-measure.**
8. **Scalability: Block-based classification with pixel-level refinement offers an appropriate balance between computational efficiency and detection performance, thus suitable for real-time applications.**
9. **Weaknesses:**
10. **Threshold Dependence: The approach relies highly on predefined thresholds (such as correlation coefficient > 0.5), which could not generalize to different video conditions without fine tuning the parameters further.**
11. **Limited Diversity of Dataset: Although this approach has been validated on two datasets, the video sequences may not limit all kinds of variations that may occur in real life such as varied weather conditions and extreme lighting changes.**
12. **Block Division Simplification: The 8x8 size of the block used limits further information in the foreground and background, which may lead to error in unusual cases and high complexity scenes.**
13. **Conclusion**
14. **The proposed technique integrates correlation coefficient analysis for foreground-background subtraction and, therefore, it is more efficient and accurate. It surpasses frame difference method in a variety of challenging cases. The authors proposed that future work might be directed to the use of auxiliary information, namely, shape and edge characteristics, to further improve the accuracy of detection.**

1. [**Milvus: A Purpose-Built Vector Data Management System**](https://www.cs.purdue.edu/homes/csjgwang/pubs/SIGMOD21_Milvus.pdf) **Introduction**

* **Authors:** Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu
* **Overview**: This paper introduces **Milvus**, a specialized vector database system that is superior to existing systems such as **Faiss and SPTAG**, which chiefly focus on the static similarity search of vectors. Milvus addresses the needs of AI and ML applications to process many forms of unstructured data by introducing a specialized vector database system designed for managing **large-scale, dynamic vector data** with fast query processing.

**Key Contributions**

1. **Efficient Dynamic Data Handling**:

* Milvus employs a **log-structured merge-tree (LSM-based)** architecture, which can support efficient insertion, deletion, and real-time querying of dynamic vector data, while solving a critical limitation in existing systems..

1. **Advanced Query Processing**:

* Milvus is different from traditional vector search systems since it supports attribute filtering and multi-vector queries, which can be helpful for more complex applications, such as finding entities that are defined by multiple features.

1. **Optimized for Heterogeneous Computing**:

* Milvus is designed to fully exploit both **CPUs and GPUs** along with **SIMD-aware** optimizations and the support of **multi-GPU**, ensuring both scalability and performance, particularly in modern computing environments.

1. **Open Source and Distributed Architecture**:

* The system is fully open-source and with a **distributed shared-storage architecture** that will allow it to share data and spread through multiple nodes in a billion-scale vector management.

**Strengths**:

* + **Performance**: According to experiments, Milvus is significantly outperforming many existing systems in terms of **query throughput** and even **dynamic data handling**, by **up to two orders of magnitude** faster than systems like Faiss and SPTAG​
  + **Scalability**: Milvus scales efficiently across multiple nodes, handling **billion-scale datasets** with ease, demonstrating its potential for large-scale AI-driven applications​
  + **Advanced Query Support**: The system's ability to handle **multi-vector queries** and **attribute filtering** allows it to support a broader range of applications compared to other vector search systems​.

**Weaknesses**:

* + **General-Purpose Limitations**: Though Milvus is great at handling the vector data, as a **general-purpose DBMS**, it is not so good compared to others such as **PostgreSQL** or **ElasticSearch** integrating vector data into relational structure​..
  + **Hardware Dependence**: Because of the reliance on **optimizing with the GPU**, a system like this is more inaccessible to organizations with less infrastructure, whereas a CPU-centric system like SPTAG might be more cost-effective​.

**Conclusion**

Milvus is a quantum step forward in **vector data management**, providing the leadership edge of high-performance, scalable and sophisticated query capabilities over the state-of-the-art, particularly towards AI/ML application. Its capacity to manage **dynamic, large-scale multi-vector data** while using **heterogeneous computing environments** like CPUs and GPUs differentiates it from other systems. Following improvements could be worked out for future scope as suggested by the paper:

* **Support for Categorical Attributes**: It can support **categorical attributes** using more sophisticated indexing techniques like **inverted lists** or **bitmaps**, and thus increase the scalability of the system to accommodate more complex queries.
* **Optimizing Multi-Vector Query Processing**: There is scope to explore optimization of the querying with **multi-vector queries** and therefore could focus on **optimal performance** given even more heterogeneous and challenging data sets..
* **Dynamic Partitioning with Machine Learning**: Combining **machine learning and statistics** can further boost the query performance and scalability further as the data is dynamically partitioned, and even the **number of partitions** required for the attribute search feature is determined.

1. [**MobileFaceNets: Efficient CNNs for Accurate RealTime Face Verification on Mobile Devices**](https://arxiv.org/abs/1804.07573)

**Introduction**

* **Authors**: Sheng Chen, Yang Liu, Xiang Gao, Zhen Han
* **Overview**: This paper addresses the challenge of real-time face verification on **mobile and embedded devices**. Face verification is being applied intensively in mobile applications such as smartphone unlock, mobile payments, and login application on the face of increasing demand for highly efficient models that can be run in real-time on even very limited computational resources. However, traditional models such as **MobileNetV1, ShuffleNet, and MobileNetV2** are either too large or lack required accuracy for face verification, leading to performance issues in mobile applications.
* **Datasets Used:**
* [CASIA-WebFace](https://paperswithcode.com/dataset/casia-webface): Used for training face verification models with ArcFace loss.
* [MS-Celeb-1M](https://paperswithcode.com/dataset/ms-celeb-1m) (Refined): A cleaned version of this large-scale dataset with 3.8 million images from 85,000 subjects, used for training to achieve enhanced accuracy.
* [LFW (Labeled Faces in the Wild)](https://vis-www.cs.umass.edu/lfw/): Used for testing the face verification accuracy.
* [AgeDB-30:](https://paperswithcode.com/dataset/agedb) Used for further accuracy evaluation on age-related variations.

**Key Contributions**

* **Introducing MobileFaceNets:** Efficient CNN Model for Face Verification---less than 1 million parameters but greater accuracy and more than **twice the speedup** compared to MobileNetV2 under the same conditions​.
* [**Global Depthwise Convolution (GDConv) Layer**](https://www.researchgate.net/figure/Two-kinds-of-convolution-a-The-global-depthwise-convolution-GDConv-b-The-global_fig4_370370630): Instead of using a **global average pooling layer**, which is commonly used in mobile networks, the authors make use of a **global depthwise convolution (GDConv) layer** to treat different units of FMap-end with different importance. It improves verification accuracy in faces by treating different spatial units of the input feature map with varying importance​..
* **High Performance Improvement**: For benchmark datasets like **LFW** (Labeled Faces in the Wild) and **MegaFace**, MobileFaceNets surpasses the prior mobile CNNs, achieving **99.55% accuracy** on LFW and **92.59% TAR@FAR10⁻⁶** on MegaFace​.
* **Small Real-Time Efficiency**: At its smallest version, MobileFaceNets still consumed only **4MB** of model size, while it took **18 milliseconds** to draw inferences on mobile devices. Hence, it was specifically suitable for real-time applications

**Strengths**:

* **Small Model Size**: With less than 1 million parameters, MobileFaceNets are significantly smaller than other models, yet their **accuracy is comparable** to much larger models like **FaceNet** and **ArcFace**​.
* **Speed**: MobileFaceNets achieve **over 2× speedup** relative to **MobileNetV2** on mobile platforms, which is much-needed for real-time applications like smartphone unlock​.
* **Performance**: On hard datasets like **MegaFace** and **LFW**, MobileFaceNets display competitive performance, beating other mobile-centric networks like **ShuffleNet** and **MobileNetV1**​.

**Weaknesses**:

* **Accuracy vs. Size Trade-off**: While MobileFaceNets are efficient, there is still the problem of **trade-off between size and accuracy**. Some larger models, though not as efficient, achieve slightly better accuracy​.
* **Limited Task Focus**: The model is specifically designed for **face verification**, meaning it may not generalize well to other visual recognition tasks. For applications requiring broader visual tasks, other models like **MobileNet** might be preferred.  
    
    
    
    
    
  **Conclusion:**

**MobileFaceNets** bring an important contribution to mobile deep learning as they achieve **high-accuracy face verification** in real-time on mobile devices, with the model being extremely compact. Adding the **GDConv layer** further improves efficiency, marking this approach as a superior alternative to existing state-of-art mobile CNN architectures.

1. [**You Only Look Once: Unified, Real-Time Object Detection**](https://arxiv.org/abs/1506.02640)

**Introduction**

* **Authors:** Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi
* **Overview:** YOLO (You Only Look Once) is a new way for real-time object detection that is unlike the two old ways of region proposal networks. Its selling point is its unified architecture which processes an image in a single pass, making YOLO very much faster than its predecessors, while still maintaining competitive accuracy.
* **Dataset**: [MS- COCO](https://cocodataset.org/)

**Key Contributions**

1. **Unified Detection Framework:** Unlike other detection frameworks that rely on region proposals followed by classification, YOLO interprets the problem of detection as a regression problem. Hence, it divides the entire image into an S×SS \\\\times SS×S grid and predicts class probabilities directly for each cell of the grid as well as bounding boxes..
2. **Speed and Efficiency:** YOLO goes full speed with the ability to process up to 45 frames per second on a standard GPU. In this way, it is apt for applications that require instant feedback such as autonomous driving and surveillance.
3. **Systematic Error Analysis:** The authors presented very exhaustive experiments to analyze the performance of YOLO in comparison with other very popular and efficient object-detection methods. It provided them insights into common issues in detection tasks, which presented their work in the form of strengths and weaknesses.
4. **Architecture:** YOLO uses a single CNN architecture that is outputting bounding boxes as well as class probability simultaneously, which greatly reduces the detection procedure..

**Strengths:**

* **Speed:** Most importantly, its real-time ability to detect objects would be very important for many applications; since it is primarily the speed that matters for applications requiring quick response times.
* **Simplicity:** YOLO is as easy to implement and to modify based on its very simple methodology, which makes it a convenient object detection framework for researchers as well as developers.
* **Generalization:** It possesses strong generalization capabilities over various datasets and object classes present and thus is promoted for its use in diverse settings.

**Weaknesses:**

* **Localization Accuracy:** YOLO is extremely fast, but it suffers at fine localization compared to many other complex methods that region proposal initialization relies upon. This grid coarser approach can cause prediction errors in bounding boxes when objects are small.
* **Overlapping Objects:** For this reason, the grid system might not make a difference between closely positioned objects and YOLO could have issues with the job of detecting overlapping objects.

**Conclusion:**

The YOLO paper has significantly progressed the field of object detection because it proposes a fast and efficient framework that can optimize for the best speed-accuracy trade-off. Its innovative approach impacted subsequent studies, resulting in further advancements in real-time object detection systems. Although there could be room for improvements, especially in localization, YOLO has seeded itself as fundamental groundwork that paved the way for future developments in the domain.

**CHAPTER IV**

**RESEARCH / TECHNOLOGY GAPS AND CHALLENGES**

The CCTV systems which are currently in use make use of 2 methods in most cases:

1. cloud computing
2. central server

the cost for each of these technologies is very high since a very huge amount of computational power is required.   
  
But while thinking on a couple of things we found that the AI cameras that are on market today already have an arm processor and a small (NPU/TPU) in them. This is currently being used to make the camera change direction towards a moving object or focus on an object of interest. But we can use the same infrastructure to perform some object-detection and human-recognition algorithms.  
  
Then we can translate this to vectors and collect and run more algorithms only on these vectors to trace the same person on multiple cameras at different times.  
  
Lightweight face recognition and object detection is already available but the challenge is to achieve a notable frame rate and then to run our custom made algorithm to detect the person at multiple cameras.

**CHAPTER V**

**OBJECTIVES**

The following are the objectives we wish to achieve moving forward in this project…

1. Around 10-15 fps of processing in the camera.
2. Able to recognise the same person from multiple cameras.
3. Should be able to run on a device with low compute power.
4. Try to keep the memory footprint to the lowest possible.
5. Try to perform maximum computing within the same camera.
6. use minimalistic models.

**CHAPTER VI**

**DATA EXPLORATION**

If we don't have to training the facial recognition and yolo model again from scratch we believe in using a video dataset similar to these:

1. <https://github.com/cvdfoundation/kinetics-dataset>
2. <https://github.com/shahroudy/NTURGB-D>

Since we aren’t able to find and video dataset which are annotated, with the names of the person present in it, we plan to use yolo and facenet to quickly give them random names and then test out our project.

**CHAPTER VII**

**CONCLUSION OF CAPSTONE PROJECT PHASE - 1**

After the comprehensive literature survey and market analysis, it is our pride to declare that our project proposal has been finally accepted by the panel of experts. In fact, it is a significant point in our research and development journey.

Through heavy and careful research, we arrived to have a deep understanding of the current technologies that exist and how it can be further improved. We have a clear idea of what we should be working on and what are the initial problems that might occur.

We now have a clear idea of the pipeline and are now able to clearly jot down a software requirement specification. We have decided upon a few technologies and libraries which we are sure to use, and have a rough idea of the entire pipeline for the project.

**CHAPTER VIII**

**PLAN OF WORK FOR CAPSTONE PROJECT PHASE - 2**

We now feel we have a rough pipeline for the project. We can begin by testing out the available solutions for parts of out project. Also we can start searching for the relevant libraries at the same time. After this we can have a clear idea of exactly what parts of the project can be outsourced. Once we have accomplished that, we are able to divide our work into chunks and start concentrating on one chunk at a time.

[Iterative and Incremental Development methodology](https://www.geeksforgeeks.org/iterative-and-incremental-development-iid/), specifically the [Spiral Model](https://www.javatpoint.com/software-engineering-spiral-model), with elements of Agile development. Here's why:

* Risk-Driven Approach
  + Continuous risk assessment critical for security and AI systems
  + Enables systematic identification and mitigation of technical risks
  + Enables progressive refinement of system capabilities
* Iterative Release Strategy
  + Several prototype versions with incremental feature enhancements
  + Each iteration focuses on:
    - Improved AI detection accuracy
    - Improved edge computing performance
    - Enhanced privacy and security features

We will also utilize a versioning system that would enable our development work with the support of git.

.**CHAPTER IX**

**REFERENCES/BIBLIOGRAPHY**

[1] Á. Carro-Lagoa, V. Barral, M. González-López, C. J. Escudero, and L. Castedo, "CITIC Research Center & Department of Computer Engineering, University of A Coruña, 15071, A Coruña, Spain," received Mar. 8, 2023, revised Jul. 26, 2023, accepted Sept. 8, 2023, available online Sept. 12, 2023, version of record Sept. 18, 2023.

[2] Y. Martínez-Díaz, H. Méndez-Vázquez, L. S. Luevano, L. Chang and M. GonzalezMendoza, "Lightweight Low-Resolution Face Recognition for Surveillance Applications," 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 2021

[3] P. Ramya, R. Rajeshwari, “A Modified frame difference method using correlation coefficient for background subtraction”, 6th International Conference On Advances In Computing & Communications, ICACC 2016, 6-8 September 2016, Cochin, India.

[4] Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, Kun Yu, Yuxing Yuan, Yinghao Zou, Jiquan Long, Yudong Cai, Zhenxiang Li, Zhifeng Zhang, Yihua Mo, Jun Gu, Ruiyi Jiang, Yi Wei, Charles Xie. 2021. Milvus: A PurposeBuilt Vector Data Management System. In Proceedings of the 2021 International Conference on Management of Data (SIGMOD ’21), June 20–25, 2021, Virtual Event, China. ACM, New York, NY, USA, 14 pages. https: //[doi.org/10.1145/3448016.3457550](http://doi.org/10.1145/3448016.3457550)  
  
[5] Chen, Sheng & Liu, Yang & Gao, Xiang & Han, Zhen. (2018). MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices. 10.48550/arXiv.1804.07573.

1. CCTV <https://en.wikipedia.org/wiki/Closed-circuit_television>
2. Ultralytics YOLO <https://docs.ultralytics.com/#yolo-a-brief-history>
3. Mobile Facenet <https://github.com/pedroprates/mobile-face-net>
4. Milvus <https://milvus.io/>
5. Raspberry Pi Zero <https://www.raspberrypi.com/products/raspberry-pi-zero/>
6. Ingenic <https://en.ingenic.com.cn/products-detail/id-21.html>