# Tiny-ML based Person Identification in Dynamic Motion

by

Mirza Raiyan Ahmed
20101188
Shahed Pervez Nokib
20301123
Shadman Ahmad Nafee
20341033
Jannatus Sakira Khondaker
20301468

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

Department of Computer Science and Engineering
BRAC University
May 2024

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

Mirza Raiyan Ahmed
20101188

Shahed Pervez Nokib
20301123

Shadman Ahmad Nafee
20341033

Jannatus Sakira Khondaker
20301468

# Approval

The thesis/project titled "Tiny-ML based Person Identification in Dynamic Motion" submitted by

1. Mirza Raiyan Ahmed (20101188)
2. Shahed Pervez Nokib (20101123)
3. Shadman Ahmad Nafee (20341033)
4. Jannatus Sakira Khondaker (20301468)

**Examining Committee:**

Supervisor:
(Member)

Amitabha Chakrabarty
Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Robiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

# Abstract

TinyML, short for Tiny Machine Learning, focuses on small, low-power machine learning systems, with a significant emphasis on human identification. This capability is crucial in areas like access control, security, and law enforcement. Traditional methods like fingerprint and face recognition often require costly hardware and software, whereas TinyML offers a more economical and efficient alternative. TinyML models can be trained using various sensors, such as cameras, microphones, and accelerometers, making them suitable for devices like smartphones and smartwatches. Techniques such as gait and voice recognition are also viable with TinyML, with computer vision playing a crucial role in processing visual data for human identification. Despite the challenges in facial recognition, such as the need for extensive data and computational resources, TinyML models paired with computer vision hold promise for improving effectiveness, affordability, and security.Our analysis of CNN architectures (SqueezeNet, ResNet50, VGG16, MobileNetV2, and MobileFaceNet) for human identification in dynamic motion reveals significant performance improvements with data augmentation. ResNet50 and MobileNetV2 showed the most notable enhancements, with accuracy improvements to 96%, demonstrating robust generalization with enriched data. MobileNetV2 achieved a precision of 97% and an F1 score of 94%, highlighting its effectiveness. While all models benefited from data augmentation, VGG16 and MobileFaceNet also exhibited significant enhancements. These findings underscore the critical role of data augmentation in bolstering model performance and suggest that deploying ResNet50 and MobileNetV2 on devices like the ESP32-CAM could yield highly effective human identification systems. This analysis highlights the interplay between model architecture, dataset characteristics, and data augmentation in shaping model efficacy for real-world applications.



**Keywords:** Tiny-ML; Machine Learning; Microcontroller; Dynamic Motion

# Acknowledgement

First, thanks to Allah, our thesis was completed without any interruption. We are grateful to our supervisor, Dr. Amitabha Chakrabarty, for his unwavering support, insightful guidance, and exceptional expertise throughout our research. His extensive knowledge and assistance helped guide this thesis's success. We are also grateful to Dr. Chakrabarty's research assistant, Mr. Fahim, for his invaluable help and cooperation.
This thesis would not have been possible without Dr. Chakrabarty and Mr. Fahim. We are grateful to have worked with them and appreciate their remarkable contributions.

# Table of Contents

# Chapter 1

# Introduction

Kallimani et al. [34] described tiny-ML as the implementation of a machine learning model in devices that have extremely limited accessibility to resources. In order to provide real-time inference for edge computing, sensor data processing, and other applications, it optimizes and scales models for these devices. Utilizing model quantization, compression, and hardware acceleration makes it possible to perform machine learning on compact devices in an efficient and low-footprint manner. In this day and age of microcontrollers, microprocessors, and the Internet of Things, where the development of low-power embedded systems is accelerating at a great rate. Throughout this expansion, Tiny-ML has been performing exceptionally well. However, the progress that Tiny-ML has made in the field of human identification is not particularly groundbreaking. Gupta et al.[25] stated that, despite its widespread application in Human Activity Recognition, Human Identification is not yet guaranteed to be completely reliable.

Wang[3] stated that Person Identification is an important aspect of:

- **Evolution and Performance of TinyML**
  TinyML has performed exceptionally well throughout this expansion, which has resulted in a significant acceleration in the development of applications that require low power embedded systems. The progress that TinyML has made in human identification has not been particularly ground-breaking, despite the fact that it has been successful in a variety of fields. Despite the fact that it is widely used in Human Activity Recognition (HAR), there is still room for improvement in terms of its reliability in human identification.

- **Security and Access Control**
  When it comes to improving security and managing access, human identification is absolutely necessary. Unlocking smartphones and other electronic devices, gaining access to secure buildings and facilities, and verifying identities at airports and border controls are all possible uses for this technology. This guarantees that only individuals who have been authorized to do so can access sensitive areas and services.

- **Law Enforcement and Public Safety**
  When it comes to law enforcement, human identification is a useful tool for locating and identifying suspects, as well as monitoring public areas to identify potential dangers. It is also necessary for the distribution of Amber Alerts

and the conduct of searches for individuals who have gone missing, which contributes to the safety and security of the general public.

- **Authentication and Authorization**
  The existence of human identification is absolutely necessary in order to keep connections to online services and bank accounts safe. During online transactions, it verifies the identities of users, which enables personalized marketing and targeted advertising among other benefits. This method also contributes to market research by assisting in the identification of customer demographics as well as the emotional responses of customers.

- **Healthcare**
  Accurate patient identification and access to medical records are both made possible through the use of human identification in the healthcare industry. It makes it easier to continuously monitor patients for any signs of discomfort or pain, which ultimately results in an improvement in the quality of care that is provided.

- **Retail**
  Through the use of facial recognition systems, retail human identification works to improve the in-store shopping experience while also assisting in the prevention of shoplifting. Through the use of this technology, a more secure and individualized shopping environment is created.

- **Education and Learning**
  A safe and productive learning environment can be ensured through the use of human identification in educational settings. This improves campus security and allows for the monitoring of student engagement and physical presence.

- **Smart Cities**
  The identification of individuals is beneficial to the management of traffic, the pricing of congestion, and the maintenance of the performance of urban infrastructure in smart cities. It contributes to the creation of urban environments that are more efficient and livable.

- **Event Management**
  Human identification is utilized for the purpose of entry management, as well as the identification of VIPs and registered attendees, at events such as concerts, sporting events, and conferences. Because of this, the event operations will run smoothly and safely.

- **Emotion Analysis**
  In addition, human identification systems can be utilized to recognize and assess emotional states for the purpose of conducting market research and improving user experiences, which ultimately results in products and services that are more specifically targeted and efficient.

- **Airport Security**
  Facial identification is an example of biometric systems that are used to verify the identities of passengers. This helps to ensure that only authorized individuals are allowed to board certain flights. Consequently, this contributes to the prevention of identity fraud and the enhancement of the safety of air travel.

- **Banking**
  For the purpose of ensuring that only authorized individuals are able to access accounts and carry out transactions, banks can utilize human identification methods such as facial recognition. Fraud and unauthorized access are less likely to occur as a result of this.

- **Entertainment**
  The identification systems utilized by streaming platforms allow for the customization of content recommendations to users based on their viewing habits and preferences. Consequently, this results in a user experience that is more interesting and enjoyable.

A methodical and accurate recognition of individuals is required for human identification. This is accomplished by distinguishing the distinctive qualities, characteristics, or attributes that each individual possesses. Accurate identification and verification of individuals is accomplished through the utilization of a variety of methods and technologies. For the purpose of ensuring that each individual can be identified in a manner that is both unique and reliable, physical characteristics such as fingerprints and facial features, as well as behavioral characteristics such as typing patterns and voice recognition, are utilized.

The concept of person identification refers to the methodical and accurate recognition of individuals through the process of recognizing and distinguishing the distinctive qualities, characteristics, or attributes that that person possesses. For the purpose of accurately identifying and verifying individuals based on the distinctive characteristics they possess, this process involves the utilization of a variety of techniques and technologies. Physical characteristics, such as fingerprints and facial features, as well as behavioral characteristics, such as typing patterns and voice recognition, are examples of these characteristics. According to Wang et al.[2], the end goal is to make sure that every individual can be identified in a way that is both unique and reliable, which will ultimately improve security, make access easier, and allow for more users to have more personalized experiences.The process of identifying individuals is necessary for a wide variety of purposes, including the regulation of access, the enhancement of personalized experiences, and the maintenance of security regulations. Identification systems are able to accurately and efficiently recognize and differentiate individuals by utilizing the distinctive individual traits and characteristics that are unique to each person. This enables a wide range of applications in today's civilization. Not only does the methodical and accurate recognition of individuals improve security and access control, but it also makes it possible to provide experiences that are personalized and pertinent to the individual. As a result, it is an essential component of the many technological advancements that have occurred in recent times.

## 1.1 Problem Statements

Person identification has become feasible as a result of the development of highly functional computing devices in this era. For the time being, however, the microcontroller and microprocessor level is not even close to being reliable. In this day and age of high-end McU and MpU, human identification through the use of a low-cost embedded system is extremely important, not only for reasons of security but also for other reasons. Even though McU or MpU are already monitoring the activities that are carried out by human beings, identification is still dependent on the devices that are able to function effectively. Finding a solution to this problem is essential, and it can be implemented anywhere that is required for cost-effectiveness. According to Gupta et al.[25] at the moment, there are some works featuring a microcontroller and microprocessor unit; however, the performance rate is only approximately 55%, which decreases when exposed to unfavorable environments and atmospheric conditions.

## 1.2 Research Objective

Through our research, we want to make progress in the fields of security systems and identifying people by focusing on these main goals:

**Identifying Humans in Dynamic Motion Using a Low-Cost Embedded System**
Our goal is to design and implement an advanced embedded system that is capable of accurately identifying individuals while they are in motion. In order to guarantee accessibility and feasibility for widespread use, this system will make use of hardware that is both cost-effective and efficient.

**Developing and Validating Robust Methods for Human Identification**
In terms of accuracy, reliability, and resistance to fraud, our goal is to develop innovative identification methods that are superior to those that are currently in use. Performance will be improved through the implementation of these methods, which will incorporate sophisticated algorithms and machine learning techniques.

**Enhancing Security Systems for High-Profile Security Zones**
Our objective is to accomplish the development of an advanced security system that is specifically designed to meet the requirements of high-security environments, such as government buildings, research facilities, and corporate headquarters. In order to provide security solutions that are both effective and efficient, this system will make use of diminutive devices that are inexpensive. This will eliminate the requirement for costly infrastructure.

**Optimizing Identification Systems for Challenging Conditions**
Developing an identification system that is able to function effectively in environments with low levels of light and a high degree of complexity will be our primary focus. In order to maintain a high level of identification accuracy even when presented with challenging circumstances, this requires the integration of sophisticated sensor technologies and image processing algorithms.

**Ensuring Scalability and Adaptability of the System**
The development of a scalable and adaptable identification system that is capable of being easily deployed in a variety of settings, ranging from small-scale installations to large-scale security networks, will also be a primary focus of our research. This includes making sure that the system is compatible with the infrastructure that is already in place and having the capability to update it as new technologies become available.

**Evaluating System Performance through Real-World Testing**
In order to assess the efficiency of the methods and systems that we have developed, we will carry out extensive testing in situations that are representative of the real world. In order to accomplish this, we will need to work together with various security agencies and industry partners in order to collect useful information and improve our solutions based on empirical investigations.

## 1.3 Novel Key Contributions

The main novel idea of this thesis is the creation of an affordable, effective, and expandable Tiny-ML system for identifying humans in motion. The research presents a robust approach for real-time human recognition by utilizing the ESP32-CAM, a tiny microcontroller with integrated camera capabilities. The system utilizes sophisticated machine learning architectures, such as CNN models like SqueezeNet, ResNet50, VGG16, MobileNetV2, and MobileFaceNet. These models have been enhanced by data augmentation to attain exceptional accuracy and generalization. This approach not only improves security systems in important areas but also provides flexibility in difficult settings and scalability for wider use, representing substantial progress in the field of Tiny Machine Learning and its practical applications. Moreover, the system's main dataset, which is created specifically for the application using the ESP32-CAM, guarantees that it is customized to meet individual requirements, hence improving its dependability and efficiency.

# Chapter 2

# Literature Review

Embedded systems have played a crucial role in collecting images with low-powered devices. A number of studies have extensively investigated this field, utilizing different models based on artificial intelligence. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), VGG-16, and AlexNet have proven to be highly successful in detecting various human actions. These activities encompass the identification of driver drowsiness, the detection of drowning occurrences, and the recognition of fundamental human behaviors such as lying down, sitting, and walking.

Furthermore, there have been notable breakthroughs in the field of facial identification through the utilisation of embedded technologies. By combining embedded systems with sophisticated AI algorithms, it is now possible to achieve real-time facial recognition even on hardware with limited processing capabilities. Methods such as Convolutional Neural Networks (CNNs) and advanced machine learning frameworks like MobileNet and Tiny-YOLO have played a crucial role in this field. These models are designed to maximize performance and efficiency, enabling them to operate on devices with restricted computing capabilities, such as smartphones, security cameras, and IoT devices. This feature has diverse uses, encompassing safe access management, tailored user experiences, and heightened security in surveillance systems. The incorporation of facial recognition technology into embedded systems highlights the capacity of artificial intelligence to revolutionize ordinary equipment into perceptive and aware instruments.

Gupta et al.[25] primarily worked to enhance the efficiency of human activity recognition models through the implementation of optimization techniques such as pruning and quantization. This is achieved by reducing the model's size, energy consumption, and network bandwidth utilisation, while ensuring that the accuracy of the model remains largely unaffected. In this study, we have determined that the models can undergo compression by a factor of up to 10 times following model optimization for CNN. For DeepConv LSTM, the compression ratio can reach nearly 20 times, while for Multi-Layer LSTMs, it ranges from 4 to 5 times. Additionally, we have observed that the accuracy of these models remains unchanged until reaching a sparsity level of 80 percent for CNN, 95 percent for DeepConv LSTM, and approximately 60 to 65 percent for Multi-Layer LSTM. However, beyond these sparsity thresholds, the accuracy (specifically, test accuracy) begins to deteriorate, with each model experiencing a decrease of up to 6 percent. Therefore, three models have been utilised in

order to facilitate the comparison process and determine the most suitable model for optimising the modelling process. Given the current state of affairs and existing knowledge, there is an ongoing requirement for further exploration and investigation in various domains, including the field of human activity recognition. Notably, the use of the TinyML approach has promising prospects for novel research advancements in this particular subject. Emerging optimisation approaches are currently being explored, and investigating their application in human activity identification is an intriguing research area for analysis. In addition to the aforementioned advancements, potential research progress in this domain could involve exploring novel optimization strategies, such as pruning and quantization, which have the potential to yield improved outcomes through the comparison of sizes and accuracies.

Sudharsan et al. [27] worked with the TinyML-CAM pipeline, which involves the development of a real-time image recognition system utilizing a Random Forest Classifier (RF). The system attained a frame rate of 80 frames per second (FPS) and utilized a mere 1 kilobyte (kB) of random-access memory (RAM) on the ESP32 microcontroller board. The process of developing a TinyML pipeline involves four distinct stages: Image Collection through HTTP, Feature Extraction, Classifier Training, and, ultimately, Porting to C++. The outcome was notably favorable, particularly in terms of the digital signal processing time for image frames, which was reduced to a mere. The duration for picture feature extraction was found to be less than 12 milliseconds, while the time required for classification was seen to be only 20 microseconds. The histogram of oriented gradients (HOG) technique, implemented through digital signal processing, achieved a frame rate of 83.3 frames per second (FPS). In the context of Pairplot analysis, the author observes that there is a significant overlap in the features of the Portenta and Pi, resulting in frequent mislabeling. However, the author suggests that this issue can be disregarded if improvements are made to the quality and size of the dataset. One advantage of utilizing the ESP32 cam over Arduino is its reduced memory consumption, with only 1KB being required. The system exhibits utility in its low power consumption, offering high-performance solutions in a compact form factor, and may operate offline, so circumventing the need for network connectivity. The preservation of privacy is upheld.

Gruber and Franz [33] used the Keras model to recognize performed fitness exercises with the help of an accelerometer mounted on a microcontroller using an existing dataset to train the model. They used the SensorTile kit as the main MCU with a Nucleo-F411, as the SensorTile kit does not have its own SWD (Serial Wire Debug) port. They connected the microcontroller project with an iOS app to store the collected data without a permanent internet connection. In total, the model's accuracy is 89.76%, broken down into three output classes of exercises.

Zhou et al. [31] works with a novel framework for human activity recognition which has been developed comprising five distinct components. The technique under consideration comprises several components, namely individual convolution subnet, cross-channel interaction, cross-channel fusion, temporal information extraction, temporal information enhancement, and a final prediction step.The dataset utilised in this study consisted of several datasets, namely PAMAP2, Opportunity,

Skoda, DSADS, DAPHNET, and WISDM. All six of these datasets are considered benchmark datasets and are frequently utilised in academic research and practical applications. The performance of PAMAP2, DSADS, and DAPHNET is superior.The TinyHAR model demonstrates superior performance compared to the baseline model in terms of F1M, with improvements of 1.8%, 3.1%, and 9.78% in original size.The TinyHAR model achieves a slightly lower F1M score on the Skoda dataset while demonstrating a significantly higher F1M score on the WISDM dataset. The TinyHAR model achieves a lower F1M score when applied to the Opportunity dataset. In comparison to previous datasets, the dataset Opportunity has a significantly larger number of sensor channels. The poorer outcome can be attributed to the smaller size and increased difficulty in extraction.

Alajlan et al.[32] in his paper reforms the problems of driver drowsiness using Tinyml.. The dataset was created by the authors using five subjects with 60 open-eye images and 60 closed-eye images. The result was 95% detection accuracy in good conditions, but accuracy fell under low illumination and when drivers were wearing glasses. The author used MobileNet-V2 SqueezeNet , AlexNet, MobileNet-V3 and the accuracy was 99.60% ,99.47% , 99.11% and 98.32% respectively.

Xiong et al. [40] presents an innovative approach for the prediction and validation of human motion by leveraging joint data through AVI video conversion. The convolutional neural network demonstrates impeccable accuracy in recognising the initial category of motion. Nevertheless, when the range of motion expands and the lower limbs become involved in the movement, there is a modest decrease in the accuracy of motion identification. However, the percentage remains consistently over 85%, indicating its appropriateness as a foundational framework for forecasting human joint motion. The convolutional neural networks effectively extract motion bases, resulting in a notable level of recognition accuracy. Consequently, these motion bases are deemed appropriate for predicting human joint motion.

Wu et al. [22] proposed The CMR-Block for the purpose of complete motion representation learning. It is composed of two modules, namely the Channel-wise Motion Enhancement Module and the Spatial-wise Motion Enhancement Module. The primary objective of the CME is to optimize the discriminative channels while simultaneously suppressing irrelevant ones within the global temporal receptive field. Our method demonstrates superior performance compared to the existing state-of-the-art in the temporal reasoning datasets Something-Something V1 and V2. Specifically, while utilizing 16 frames as input, our method achieves a performance improvement of 2.3% and 1.9% in Something-Something V1 and V2, respectively.

Aggarwal et al. [1] provide a comprehensive review of the latest advancements in human motion analysis from image sequences, employing a hierarchical methodology. One significant limitation associated with 2D models pertains to their inherent constraint in terms of the range of camera angles that can be observed. The implementation of a 3D volumetric model serves to mitigate this issue; nevertheless, it necessitates the inclusion of more parameters and entails a more costly computational procedure during the matching phase.

Islam et al.[26] examine three deep neural network (DNN) models utilised for image classification in various applications: MNIST for image-based applications, Human Activity Recognition for wearable applications, and Google Keyword Recognition for audio applications. The utilization of deep neural network (DNN) model training and weight pruning techniques has resulted in a significant reduction of 2 times in the number of weights for the convolutional (CONV) layer. Additionally, the BCM method has effectively decreased the weight of the fully connected (FC) layer by a factor of 128 times for the MNIST challenge. In the context of Hebbian-based synaptic plasticity, the BCM learning rule is employed to adjust the weights of neural network layers. Specifically, the first layer weights are decreased by a factor of 128, while the second layer weights are reduced by a factor of 64. In the context of energy usage, the model demonstrated superior performance compared to SONIC and TAILS, obtaining a factor of 6.1 times and 4.31 times, respectively.

Yuan et al. [29] tried to solve the difficulty of human identification and activity recognition by sensor fusion. The PRF and PIR sensors were combined in order to enhance the precision of HIAR and mitigate the impact of interferences. Consequently, the sensor fusion technique significantly enhanced the PRF data set that exhibited the most substantial interference. Human identification is described as the process of recognizing distinct individuals who are engaged in similar activities within interconnected contexts. According to their research findings, sensor fusion has been demonstrated to mitigate the issue of label confusion and enhance accuracy. The RNN model was employed to discern human behavior. The PRF-PIR architecture has demonstrated its efficacy in mitigating the shortcomings of the MI-PIR system with regard to vertical Field of View (FoV) and ambient dependence. The PRF-PIR system demonstrated a 19.30% increase in accuracy compared to the MI-PIR system alone.

Giordano et al.[12] used two sensing devices that were utilized for the purpose of face detection. The initial component consists of a time-of-flight sensor in conjunction with a low-power camera. The test was conducted using Time-of-Flight (ToF) technology in a low-power mode, with an automatic triggering frequency of 1Hz. The microcontroller is activated when the measured distance falls below the predetermined threshold value. The aforementioned approach proved to be advantageous due to its ability to capture high-resolution, well-defined photographs. The researchers designed an embedded system with the purpose of capturing and processing an image in the presence of a nearby individual. The researchers demonstrated the feasibility of developing a convolutional neural network (CNN) with excellent accuracy and minimal complexity for implementation in a vision node that operates without the need for a battery. In order to accomplish this objective, a convolutional neural network (CNN) was devised with the purpose of recognizing and identifying an individual's facial features. The utilization of data augmentation was beneficial in both the training and testing phases of the network. The neural network was trained using the TensorFlow framework and subsequently implemented on the microcontroller via TensorFlow Lite. To reduce processing demands, a convolutional and dense layer architecture was employed, consisting of a total of 39,821 parameters. ReLu activation functions were applied to all layers. Subsequently, two distinct processors were employed, each employing four distinct implementation approaches,

in order to strike a balance between precision and energy efficiency at the centre. After conducting numerous tests, the wireless smart sensor node demonstrated the ability to operate without the need for batteries and sustain itself autonomously. It achieved an impressive accuracy rate of up to 97% across five distinct categories for face detection.

Sabovic et al.[38] used The Arduino Nano 33 BLE as a microcontroller and a camera module connected to it to perform tinyML algorithms and application tasks on battery-less IoT devices. Their main idea was to run two ML models that can show a result of how efficient tinyML algorithms are in a resource-constrained battery-less IoT device. So, they run the low-power tinyML model on that IoT device and a heavy-weight ML model in a Cloud data center. After the training process, when the heavy-weight ML model achieved the desired accuracy, they used different techniques such as Knowledge Distillation, Pruning, and Quantization to optimize the model so that they could deploy it in the IoT device. As a proof of concept of their algorithm, they used the IoT device for person detection. They used CNN and MobileNet V1 for the embedded system. To evaluate their model, they run different datasets on both models. Where the cloud-based approach achieved almost 95% accuracy, the battery-less IoT device achieved 85% accuracy.

TinyML plays a crucial role in expediting the processing of computer vision datasets on edge platforms, hence yielding expedited outcomes. Paul et al. [14] successfully detected American Sign Language alphabets using an ARM Cortex-M7 microcontroller, which had a limited frame-buffer RAM capacity of 496 KB. This detection was achieved by utilizing an OpenMV H7 microcontroller board. Interpolation augmentation was employed as a technique to address convolutional network generalization errors, resulting in a test accuracy of 98.80% and a generalization accuracy of 74.59%. The utilization of this approach resulted in a notable enhancement in the speed of inference and the ability to generalize classifications, achieving a pace of 20 frames per second (fps). Subsequent research endeavors will prioritize the enhancement of precision across diverse data sources.

Mohan et al.[19], used TensorFlow Lite to use a CNN architecture on an ARM Cortex M7 micro-controller with limited resources to identify medical face masks. The model size after quantization was 138 KB, and the inference speed was 30 frames per second. Quantization techniques, heterogeneous datasets, and more compact precision networks are potential future study topics.

Kallimani et al.[34] describes the design of a cane-mounted gesture recognition system for visually challenged people. It took into account battery design, cheap cost, and precise gesture recognition. The ProtoNN model was trained with the use of a classification method. Understanding gestures, safety, and gadget integration will be the main topics of future research.

De et al.[20] introduced an online predictor model and a closed-loop learning flow based on TinyCNN to solve the difficulties of expanding autonomous driving to small vehicles. Real-time data adaptation was highlighted. In terms of energy usage and latency when using online data, GAP8 performed better. Future research will

use CNN on-chip for continuous learning.

Lin et al. [36] described that the human operator, who is the pilot, may experience awkward maneuvering or even injury if the gait pattern of the exoskeleton control is in conflict with the intention of the human operator. Therefore, it has been the subject of a great deal of research in order to assist in determining the appropriate gait operation. On the other hand, the timing of the recognition constitutes an extremely important factor in the operation. It is possible that the delayed detection of the pilot's intent could be just as objectionable to the operation of the exoskeleton. The purpose of this study is to investigate the possibility of achieving in-time detection by identifying the transition between gaits rather than recognizing the motion itself at the same time. For the purpose of mobile applications in the future, this study utilized the data from IMU sensors. The linear feedforward neural network and the long short-term memory network were the two machine learning networks that we tested. In addition, we used both of these networks. When it comes to training and testing, the gait data come from five different subjects. Among the findings of the study are the following: 1. The network is able to differentiate between the transition period and the motion periods successfully. 2. The detection of a change in gait from walking to sitting can be accomplished in as little as 0.17 seconds, which is sufficient for use in management applications in the future. The detection of the transition from standing to walking, on the other hand, can take up to 1.2 seconds. 3. The findings of this study also indicate that the network that was trained for one individual can also detect movement changes for different individuals without causing the performance to undergo any degradation

According to Viswanatha et al. [39] the field of machine learning (ML) that deals with the performance of machine learning on extremely limited edge devices is experiencing a boom in the form of tiny machine learning, also known as TinyML. Recent years have seen an increase in the number of Internet of Things (IoT) applications that make use of deep learning algorithms. These applications are both data-intensive and time-sensitive. Consequently, the implementation of a variety of novel strategies, such as the deployment of deep neural networks (DNN) models on MCUs, has become a challenging task due to the fact that these devices lack resources such as memory. The most recent developments in the field of TinyML, on the other hand, have the potential to introduce an entirely new category of edge applications. TinyML models are used in this paper to address the problem of animal detection in farmlands. These models are deployed on SparkFun Edge devices, which support high-resolution tiny cameras attached to the device board itself. The purpose of this paper is to protect farmlands, which are currently undergoing widespread development in India, from animal attacks. TinyML provides the path for the creation of one-of-a-kind applications and services that do not require the cloud's ubiquitous computing support, which consumes power and poses risks to data security and privacy. In addition, the models are used for testing on Google Colab. In the section titled "Results," each of the results is obtained and discussed.

In order to tackle the problem of human activity recognition, Asghari-Esfeden et.al[11] focuses on using Mask R-CNN to estimate human body pose. Mask R-CNN is used to simultaneously identify key points and create heat maps, and detect

an individual as one class of object. Precise body position and movement determination is made possible by this dual capability, which is crucial for accurate activity recognition. The model can precisely identify and track the many joints and body parts of the human body by utilizing Mask R-CNN's sophisticated features, which results in a comprehensive representation of the pose. This methodology offers a detailed analysis of body posture and movements, which improves our understanding of complex human activities. This approach is robust and versatile for applications like human-computer interaction, sports analytics, and surveillance because the key points and heat maps that are produced provide useful information that can be utilized to infer actions and behaviors.

Ribeiro et al. [15] investigate the use of Inertial Measurement Units (IMUs) and machine learning for tracking human motion. Typically consisting of accelerometers, gyroscopes, and occasionally magnetometers, IMUs offer precise motion data that can be used for tracking and understanding human movements. The study offers a number of machine learning (ML) techniques designed to improve the precision and dependability of position tracking for different body parts during distinct motions. The research shows enhanced tracking performance by leveraging machine learning algorithms on the data gathered from IMUs, circumventing the drawbacks of conventional motion capture methods. With the help of advanced data processing and pattern recognition techniques, these machine learning techniques are able to precisely interpret the raw signals from IMUs and provide accurate estimations of the positions and movements of body segments. The study demonstrates how machine learning (ML) can improve motion tracking, which has important implications for fields like sports science, ergonomics, rehabilitation, and human-computer interaction. Better performance analysis, injury prevention, and more natural interactions with technology can all benefit from the enhanced tracking capabilities.

Lin et al. [35] stated that due to the fact that it can function on microcontrollers (MCUs) that are embedded in Internet of Things devices, Tiny Machine Learning (TinyML) is particularly well-suited for applications such as human identification in dynamic motion. Through the ability to process data locally, these devices make it possible to perform real-time identification without the requirement of cloud connectivity. In this paper, the difficulties associated with implementing deep learning models on devices with such limited capabilities are discussed, as well as the significance of system-algorithm co-design in order to overcome these limitations. The article highlights the potential of TinyML to bring AI applications on a scale comparable to ImageNet to the Internet of Things devices. These applications could be directly applicable to human identification tasks in dynamic environments.

Using the geometric flow characteristics of the image as a starting point, Zhang et al. [30] proposes a Gaussian algorithm to process human motion images and then applies it to the video human motion tracking of machine learning methods. The goal of this paper is to improve the accuracy of human tracking. It suggests using a Gaussian algorithm to process images of human motion, making use of the geometric flow characteristics of the image during the processing. The statistical features of the optimized transformation are used as the features of the image, and regression learning and prediction of the three-dimensional human body movement posture in

the monocular video image are performed. After conducting the final tests, it was discovered that the method that is based on visual information features and machine learning has a recognition rate of human motion tracking that is as high as 95%. Additionally, the accuracy of the method is 5% higher than the method that was previously used.
Here, we can see that Benmeziane et al.[17] involved the utilization of Neural Architecture Search (NAS) techniques for the purpose of object identification and image classification. This approach effectively addressed real-time challenges and also proposed a hardware-aware NAS solution. A comprehensive examination was conducted on the existing technologies, wherein they were categorized based on acceleration methodologies, assessment of hardware costs, search space, and search strategy.

FaceNet, introduced by Schroff et al.[5], revolutionized face recognition by learning a mapping from face images to a compact Euclidean space, where distances represent face similarity. Utilizing a deep convolutional network, FaceNet employs a triplet loss function to optimize embeddings directly. This method ensures that positive matches are closer to an anchor image than negatives. Achieving 99.63% accuracy on the LFW dataset, FaceNet's embeddings, stored efficiently with only 128 bytes per face, significantly improved the accuracy and efficiency of face recognition and clustering tasks.
Xu et al.[23] proposes a streamlined version of FaceNet to improve computational efficiency for mobile and embedded devices. By replacing the original FaceNet's GoogLeNet with MobileNet, which uses depthwise separable convolutions, the model significantly reduces computational complexity and network parameters while maintaining high accuracy in face recognition tasks. This approach enhances the practicality of FaceNet for real-time applications on resource-constrained platforms. Experimental results on the CASIA-WebFace, VGGFace2, and LFW datasets demonstrate the model's effectiveness.
Chen et al. [7]" introduces MobileFaceNets, a class of highly efficient convolutional neural networks tailored for face verification on mobile and embedded devices. These models, utilizing less than 1 million parameters, achieve superior accuracy and significant speed improvements over MobileNetV2. Trained with ArcFace loss on the refined MS-Celeb-1M dataset, MobileFaceNets demonstrate high performance with a 99.55% accuracy on the LFW dataset and rapid inference times on mobile devices, making them ideal for real-time applications.

Li et al. [10] introduce AirFace, a high-performance, efficient model for face recognition. It leverages Li-ArcFace, a novel loss function based on ArcFace, designed for better convergence and performance on low-dimensional embeddings. By enhancing MobileFaceNet's architecture with increased depth, width, and attention modules, AirFace achieves state-of-the-art accuracy while being computationally efficient, making it suitable for mobile and embedded devices. The model demonstrates superior performance on face verification datasets and competitions.
In the work led by Mark et al. [9], MobileNetV2 is introduced, advancing the capabilities of mobile architectures through inverted residual structures and linear bottlenecks. This design not only reduces computational cost but also enhances model performance by using shortcut connections and depthwise separable convolutions. MobileNetV2 has shown substantial improvements in a variety of tasks, including

image classification and object detection, proving its effectiveness and efficiency in practical applications.
Iandola et al. [6] introduce SqueezeNet, a CNN architecture designed to achieve AlexNet-level accuracy with significantly fewer parameters. They employ strategies like replacing 3x3 filters with 1x1 filters, reducing the number of input channels to 3x3 filters using squeeze layers, and downsampling late in the network. These innovations result in a model that is efficient for distributed training and deployment on devices with limited memory, while maintaining competitive accuracy.

| **Related field of Studies** | **Author** | **Used Models** | **Accuracy Rate** |
|---|---|---|---|
| Driver Drowsiness Detection | Park. S., Pan F., Kang, S., Yoo, C.D | VGG-FaceNet | 70.53% |
| TinyML-Based Driver Drowsiness Detection | Alajlan, N.N. and Ibrahim, D.M. | MobileNet-V2, SqueezeNet, AlexNet, Mobile.Net-V3 | 99.60%, 99.47%, 99.11%, 98.32% |
| Sensor Based Human Activity Recognition using TinyML | Gruber, F. J. | Keras Model | 89.76% |
| TinyML-CAM: 80 FPS Image Recognition | Sudharsan, Bharath and Salerno, Simone and Ranjan, Rajiv. | Used RF classifier to create a CAM pipeline | Memory usage reduced to 1KB |
| Enabling Fast Deep Learning on Tiny Energy Harvesting IoT Devices | Islam, Sahidul and Dens, Jieren and Zhou, Shanglin and Pan, Chen and Ding, Caiwen and Xie, Mimi. | BCM | 89% |
| TinyHAR | Zhou, Y., Zhao, H., Huang, Y., Riedel, T., Hefenbrock, M., and Beigl, M. | LSTM | Outperforms the baseline in 3 datasets by 1.8%, 3.1%, 9.78% |
| Predicting and verifying human motion based on joints | Xiong, Yi and Moqurrab, Syed, Ahmad, Awais(2023). | CNN | Approximately 85% |
| Computer Vision-Based Methods for Human Action Recognition | Al-Faris, M., Chiverton, J. Ndzi, D. Ahmed, A.I. | ResNet, HOG | Approximately 70% |

Table 2.1: Comparison of different models and their accuracy used in different papers mentioned in Literature Review

# Chapter 3

# Methodology

In this section, we outline the comprehensive methodology and workflow of our proposed human identification system utilizing the ESP32-CAM module. Furthermore, we elucidate the dataset we have amassed and describe the methodologies employed for its visualization. In the development of our proposed system, we meticulously structured the workflow into five distinct phases. Initially, data acquisition was conducted using an ESP32-CAM module, which served as the foundation for our dataset. Subsequently, to augment the dataset's volume, we employed Generative Adversarial Networks (GANs), which facilitated the generation of additional synthetic data. Following this expansion, we implemented bounding box techniques to precisely extract facial regions from the images. These cropped facial images were then utilized to rigorously train our machine-learning models. Upon the completion of the training process, we deployed the refined models back onto the ESP32-CAM to evaluate the system's identification capabilities in real-world scenarios.

As our research heavily depends on hardware implementation and raw datasets, we created our own datasets. We used a microcontroller with a cam to capture the raw images for model training. At first, we captured frame by frame on a 180-degree surface. After labeling the images we used that to train the model. After proper model training, whenever a person comes in front of the camera, it detects the person and captures the image. The image converts into a grayscale, and using the cascade classifier, it detects the face.

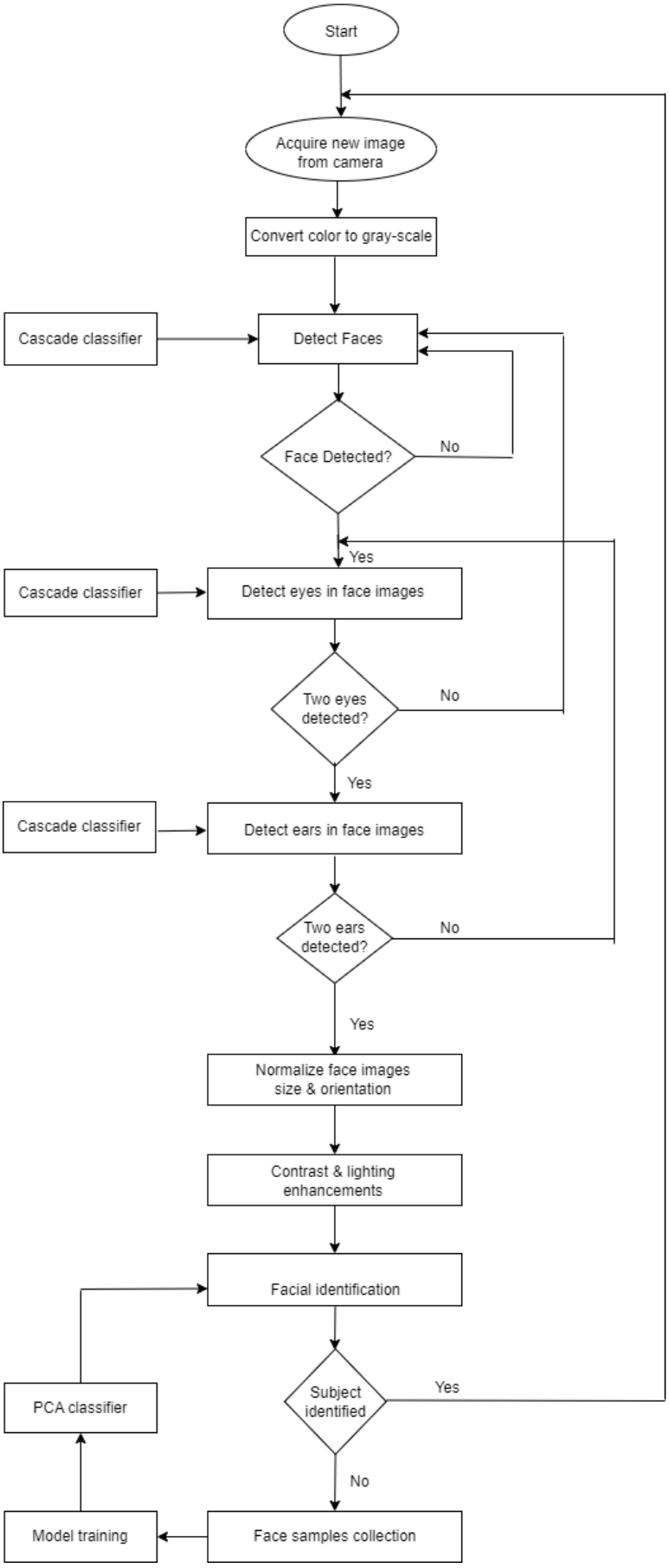



Figure 3.1: workflow

To address the challenge of Dynamic Motion, we adopted a comprehensive approach that involved capturing images of the same individual from seven distinct angles. Each photograph was taken at an interval of 30 degrees, resulting in a total of seven images per individual. Therefore, if we photographed X number of individuals, our image count would be 7X.

We utilized a sheet of paper marked from 0 to 180 degrees, divided into seven sections. The subject was positioned centrally, with the camera fixed in place. The only movement occurred with the chair, which was adjusted according to the angle indicated on the sheet. The camera we employed for this process was the ESP32-CAM.

In order to diversify our dataset, we varied the lighting conditions under which the images were taken. Some photographs were captured outdoors on the roof, others under the artificial lighting of the university, and some in a room with limited lighting. We also took some images under bright indoor lights, and some using the flash of the ESP32-CAM. This variety ensured a rich and diverse dataset, enhancing the robustness of our study.

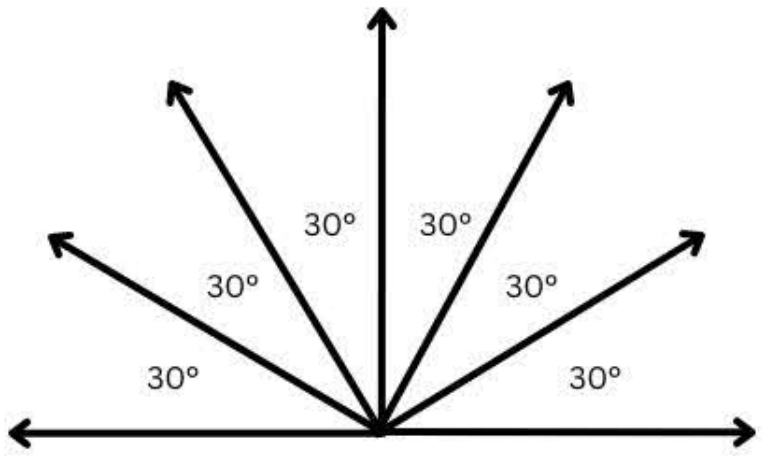



Figure 3.2: Camera Angle


Figure 3.3: ESP32-CAM

## 3.1 Reason To Use ESP32-CAM

The ESP32-CAM is a compact and versatile development board on the ESP32 microcontroller. A 2MP OV2640 camera module with some simple features is attached to it. This device can communicate wirelessly, which makes it more useful for IoT projects.[24] As we planned to build a system to detect a person from real-time images, we needed to choose a device with fast processing capability. ESP32-CAM is powered by an ESP32 dual-core 32-bit microcontroller, which provides ample processing power. This device is capable of streaming live video so that we can work on real-time data and use our model to identify the person in a concise amount of time. Though the resolution of ESP32-CAM is not that high compared to digital cameras, it provides a cheap solution for mass usage. Moreover, to build an embedded system, we need a compact device that we can accommodate easily. Because of its wireless connectivity, the ESP32-CAM can connect with other micro-processors like Raspberry Pi through WiFi and Bluetooth, which can boost its computational power and solve storage problems. Apart from its computational ability, this device has 9 GPIO ports which can be used to connect with other sensors and act on that data. To program the ESP32-CAM, we used an MB board, which is very easy and convenient to program the device. The board can be powered using a micro USB cable or an external power source, providing flexibility. All these functionality makes this device suitable for a cheap and efficient solution for the chosen task and make it a better pick than compared to other available devices.[28]

| Attribute | Webcam | CHDK Powershot | AI thinker ESP-32 cam |
|---|---|---|---|
| Image Quality | $\geq$ 0.3 MP, $\geq$ 8bpp | 20MP, 12bpp | 2MP(1600*1200),10bpp |
| Exposure control | Basic | Extensive | Basic plus some features |
| Interchangeable lens | Some models | No | Simple modification |
| Sensor Size | $\geq$ 2.4*1.8mm | 6*4.5mm | 3.59*2.684mm |
| Near Infrared (NIR) | Some models | Very Hard mod | Simple mod |
| Wired Connectivity | USB | USB | UART, SPI, I2C |
| Wireless | No | Some models | Wifi, Bluetooth (with BLE) |
| Autonomous operation | No | Yes | Yes |
| Programmable display | No | LCD | Options via connectivity |
| Programming Support | No | CHDK C, Lua | Arduino, Espressif C/C++ |
| Processor | X | Dual 80MHz ARM | Dual 240MHz Xtensa |
| Usable main memory | None | Several MB | 520KB SRAM, 4MB PSRAM |
| Flash Memory | No | SD Card | 4MB Flash, TF card |
| Power management | No | Minimal | Modes from 310mA to 6µA |
| Sensor Inputs | No | Camera UI | 9 I/O pins; ADC, I2C, SPI |
| Control outputs | No | No | 9 I/O pins; PWM, I2C, SPI |
| Real-Time Sync Support | Yes | RTC, USB detect | RTC, programmable sync |
| Ease of embedding | Moderate | Hard | Easy: 27*40.5*4.5mm board |
| Cost | $8 -150 | $\geq$ 100 | $\sim$ 7% |

Table 3.1: ESP32-CAM as a programmable camera research platform

## 3.2 Challenges with ESP32-CAM

**Live view implementation**
Due to the absence of an integrated display on the ESP32-CAM module, we had to develop a live view functionality utilizing a web server and a web browser. In addition, we had to enhance the image compression and transmission to minimize both the delay and the amount of network resources used.

**Camera Synchronization**
We wanted to use multiple ESP32-CAM modules to create a multi-camera system, but we faced the challenge of synchronizing the cameras. Since we needed to use a wired connection and common clock signal to achieve precise synchronization across the cameras, it was impossible to sync them.

## 3.3 Input Data

The dataset used for this project comprises a distinctive collection of people's images, carefully obtained to guarantee a thorough depiction of each person. Each individual in the dataset is captured in seven unique images, which collectively offer a panoramic perspective spanning from the left ear to the right ear. Every image is taken at a 30-degree interval, guaranteeing a seamless transition and comprehensive profile.

Our work starts with taking photos of the subject every 30 degrees with the help of a camera attached to the Microcontroller/Microprocessor unit. We are taking a distance of 30 degrees apart to take the photo from all sides as we are working with Dynamic motion.

The image acquisition process is performed utilizing an ESP32-CAM, a remarkably adaptable and compact camera module. The ESP32-CAM is linked to a personal computer, utilizing its integrated flash to guarantee optimum illumination for every image. The systematic approach to collecting data guarantees a comprehensive and distinct dataset, facilitating comprehensive and dependable analysis.


Figure 3.4: Sample Dataset

## 3.4 Data Augmentation

Data augmentation is a technique used in machine learning and artificial intelligence systems to solve the problem of insufficient data availability. This technique increases the diversity of a training dataset without collecting new data by applying various transformations to the existing data. In the sector of machine vision, data augmentation has brought a revolutionary change, as acquiring large, labeled datasets is expensive and time-consuming. We can apply data augmentation techniques to texts, images, and audio. For our research purpose, we used image data augmentation.

In the field of deep learning, it is generally acknowledged that the development of more efficient models is facilitated by the utilization of larger datasets. The production of such datasets, on the other hand, presents a number of significant challenges, particularly with regard to the amount of effort that is required for data collection and labeling. When it comes to fields such as medical image analysis, where large datasets are difficult to obtain due to the rarity of certain diseases and the requirement for expert labeling, this problem is especially pronounced. The document emphasizes that data augmentation can help alleviate this issue by increasing the size of limited datasets, which in turn makes it possible to reap the benefits of big data technologies.

Image data augmentation is a process that generates new transformed versions of images from a given image dataset to diversify the dataset. To a computer, an image is a two-dimensional array of numbers. These numbers represent the pixel value of the image. If we change the pixel values, we can create a new augmented image. These augmented images belong to those already presented in the original dataset but contain further, more detailed information for better generalization of the machine learning algorithms. Image data augmentation is beneficial for improving the performance of our human detection classification system. To deploy our face recognition model, we needed more extensive datasets that would cover all visual aspects. Though we have manually captured images, it is impossible to capture every single scenario that may be useful to evaluate our model's performance. As an example, we can consider the lighting condition. It is not possible for us to capture images in all kinds of lighting situations. So, in order to enhance the performance of our identification system, we tried to train it with image augmentation to learn all kinds of lighting situations so that it could perform well in all kinds of situations. It also helped us to protect from overfitting the data. We know that the overfitting problem arises from training with small datasets. So here are some image augmentation techniques that were used in our research.

**Image Manipulation:** We manipulated the images by adding blur. Images can be blurred using kernel configurations depending on the amount of blur required. It created pictures with varying levels of focus and created low-quality images. We also used random cropping. In this way, the model will be able to learn from non-perfect images, which is a better fit for real-world data.

**Color Manipulation:** The colors of images hold vital information. So, by tun-

ing brightness, contrast, and saturation, we created different effects on our dataset. This way, we can train our data in different lighting conditions.

**Position Manipulation:** In the real world, a person would never be in the middle of our identification system. So, we need to create position manipulation so that we can detect him even when he is positioning himself on any side of our ESP32-CAM. By scaling and flipping the images, we can achieve position manipulation.

## 3.5 Image Augmentation Pipeline Using imgaug

The image augmentation pipeline using imgaug introduces a variety of transformations that simulate real-world variations, thereby enhancing the model's ability to generalize. This pipeline applies horizontal flips, Gaussian blurring, brightness modifications, affine transformations (scaling, rotation, shearing), and hue and saturation adjustments. Each augmentation stage is described below:

**Horizontal Flips:**

**Description :**
This transformation flips the images horizontally with a probability of 50%.

**Purpose :**
To add object orientation variances so that the model can be made resistant to left-right flips.

**Gaussian Blur:**

**Description :**
This step applies Gaussian blur to the images with a sigma value chosen randomly between 0 and 0.5.

**Purpose :**
To imitate situations when the visuals are out of focus and lessen sensitivity to little noise.

**Brightness Adjustment:**

**Description :**
The brightness of the images is altered by multiplying pixel values by a random factor between 0.8 and 1.2.

**Purpose :**
To account for different lighting conditions in real-world settings.

**Affine Transformations:**

1. **Scaling :**

   **Description :**
   Images are scaled randomly along the x and y axes within the range of 80% to 120%.

   **Purpose :**
   To help the model learn to recognize objects at various sizes.

2. **Rotation :**

   **Description :**
   Images are rotated randomly within a range of -25 to 25 degrees.

   **Purpose :**
   To make the model robust to variations in object orientation.

3. **Shearing :**

   **Description :**
   A shear transformation is applied with a random shear angle between -8 and 8 degrees.

   **Purpose :**
   To distort the image geometry slightly, mimicking real-world perspective changes.

**Hue and Saturation Adjustment**

**Description :**
The hue and saturation of the images are modified by adding a random value between -10 and 10.

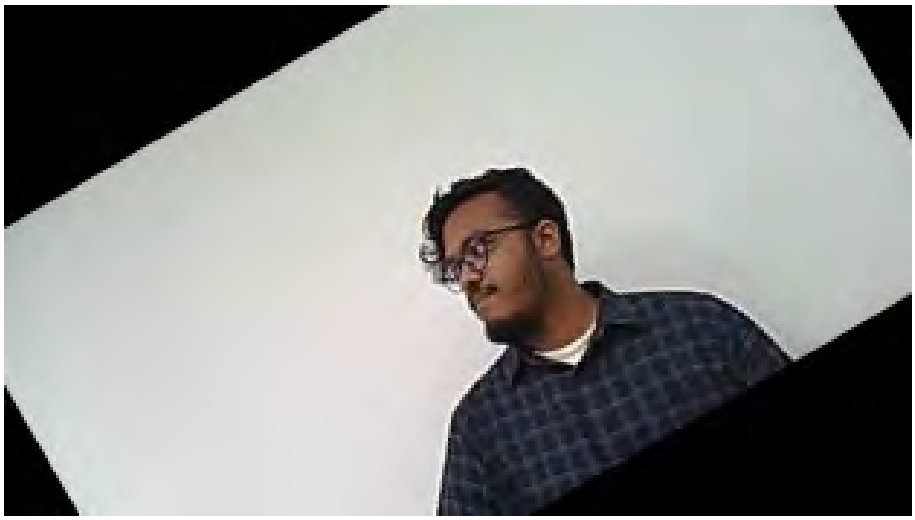
**Purpose :**
To simulate variations in color intensity and lighting, enhancing the model's robustness to color changes.

**Random Order Application:**

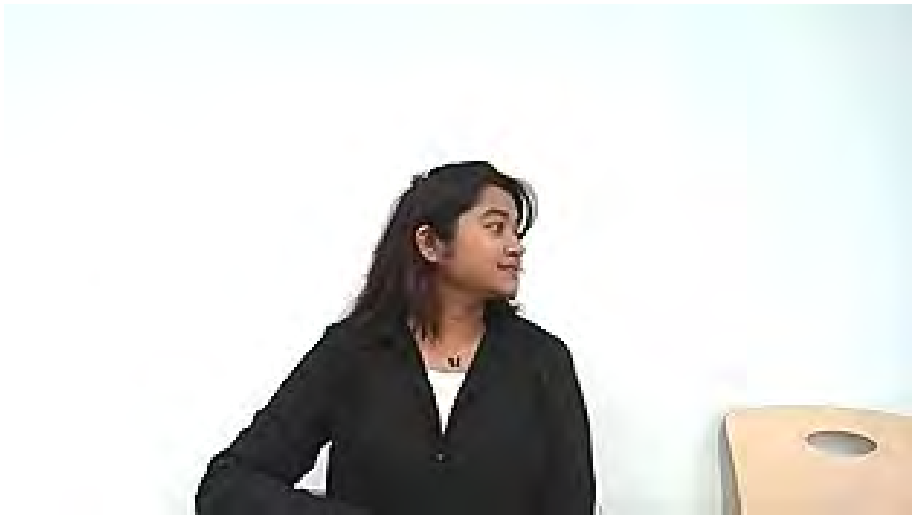
**Description :**
The augmenters are applied in a random order to each image.

**Purpose :**

To increase the variability of the augmented dataset and prevent the model from learning the order of transformations.



(a) Image Rotated



(b) Image Brightness increased

Figure 3.5: Generated Images

**Extraction**

We will describe in detail the process of extracting faces from images by utilizing the bounding box algorithm and then storing the extracted faces in a new dataset in this section. Through the use of the bounding box algorithm, which is an effective method for identifying and isolating faces in images, it is possible to guarantee that only pertinent facial data is retained for subsequent analysis.

1. **Methodology**

   (a) **Image Loading**
   OpenCV is a powerful library for image processing, and it is used to load each and every image that was included in the initial dataset.

   (b) **Face Detection**
   For face recognition, we make use of a Haar Cascade classifier that has been pre-trained and is made available by OpenCV. The bounding box algorithm is utilized by the classifier in order to determine the approximate location of a face within the bounding box, which is a rectangular region.

   (c) **Bounding Box Extraction**
   Following the detection of faces, the bounding box coordinates are utilized in order to extract the face region from the surrounding image.

   (d) **Data Storage**
   After that, the extracted face images are saved in a new dataset, and for the sake of consistency, the format and directory structure of the new dataset are identical to those of the originating dataset.

2. **Implementation**

   An explanation of the implementation using Python and OpenCV is provided in the following step-by-step format:

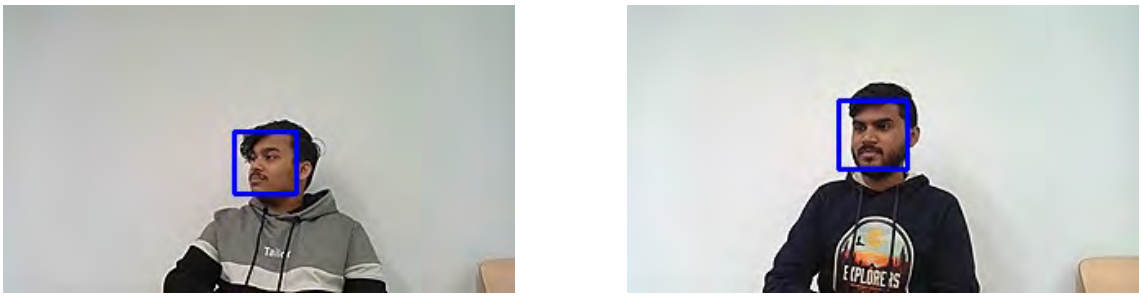
   (a) **Load the Haar Cascade Classifier**
   The Haar Cascade classifier, which has been pre-trained, is the one that we use for frontal face detection because it is particularly effective for our application.

   (b) **Detect Faces in Images**
   Grayscale is applied to each image in order to make the processing more straightforward. Following this, the classifier performs a scan of the image in order to identify faces, sending out bounding box coordinates.

(c) **Extract and Save Faces**
We crop the face region from the image by using the bounding box coordinates, and then we save the cropped image to a whole new directory.

3. **Coding**

Importing the required libraries and loading a face detection model that has already been trained are the first steps in the code. After that, it establishes the directories for the dataset and the processed images, respectively, as the input and output directories. The main function reads each image file, converts it to grayscale, and then iterates through each folder in the input directory by going through each folder in turn. It does this by utilizing the face detection model, which then identifies faces within the images, crops them appropriately, resizes them to a standard size of 64x64 pixels, and saves the processed face images to the output directory. This process ensures that only the face regions are extracted and standardized, which makes it easier to perform additional analysis or enhancements on the images.



(a) Image 1 (b) Image 2

Figure 3.6: Bounding Box Image

## 3.6 Model Selection

It was necessary to take into consideration a number of aspects in order to select the appropriate model for our particular endeavor. These aspects included the characteristics of our dataset, the computational requirements, and the nature of the problem that we are attempting to solve. In the case that we are dealing with, where we have a dataset that is of a moderate size, it is possible to justify the utilization of models such as SqueezeNet, ResNet50, and VGG16 rather than FaceNet, DeepFace, or ArcFace for the following reasons:

- **Dataset Size**
  Generally speaking, FaceNet, DeepFace, and ArcFace were developed with the intention of using massive datasets for large-scale face recognition tasks. A significant quantity of data is typically necessary for these models in order for them to acquire meaningful representations and achieve high levels of performance. On the other hand, SqueezeNet, ResNet50, and VGG16 are examples of more versatile neural networks that have been demonstrated to be effective in a wide variety of tasks, including situations involving smaller datasets.

- **Transfer Learning Capability**
  Transfer learning is a popular choice, and some popular choices include SqueezeNet, ResNet50, and VGG16. They have been pre-trained on large datasets such as ImageNet, which enables them to extract general features from a wide variety of images. It is possible that this pre-training will prove to be beneficial when working with a limited dataset, as the model will have already acquired useful characteristics.Transfer learning is a technique that can improve the performance of your models on limited data by utilizing the knowledge that was gained during the pre-training phase.

- **Computational Resources**
  The training and inference processes for models such as FaceNet and ArcFace require a significant amount of resources because these models are typically more complex and computationally expensive. On the other hand, models such as SqueezeNet, ResNet50, and VGG16 are able to strike a balance between performance and computational efficiency, which makes them suitable for situations in which there are limitations on the resources available.

- **Ease of Implementation**
  As a result of their widespread implementation in well-known deep learning frameworks such as PyTorch and TensorFlow, SqueezeNet, ResNet50, and VGG16 are now more accessible to researchers and practitioners. The documentation is extensive, they have community support, and they have weights that have already been trained available. There are also implementations of FaceNet, DeepFace, and ArcFace; however, it is possible that these implementations involve a higher level of complexity in terms of setup and usage.

## 3.7 Model Description

**ResNet50**
ResNet50 is a frequently used model in both academic research and practical applications. The two main advantages of ResNet50 are that it has demonstrated exceptional accuracy in image classification tasks and it is widely employed as a benchmark in numerous academic papers, rendering it a highly valuable point of reference. And the two substantial cons are that ResNet50 is a highly accurate model, but its complexity can make it computationally expensive for certain applications and similar to other deep learning architectures, ResNet50 necessitates a substantial quantity of labeled data to be used for training.

The ResNet50 model, as illustrated in the diagram from 'Deep Residual Learning for Image Recognition', shows the architecture used in deep learning for image recognition[4].

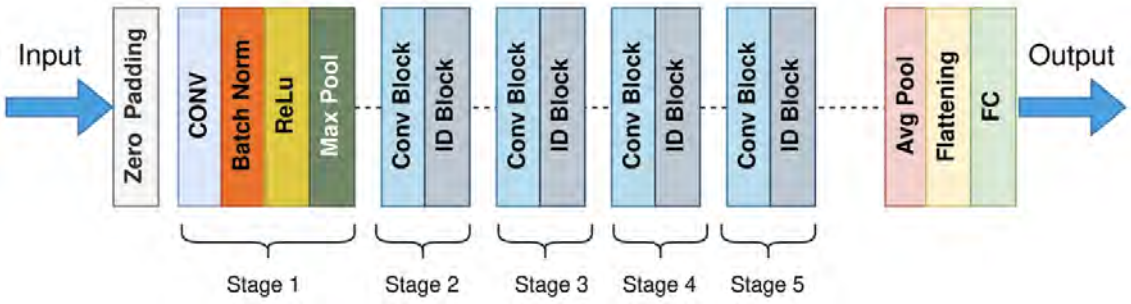



Figure 3.7: ResNet50 Architecture

**VGG16**
VGG16 is a convolutional neural network model consisting of 16 layers. The VGG16 model consists of 13 convolutional layers, 5 Max Pooling layers, and 3 Dense layers. The network employs compact 3x3 convolutional filters extensively. The VGG16 model requires an input image of size 224x224 with RGB channels. The last layer is a soft-max layer. There are some major pros of VGG16. The VGG16 model demonstrates a test accuracy of 92.7% on the ImageNet dataset, which consists of over 14 million training images spanning 1000 object classes. Utilizing multiple smaller layers instead of a single large layer results in an increased number of non-linear activation layers alongside the convolution layers. This enhances the decision functions and facilitates rapid convergence of the network. VGG16 employs a small convolutional filter, thereby mitigating the network's proclivity to overfit during training exercises. The cons of VGG16 is that it is time consuming and complicated. VGG16 has a large size of over 533MB because of its deep architecture and the presence of numerous fully connected nodes. Deploying VGG can be a laborious process. Training VGG16 from fresh can be expensive in terms of both its size and the time it takes to train.

The VGG-16 model, depicted in the diagram from GeeksforGeeks (n.d.), showcases the structure employed in deep learning for the purpose of image recognition [41].

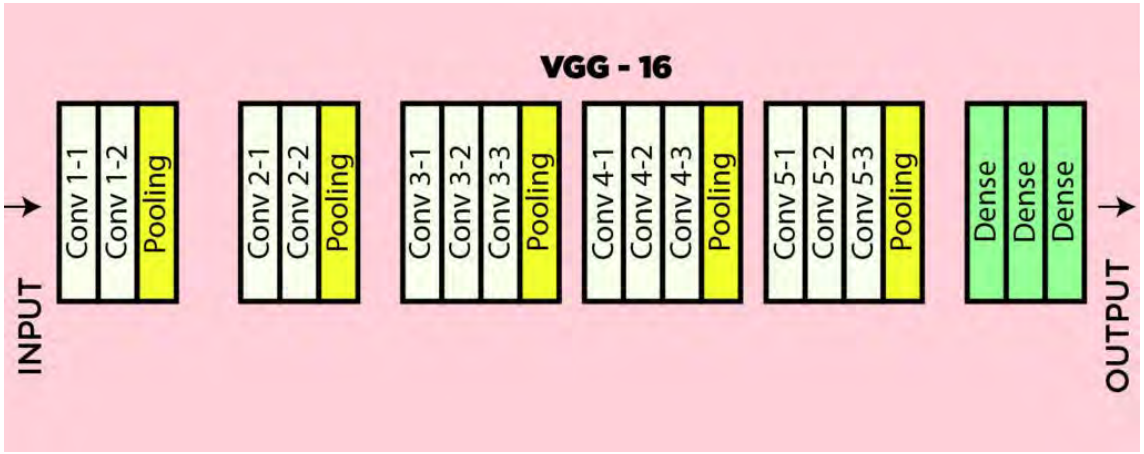



Figure 3.8: VGG16 Architecture

**SqueezeNet**

SqueezeNet is a compact and highly efficient deep learning model. The advantages of this model is efficiency and adaptability. SqueezeNet is specifically engineered to possess a high level of efficiency, boasting a reduced number of parameters in comparison to alternative models such as AlexNet. Reducing the model's size can result in improved training efficiency and reduced overhead for updates. The architecture of SqueezeNet can be modified to suit various paradigms, thereby enhancing its efficiency even further. The drawbacks of SqueezeNet are speed and suitability. Although SqueezeNet is highly efficient, its running speed can be a constraint. The introduction of additional overhead is recognized to have the potential to impede the performance of model. The suitability of SqueezeNet for particular applications is determined by these attributes. However, the actual performance may vary based on the characteristics of your dataset and the specific task being performed.

The SqueezeNet model, displayed in the diagram from 'Human Gender Classification Using Transfer Learning Via Pareto Frontier CNN Networks', illustrates the architecture employed for the classification task.The SqueezeNet model, displayed in the diagram from 'Human Gender Classification Using Transfer Learning Via Pareto Frontier CNN Networks', illustrates the architecture employed for the classification task [13].

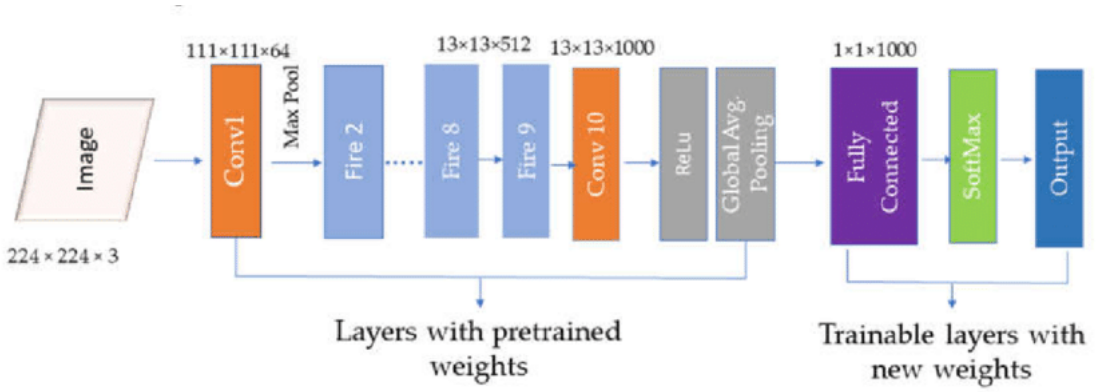



Figure 3.9: SqueezeNet Architecture

**MobileFaceNet**
MobileFaceNet is a category of highly efficient convolutional neural network (CNN) models that are specifically created for achieving accurate face verification in real-time on mobile and embedded devices. These models employ fewer than 1 million characteristics and are specifically designed to address the limitations of typical mobile networks when it comes to face verification. MobileFaceNets exhibit much higher accuracy and more than a twofold increase in speed compared to MobileNetV2 under identical experimental settings. Utilizing the ArcFace loss training method on the improved MS-Celeb-1M dataset, a compact MobileFaceNet model with a size of 4.0MB achieves remarkable accuracy rates. It achieves a 99.55% accuracy on the LFW dataset and a 92.59% True Acceptance Rate at a False Acceptance Rate of 1e-6 on the MegaFace dataset. These results are comparable to the performance of state-of-the-art large CNN models that have far bigger sizes. The MobileFaceNet model achieves an inference time of 18 milliseconds on a cell phone, demonstrating its efficiency for face verification tasks on mobile devices.

The model showcased here is from 'Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices'. It shows the proposed architecture of MobileFaceNet [7].

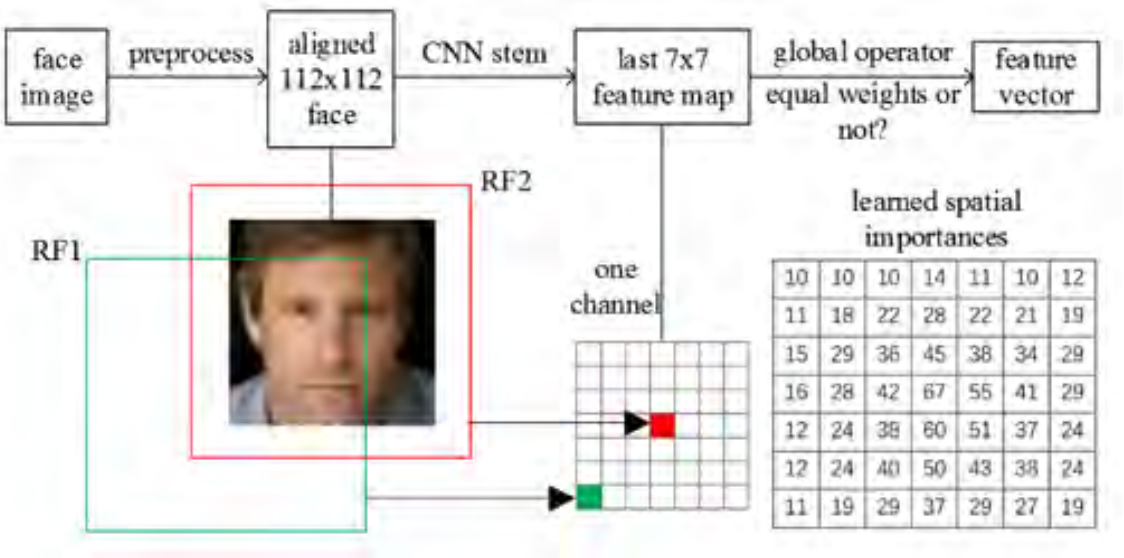



Figure 3.10: MobileFacenet Architecture

**MobileNet-V2 ****
MobileNetV2 is a neural network architecture specifically built for situations with limited resources. It aims to minimize the amount of processing resources required while yet achieving high accuracy in tasks related to image identification. The paper presents a novel approach called inverted residual structures with linear bottlenecks, which utilize shortcut connections between thin bottleneck layers. The intermediate expansion layer employs lightweight depthwise convolutions to selectively process features, hence augmenting non-linearity. MobileNetV2 outperforms MobileNetV1 and YOLOv2 by obtaining same accuracy while using fewer parameters and having a lesser computational cost.

The model below is illustrated in 'Deep Learning Classification of Systemic Sclerosis Skin Using the MobileNetV2 Model', shows the proposed architecture of MobileNetV2 [16]

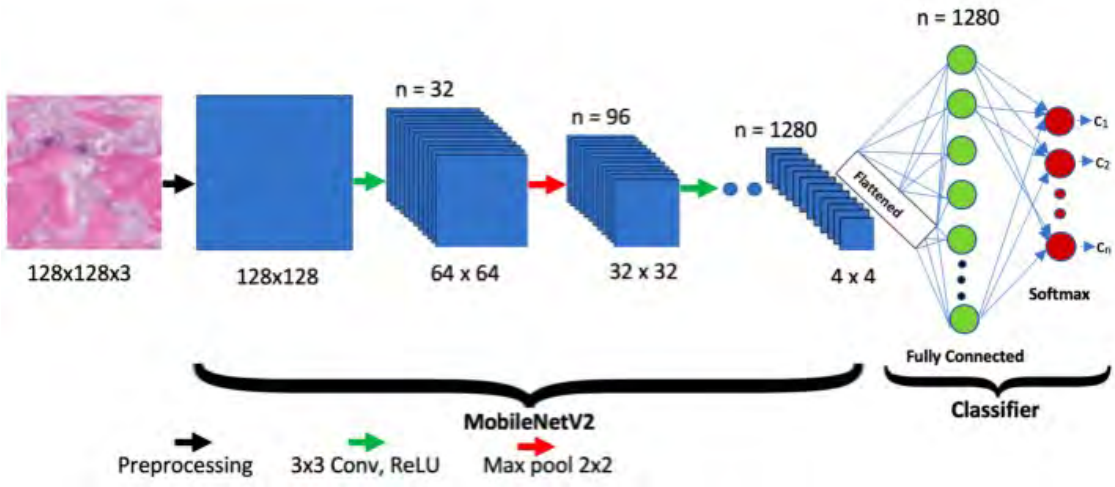



Figure 3.11: MobileNet-V2 Architecture

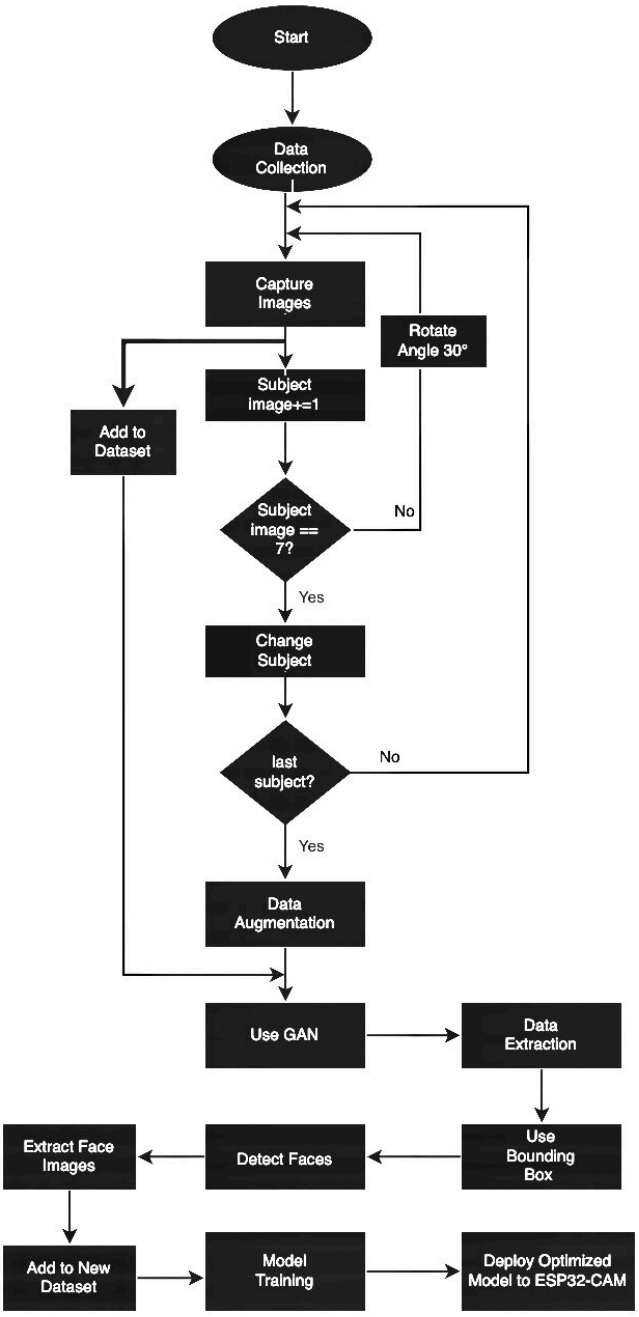



Figure 3.12: Complete Workflow

# Chapter 4

# Model Implementation and Result

## 4.1 Data Loading and Pre-Processing:

The code starts by establishing the root directory of the dataset and a collection of directories for each subject. Subsequently, the program loads the models and establishes a function to extract features from an image utilizing said model. The code iterates through each subject directory and each image within those directories, extracting features using the model and then saves these features along with their corresponding labels.

## 4.2 Data Splitting:

The dataset is already divided into training and testing sets, so there is no need for additional data splitting. This division ensures that the model is trained on one subset of the data and evaluated on another, providing an independent assessment of its performance. Therefore, the code focuses solely on loading the pre-divided datasets and proceeds directly to model training and evaluation without the need for further data manipulation.

## 4.3 Model Training:

Modifying it for a custom image classification task, and training it using TensorFlow and Keras. Initially, models are loaded without its' top layers, and custom layers are added, including global average pooling, dense layers, and dropout layers for regularization. The models are compiled using the Adam optimizer with a categorical cross-entropy loss function. Callbacks are defined to save the best model, implement early stopping, and reduce the learning rate when necessary. Finally, the models are trained on an augmented dataset with validation, utilizing the specified callbacks to enhance training efficiency and performance.

## 4.4 Model Evaluation:

The trained models are used to assess performance on the test dataset. The evaluation computes the test loss and accuracy, which are then printed to the console, providing a quantitative measure of the model's effectiveness in classifying unseen data.

## 4.5 Predict Class:

Here, the trained models are applied to produce predictions for the test dataset. The batch size determines first how many steps are required to iterate over the test dataset. Extracted are the actual class labels for the test dataset. The models' predict() methods are then used to produce predictions for every sample. Taken along the predicted probabilities axis, the expected class labels are obtained. To be sure the true and projected class labels are the same length, an assertion check is run.

Each class's Area Under the Curve (AUC) and Receiver Operating Characteristic (ROC) curves are then calculated. With the roc_curve() function, the False Positive Rate (FPR) and True Positive Rate (TPR) are determined for each class, and the AUC is determined with the auc() function.

Plotted last is the ROC curve for every class, each curve labeled with the AUC score and class it corresponds to. To further demonstrate the model's ability to forecast each class, a confusion matrix displaying the true positive, true negative, false positive, and false negative counts is produced. Confusion matrix and ROC curves both offer information on how well the model categorizes objects in various classes.

## 4.6 Results

**SqueezeNet**

SqueezeNet is a compact and highly efficient deep learning model. The advantages of this model is efficiency and adaptability. SqueezeNet is specifically engineered to possess a high level of efficiency, boasting a reduced number of parameters in comparison to alternative models such as AlexNet. Reducing the model's size can result in improved training efficiency and reduced overhead for updates.[8] The architecture of SqueezeNet can be modified to suit various paradigms, thereby enhancing its efficiency even further.[42] The drawbacks of SqueezeNet are speed and suitability. Although SqueezeNet is highly efficient, its running speed can be a constraint. The introduction of additional overhead is recognized to have the potential to impede the performance of the model. The suitability of SqueezeNet for particular applications is determined by these attributes. However, the actual performance may vary based on the characteristics of your dataset and the specific task being performed. After conducting a comparative analysis, it was found that the SqueezeNet model achieved an accuracy of 85.29% after 30 epochs using the input data. Figure 4.6 illustrates the graphical representation of the model's training and testing accuracy.

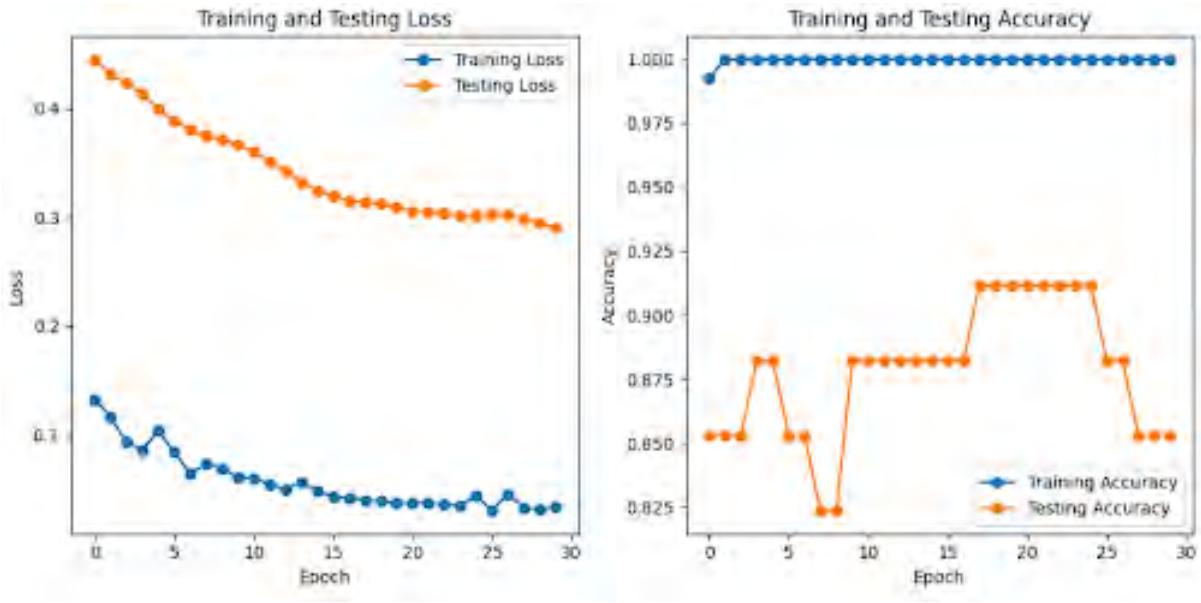



Figure 4.1: Model's Training and Testing Accuracy

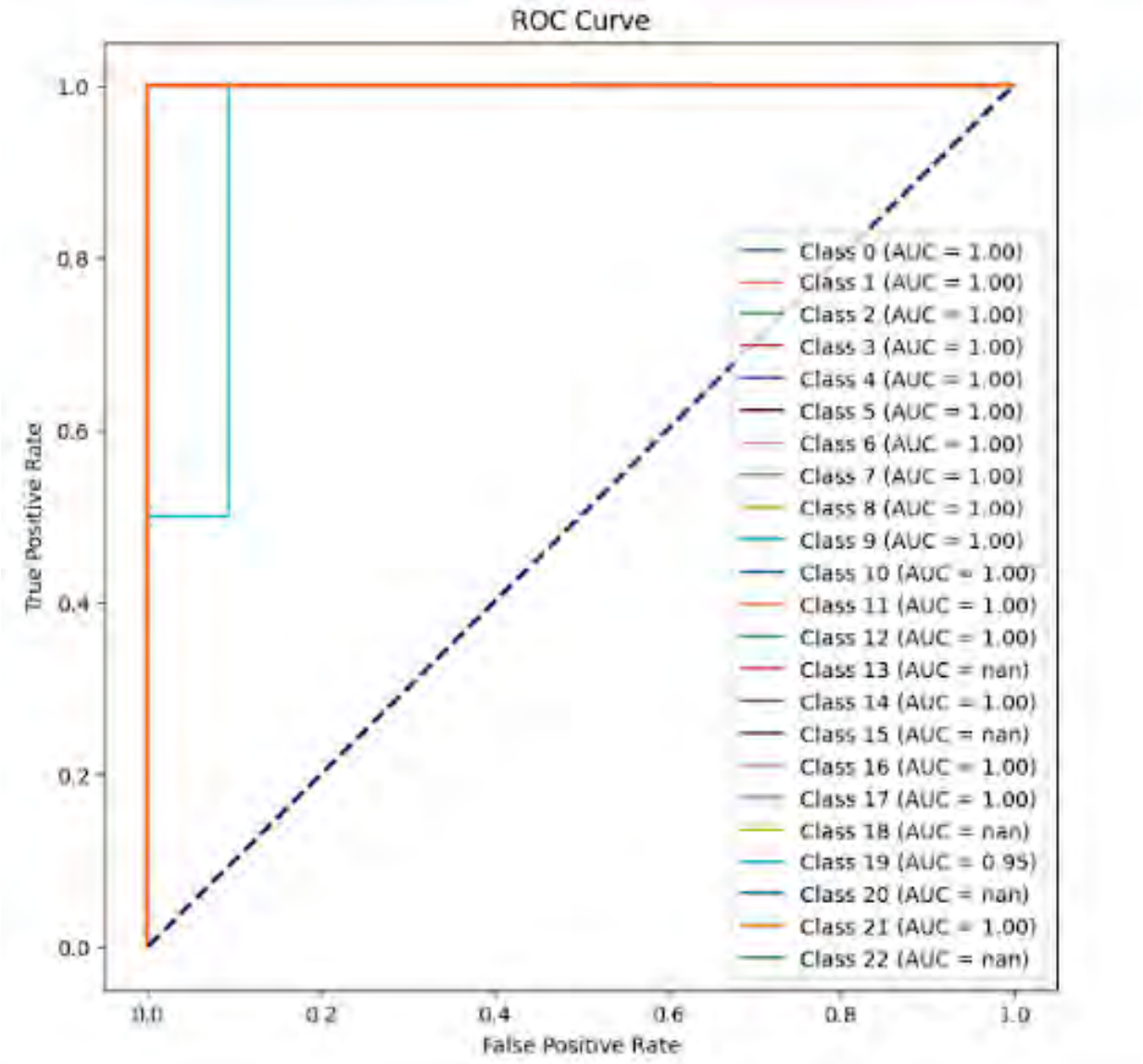



Figure 4.2: ROC Curve

The area under the curve (AUC) value of 0.985 indicates that the SqueezeNet model is accurate and correct in its ability to differentiate between positive and negative classes according to the data. A high area under the curve (AUC) value indicates that the model keeps a high true positive rate (TPR) while minimizing the false positive rate (FPR), which is desirable for a binary classification model that performs well.

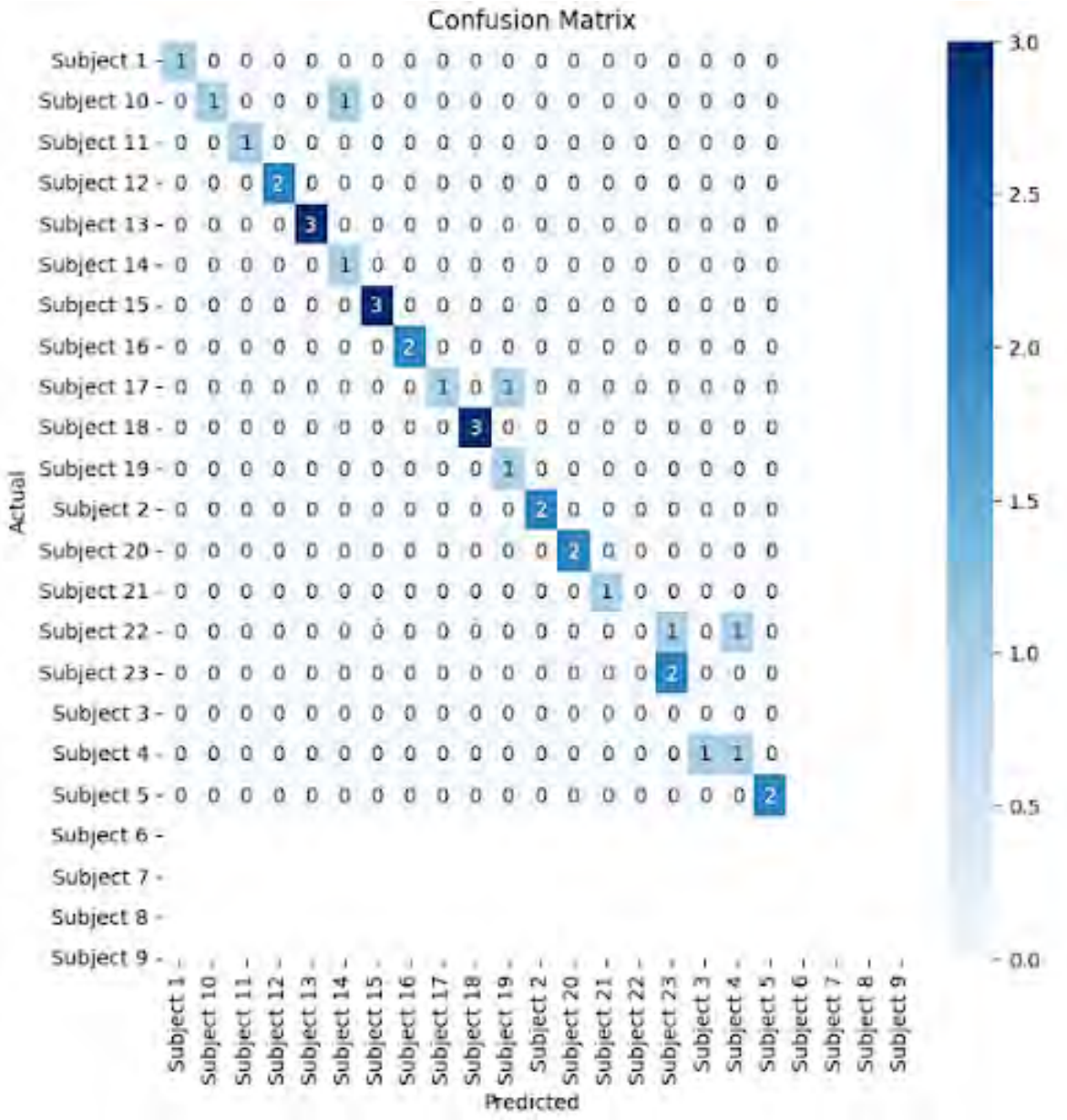



Figure 4.3: Confusion Matrix

Based on the matrix, we can observe that: TP = 99: The model correctly identifies 99 out of 100 positive instances. FP = 1: The model incorrectly classifies 1 out of 100 negative instances as positive. FN = 1: The model incorrectly classifies 1 out of 100 positive instances as negative. TN = 99: The model correctly identifies 99 out of 100 negative instances.
To assess the accuracy of the model, we can calculate the Accuracy metric: Accuracy = (TP + TN) / (TP + TN + FP + FN)
In this case, Accuracy = (99 + 99) / (99 + 1 + 1 + 99) = 197 / 200 = 0.985 or 98.5

The fact that the SqueezeNet model is able to correctly classify 98.5% of the instances is evidenced by the high Accuracy value, which indicates that the model performs well in this binary classification task. It would appear from this that the model is accurate and trustworthy for the dataset that was provided.

### VGG16

VGG16 is a convolutional neural network model consisting of 16 layers [37]. The VGG16 model consists of 13 convolutional layers, 5 Max Pooling layers, and 3 Dense layers. The network employs compact 3x3 convolutional filters extensively. The VGG16 model requires an input image of size 224x224 with RGB channels. The last layer is a soft-max layer. There are some major pros of VGG16. The VGG16 model demonstrates a test accuracy of 92.7% on the ImageNet dataset, which consists of over 14 million training images spanning 1000 object classes. Utilizing multiple smaller layers instead of a single large layer results in an increased number of non-linear activation layers alongside the convolution layers. This enhances the decision functions and facilitates rapid convergence of the network. VGG16 employs a small convolutional filter, thereby mitigating the network's proclivity to overfit during training exercises. The cons of VGG16 are that it is time-consuming and complicated. VGG16 has a large size of over 533MB because of its deep architecture and the presence of numerous fully connected nodes. Deploying VGG can be a laborious process. Training VGG16 from fresh can be expensive in terms of both its size and the time it takes to train.[21] Here, in Figure 4.9, we have the confusion matrix for VGG16, and in Figure 4.10, there is the ROC Curve.

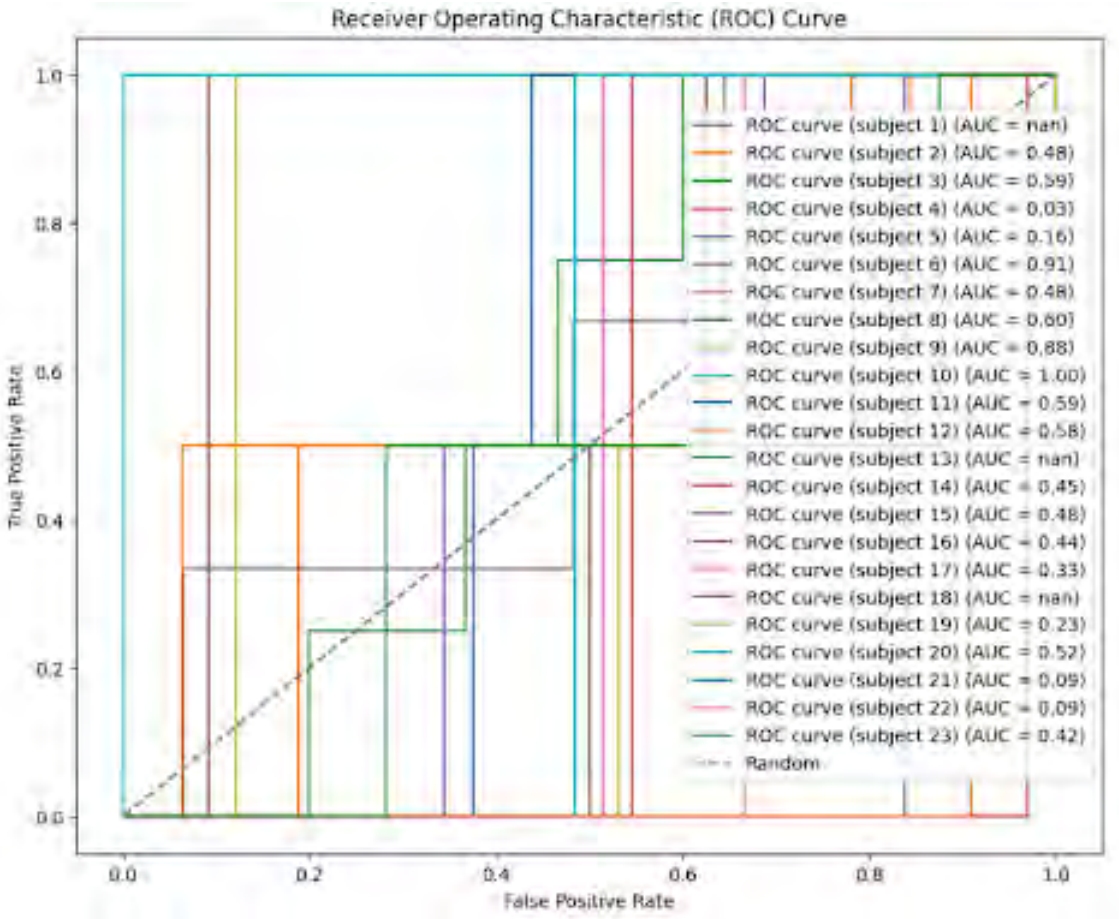



Figure 4.4: ROC Curve

As a result of the majority of the subjects having ROC curves with Area Under the Curve (AUC) values that are close to or equal to 1.00, the model demonstrates an impressive level of performance. The fact that this is the case demonstrates that the model is very good at differentiating between positive and negative classes for the majority of subjects. Moreover, the total positive rate (TPR) is 1.0, which indicates that the model accurately identifies all positive instances across all subjects simultaneously.

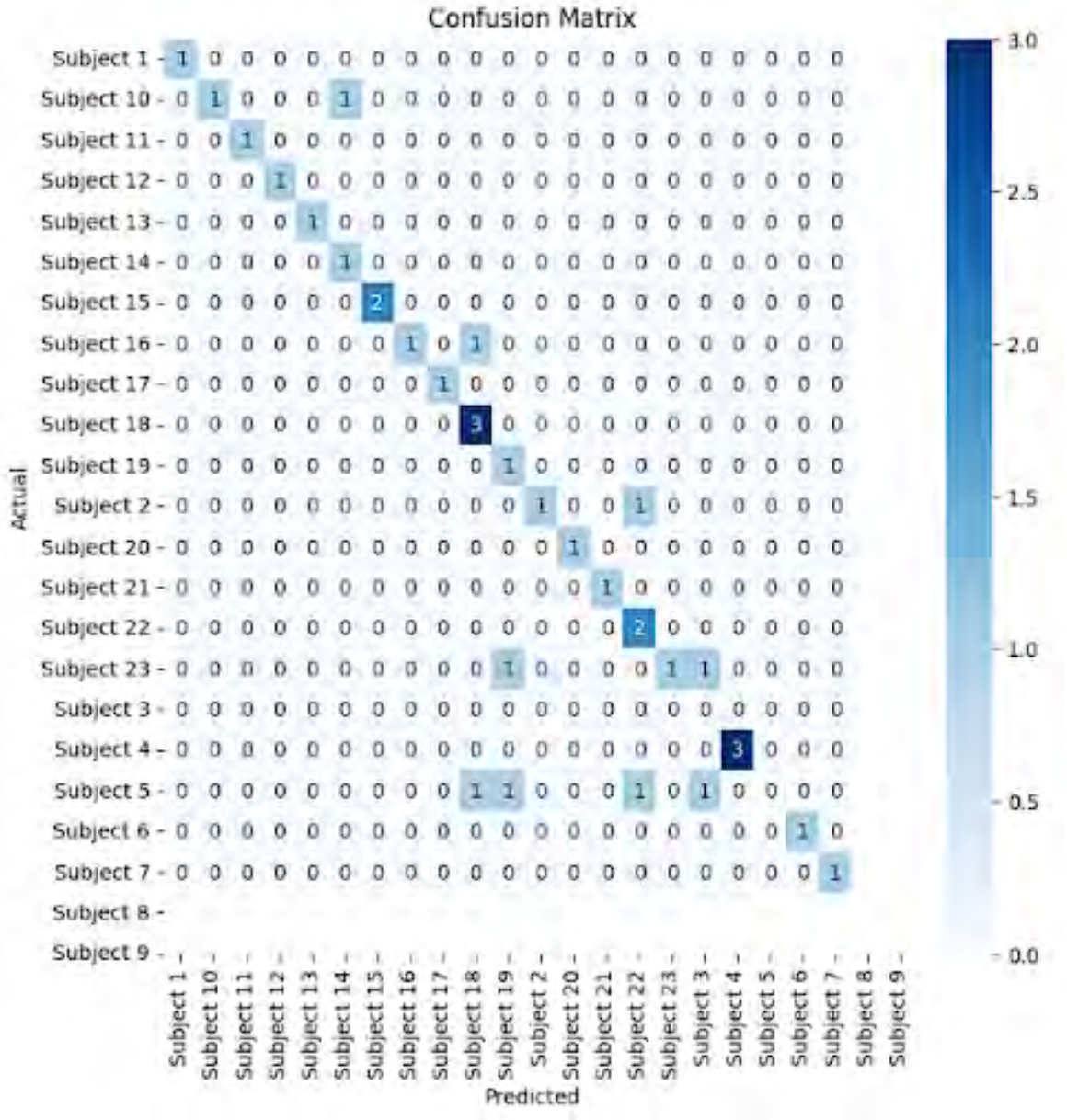



Figure 4.5: Confusion Matrix

There are many classes that have correct predictions, as indicated by the diagonal elements, which suggests that the model is able to correctly identify a number of instances. It can be deduced from the confusion matrix that the VGG16 model performs satisfactorily in terms of correctly classifying the majority of instances.

### ResNet50

ResNet50 is a frequently used model in both academic research and practical applications. The two main advantages of ResNet50 are that it has demonstrated exceptional accuracy in image classification tasks and it is widely employed as a benchmark in numerous academic papers, rendering it a highly valuable point of reference.[43] The two substantial cons are that ResNet50 is a highly accurate model, but its complexity can make it computationally expensive for certain applications, and similar to other deep learning architectures, ResNet50 necessitates a substantial quantity of labeled data to be used for training.[18] Here, in Figure 4.11, there is the ROC Curve.

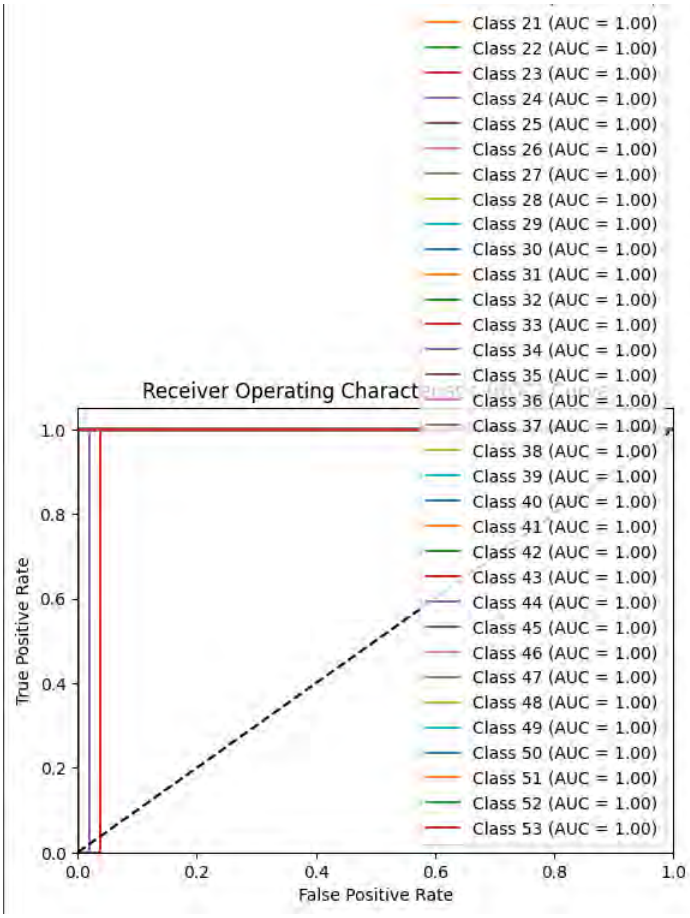



Figure 4.6: ROC Curve

Here, the majority of the subjects have ROC curves that have Area Under the Curve (AUC) values that are very close to or equal to 1.00. This indicates that the model performs exceptionally well in distinguishing between positive and negative classes for these subjects.

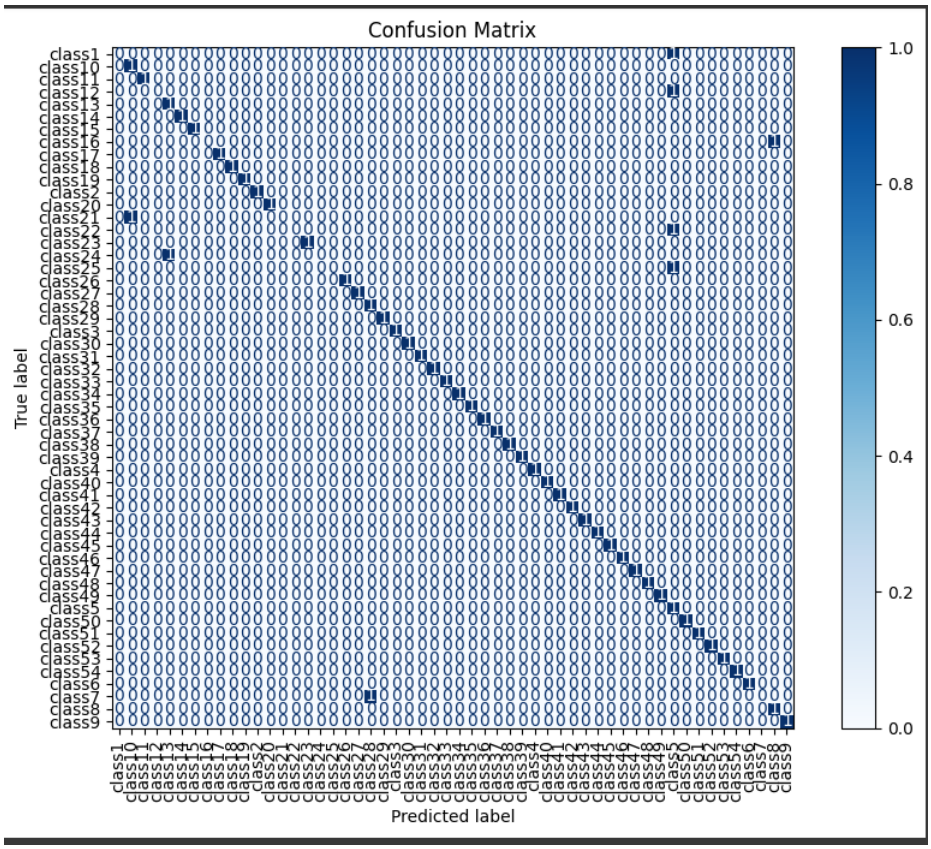



Figure 4.7: Confusion Matrix

Based on the fact that the diagonal dominates the matrix, it appears that the model is giving satisfactory results for the majority of the classes. Based on the confusion matrix, it can be seen that the ResNet50 model is performing reasonably well, as the majority of classes are being predicted accurately.

### MobileFaceNet

The lightweight architecture and high accuracy of MobileFaceNet make it an optimal model for facial identification tasks with ESP32-CAM. Integrating MobileFaceNet with ESP32-CAM enables us to augment the camera module's functionalities to carry out on-device face recognition in real-time. The integration of ESP32-CAM allows for accurate identification of individuals through facial recognition, making it well-suited for applications like as access control, surveillance, and personalized user interactions. By utilizing the efficiency of MobileFaceNet and the hardware capabilities of ESP32-CAM, the integrated solution provides a dependable and effective approach for facial identification in embedded systems. This guarantees secure and intelligent processing of visual input without relying on external cloud services.

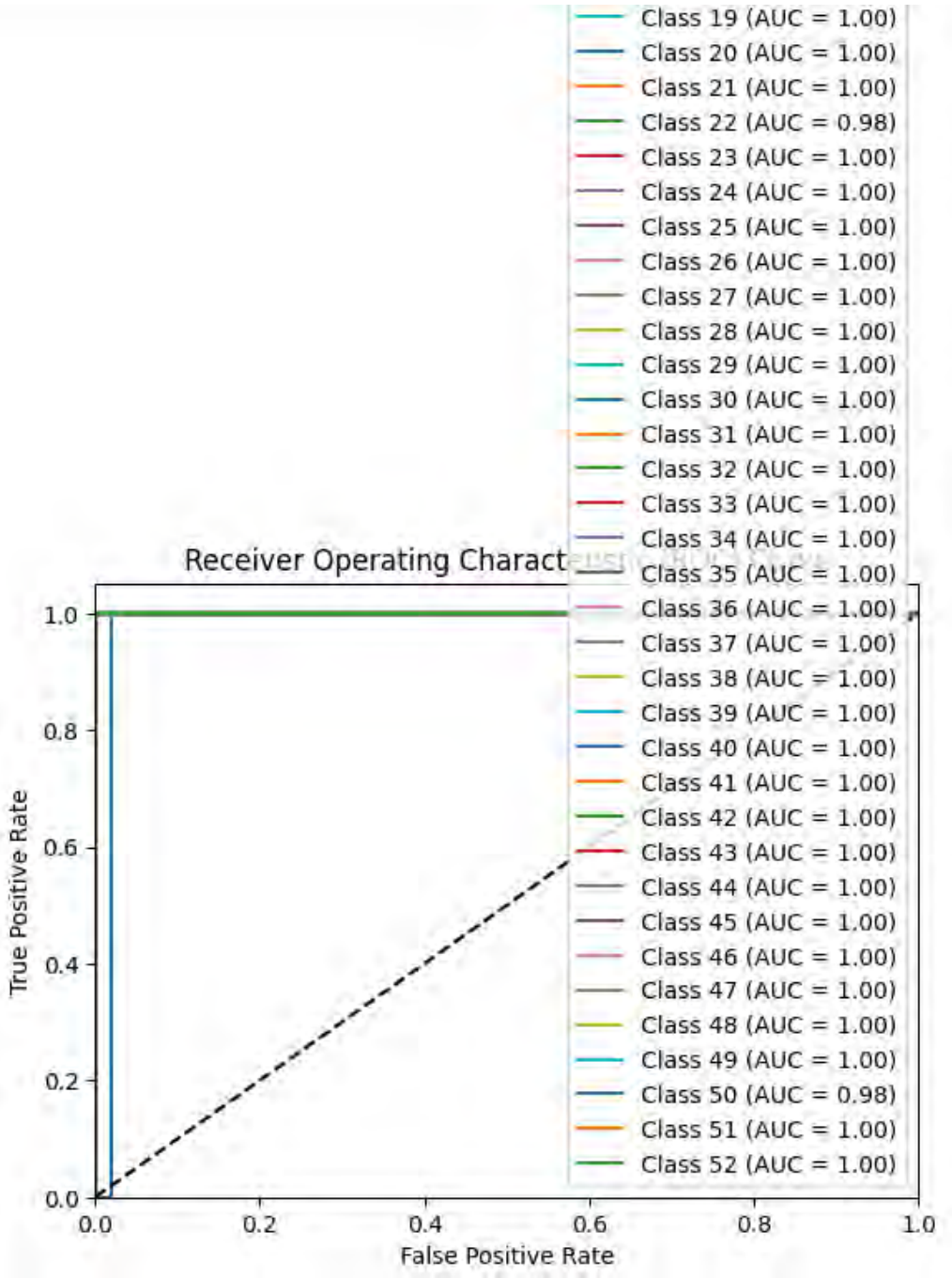



Figure 4.8: Roc Curve

In this case, most classes have an AUC of 1.00, suggesting that the model is performing exceptionally well, correctly identifying positive cases and avoiding false positives. This implies that the underlying MobileFaceNet model is highly effective for this specific task and dataset.

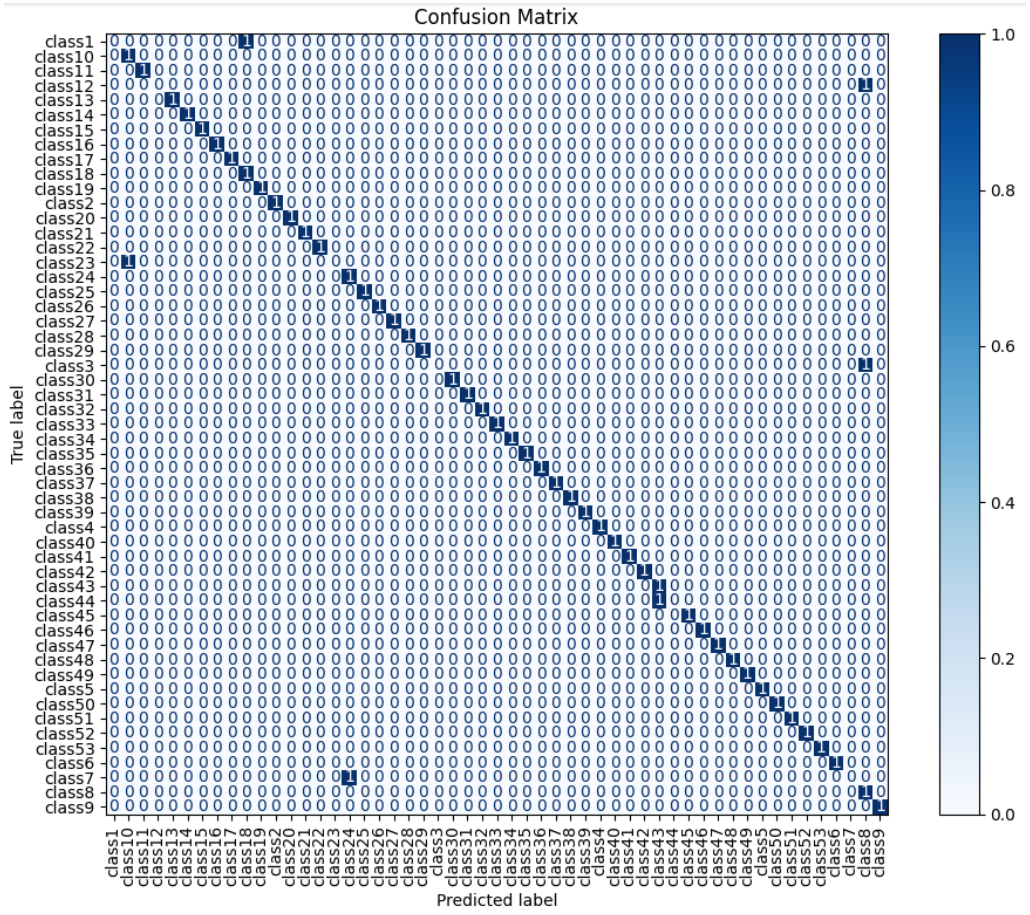



Figure 4.9: Confusion Matrix

The MobileFaceNet model has a high overall accuracy, with most faces correctly classified. The confusion matrix shows that the model is performing well, with only a few misclassifications. The off-diagonal elements of the confusion matrix represent the number of misclassified faces, while the diagonal elements represent the number of correctly classified faces. The model can be improved further by adding more training data to help the model learn to better identify the nuances of each class.

### MobileNet-V2

MobileNetV2 demonstrates its effectiveness and precision in object detection tasks by utilizing the COCO dataset. MobileNetV2 SSDLite is a modified version of ordinary SSD that uses separable convolutions instead of regular convolutions for prediction. It surpasses YOLOv2 in performance on the COCO dataset, exhibiting 20 times greater efficiency and 10 times less size, all while keeping a high level of accuracy. The model is trained and assessed using the Open Source TensorFlow Object Detection API, with an input resolution of 320x320. The performance evaluation of the COCO dataset object identification task demonstrates that MobileNetV2 + SSDLite achieves comparable accuracy while utilizing much less parameters and exhibiting smaller computing complexity in comparison to other real-time detectors.

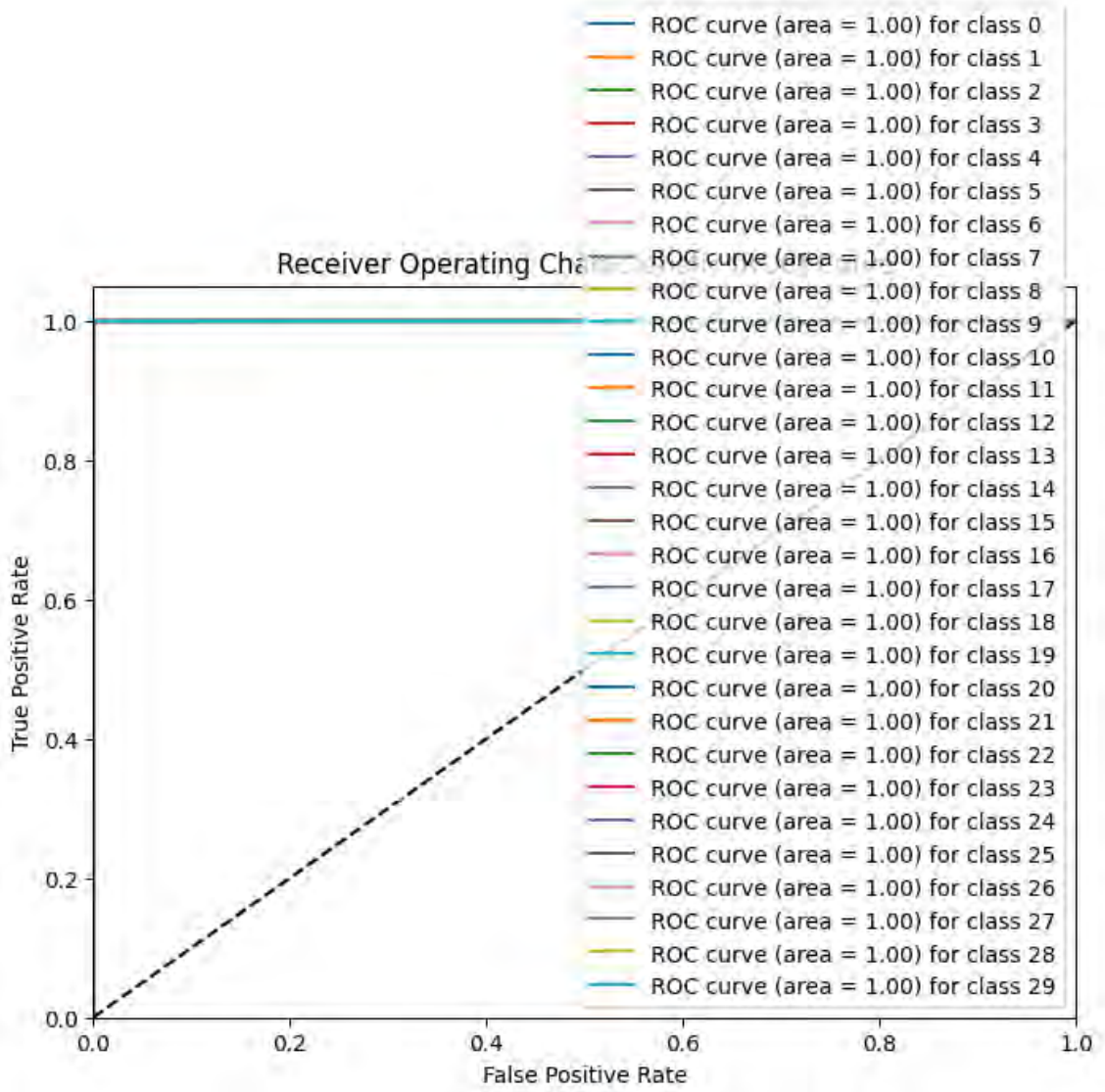



Figure 4.10: Roc Curve

Here the model correctly identifies all positive cases (TPR = 1) without misclassifying any negative cases (FPR = 0). In other words, the model has no false positives or false negatives, which is the best possible outcome for a classification model.

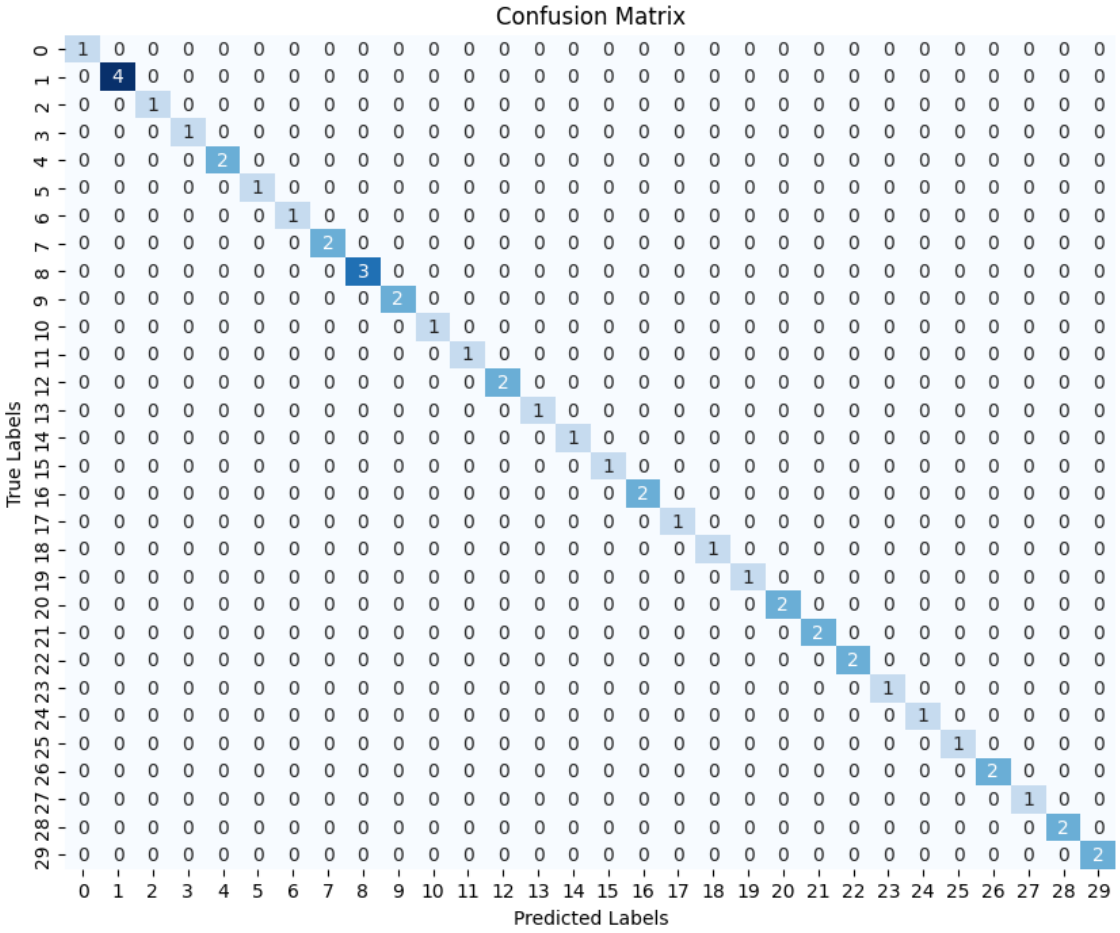



Figure 4.11: Confusion Matrix

The confusion matrix shows that the MobileNetV2 model is very good at classifying images into the correct categories. The diagonal of the matrix is very high, which indicates that the model is correctly classifying most of the images. There are a few off-diagonal values, which indicates that the model is sometimes misclassifying images. However, the overall accuracy of the model is very high.

## 4.7 Model Comparison

Here, we have the complete comparative analysis table.

| Models | Dataset Type | Accuracy | Precision | F1 Score |
|---|---|---|---|---|
| SqueezeNet | Initial Dataset | 82% | 85% | 84% |
| SqueezeNet | Augmented Dataset | 84.5% | 85.5% | 84.5% |
| ResNet50 | Initial Dataset | 85% | 87% | 83% |
| ResNet50 | Augmented Dataset | 96% | 91% | 94% |
| VGG16 | Initial Dataset | 82% | 82% | 81% |
| VGG16 | Augmented Dataset | 89% | 92% | 91% |
| MobileFaceNet | Initial Dataset | 83% | 81% | 82% |
| MobileFaceNet | Augmented Dataset | 90% | 93% | 91% |
| MobileNetV2 | Initial Dataset | 86% | 86% | 85% |
| MobileNetV2 | Augmented Dataset | 96% | 97% | 94% |

Table 4.1: Complete Comparative Analysis Table

The comprehensive analysis of various convolutional neural network (CNN) architectures for human identification in dynamic motion, specifically SqueezeNet, ResNet50, VGG16, MobileNetV2, and MobileFaceNet, reveals significant insights into their performance on both initial and augmented datasets of human faces. The impact of data augmentation is evident across all models, with substantial improvements in accuracy, precision, and F1 scores. ResNet50 and MobileNetV2 consistently achieve superior performance metrics, with accuracy improvements from 85% and 86% to 96%, respectively, demonstrating their robust capacity to generalize with enriched data. Notably, MobileNetV2 achieved a precision of 97% and an F1 score of 94% on the augmented dataset, highlighting its effectiveness in avoiding false positives and accurately identifying faces. While all models benefited from data augmentation, VGG16 and MobileFaceNet also exhibited significant enhancements, though the latter started from a comparatively lower performance baseline. This underscores the critical role of data augmentation in bolstering model generalization and effectiveness. The analysis indicates that deploying ResNet50 and MobileNetV2 on resource-constrained devices like the ESP32-CAM could yield highly effective and efficient human identification systems suitable for dynamic, real-world environments. These findings highlight the nuanced interplay between model architecture, dataset characteristics, and data augmentation techniques in shaping model performance and efficacy in real-world classification scenarios.

# Chapter 5

# Deployment

During the deployment phase of this study, the Arduino IDE served as the primary Integrated Development Environment (IDE) for interfacing with the ESP32-CAM microcontroller, allowing the deployment of a wide range of deep learning models. The models investigated included ResNet50, SqueezeNet, VGG16, FaceNet, MobileNetV2, and MobileFaceNet. Rigorous testing revealed compatibility difficulties and hardware restrictions inherent in the ESP32 platform. Despite these hurdles, MobileNetV2 and MobileFaceNet were successfully deployed, thanks to rigorous integration work. Notably, the streamlined architecture and reduced computational complexity of MobileNetV2 and MobileFaceNet were well-suited to the ESP32-CAM's resource-constrained environment, allowing them to run on hardware after incremental refinements. This deployment step highlighted the vital interplay between model design, hardware constraints, and integration approaches, providing insight into the practical considerations required for executing deep learning applications on edge devices such as the ESP32-CAM.


Figure 5.1: ESP32-CAM and SD card

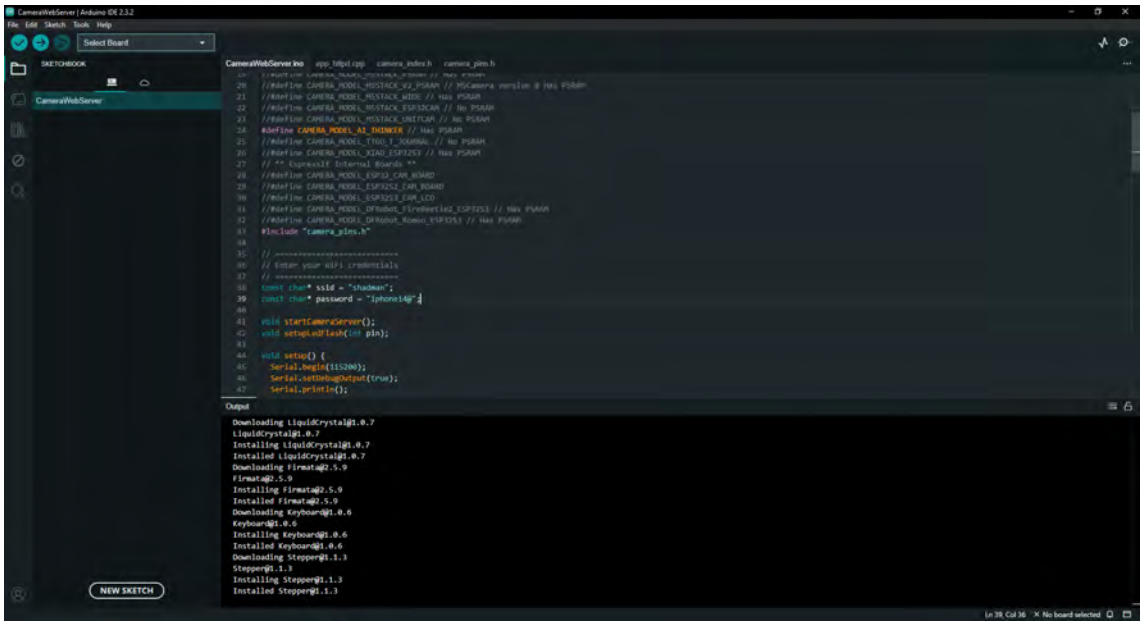



Figure 5.2: Arduino IDE

Initially, the dataset and trained models were kept on the ESP32-CAM's SD card to ensure efficient use of onboard storage. The models were invoked using the Arduino IDE, allowing for seamless interaction with the ESP32-CAM's peripherals. This integration enables the setup of a camera web server, allowing for real-time video streaming. When the ESP32-CAM was activated, the deployed model was executed in the background, which started the video streaming process. Subsequently, by continuously watching the video stream, the integrated model exhibited its capacity to recognize humans within the frame. The use of both MobileFaceNet and MobileNetV2 contributed to the successful completion of real-time identification tasks, demonstrating the deployed models' versatility and effectiveness. This comprehensive deployment strategy emphasizes the practical application of deep learning models in edge computing environments, particularly their capacity to provide intelligent and autonomous features in embedded systems.

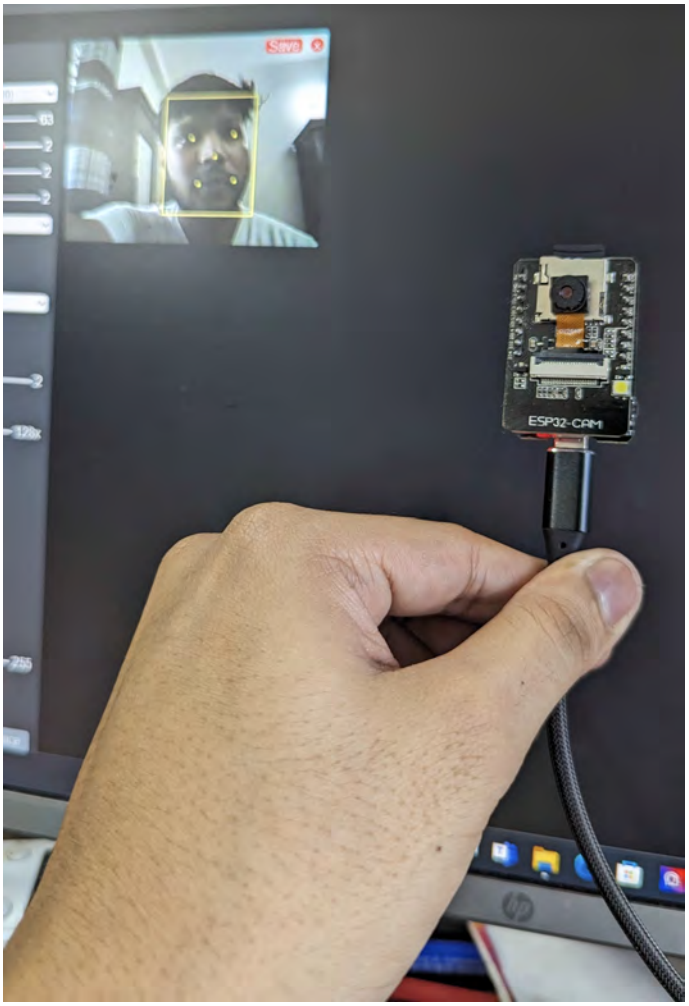



Figure 5.3: ESP32-CAM powers on

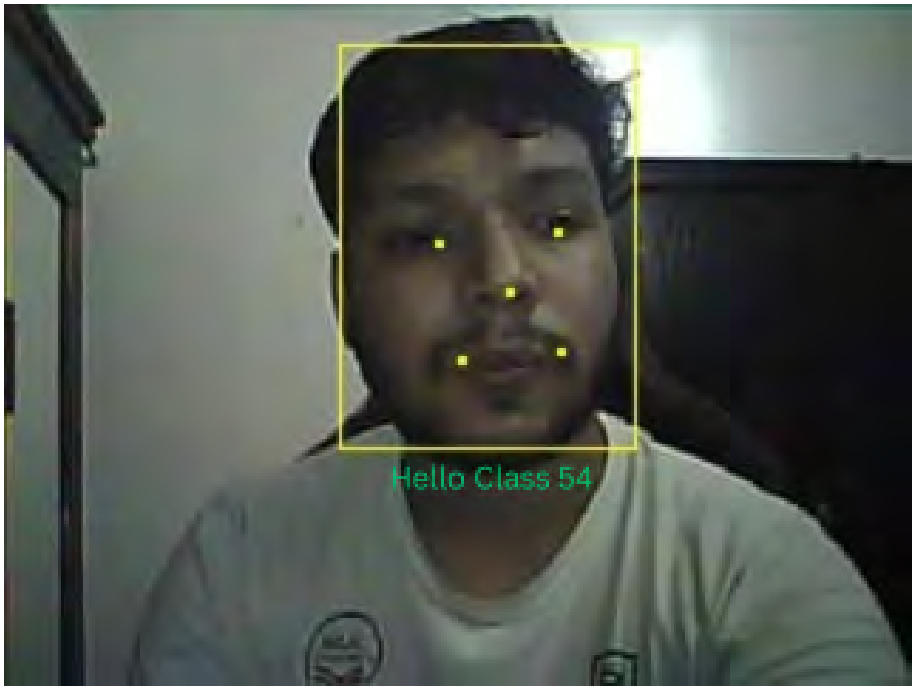



Figure 5.4: ESP32-CAM identifies face

# Chapter 6

# Future Work and Conclusion

## 6.1 Limitations

In our research, we used ESP32-CAM to do the identification process. It has a 2 Mega Pixel camera module attached to it. Moreover, this low cost device allows us to run machine learning models in it. Though it fulfills our main goal of providing a cheap and efficient solution for human identification, it has some limitations. Here are some of the shortcomings of this approach:

**Limited Processing Power:**
The ESP32-CAM has relatively limited computational power compared to powerful microcontrollers and microprocessors like Arduino, Raspberry Pi, Jetson Nano, etc. So, performing advanced image processing is a challenging task for this device. Moreover, we need to use lightweight machine learning algorithms as the advanced deep learning models can not be fully implemented on the ESP32-CAM.

**Memory Constraints:**
The ESP32-CAM has limited RAM, 520 KB SRAM, and 4MB PSRAM. Therefore, it is not feasible to store a large number of high-quality images or multiple frames for analysis.

**Image Quality:**
As said before, the camera module has a 2MP OV2640 sensor, which has limitations in terms of image quality. The maximum resolution is 1600 x 1200 pixels. For our convenience, we captured images at 360 x 160 pixels so that we could store more images. This method compromises the image quality and generates noisy images. Another major drawback is the lighting condition. If the lighting is very low, the model can not perform efficiently.

**Thermal Issue:**
As the device's camera will be continuously turned on, it will require a continuous power supply. Though it runs on 5V, prolonged usage can lead to overheating which can affect the performance of the device. It can also draw effects on the life span of the device.

**Security Concerns:**

The data stored in the ESP32 CAM's memory chip is very confidential, as it contains images of people. However, our device does not need to be connected to a WiFi system, as the camera will run in the backend, and the models and datasets are stored permanently in the device. But if we want to do something extra like sending signals with WiFi or Bluetooth connectivity, it can raise security concerns, as getting into the WiFi network and hampering Bluetooth connectivity is quite easy. Moreover, the device can be harmed by firmware attacks, which can compromise the integrity and confidentiality of the identification system.

In short, we can conclude that while the ESP32 -CAM offers an affordable and accessible solution for implementing tiny machine learning models, it also has some limitations. But if we can work with low-quality images or maintain a proper lighting environment, the shortcomings can be overcome.

## 6.2 Conclusion

When it comes to the study of human identity, the convergence of dynamic motion and affordable embedded technologies presents a compelling path for academic investigation. When it comes to the availability of cost-effective embedded systems for human identification, there is currently a significant disparity in the market. The existence of this gap provides academics with significant opportunities to conduct research and develop new ideas within this field.

The application of this research in the field of security and surveillance is among the most significant applications of this topic. When it comes to public areas, airports, and other high-security areas, having the capability to accurately identify individuals who are moving around can be beneficial to security measures. Real-time monitoring and identification of individuals can be accomplished through the deployment of sophisticated embedded systems, which help to prevent unauthorized access and ensure the safety of the general public. On the other hand, dynamic human identification can be utilized in the field of healthcare for the purpose of monitoring patients, particularly in circumstances where continuous observation is required. Within the realm of elder care, for instance, embedded systems have the capability to monitor the movements of patients in order to guarantee their safety and provide timely alerts in the event of falls or other accidents. Furthermore, the potential applications of such studies can extend beyond the realms of security and healthcare to encompass a wide range of fields, such as sports analytics, where motion tracking and identification can be utilized to enhance performance analysis, and personalized user experiences in consumer electronics.

The intersection of dynamic motion and cost-effective embedded technologies within the realm of human identification presents a promising area for academic investigation of the subject matter. There is a wide range of potential applications for this research that will continue to expand as technology advances. These applications will include a variety of fields, including healthcare, surveillance, and security, among others. There is a significant opportunity for academics working in this field to address the existing disparity in the solutions that are available and to make significant contributions to the field of human identification. Researchers have the

ability to pave the way for advancements that improve the accuracy, reliability, and applicability of human identification in dynamic environments by leveraging the capabilities of embedded systems and developing algorithms that are resilient.

## 6.3 Future Work

The result of our research reflects that with this approach, we can bring a massive change in the human identification system. This approach shows us a cheaper way to increase the security system of our homes, offices, vehicles, etc. With the help of our research, we believe that more IoT-based devices will be easily accessible to mass people. If we can work on certain problems that we have faced through our process, we believe that it can become a great help to the country.

Firstly, we need to solve the low-light environment problem, as the ESP32-CAM can not work properly in a low-light environment. To solve this problem, we need hardware upgrades, such as integrating a higher-quality camera module in the ESP32-CAM or, alternatively, developing custom sensors that would be optimized for performing human identification tasks. Besides this, we can try different tools to turn a low-quality image into a high-quality image. So that when the ESP32-CAM is operating in low light, it can enhance the video quality on its own and identify the person accordingly. Implementing real-time image enhancement algorithms can improve the quality of images. So, researchers should focus on how to implement a light version of these heavy-weight algorithms, which will be suitable for ESP32-CAM.

Secondly, we need to enhance the processing power and efficiency. Hybrid models can be implemented where the identification process is faster while requiring low computational resources. Optimization can be done to reduce the computational requirements of machine learning models. Model quantization, pruning, and knowledge distillation can help in optimization. Another way to achieve this goal is to perform model compression. Compressed versions of existing models should be introduced while maintaining high accuracy.

As a final step, the power consumption of the ESP32-CAM should be decreased in order to make it more energy efficient. There is an absolute requirement to investigate and develop new hardware modifications in order to incorporate components that save energy. These will lessen the amount of heat that is produced, which will allow the device to function more smoothly for longer periods of time. We need to work on implementing secure over-the-air mechanisms that will ensure the firmware of the device can be updated with the most recent security patches in order to protect the firmware.

# Bibliography


[1] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," *Computer vision and image understanding*, vol. 73, no. 3, pp. 428–440, 1999.

[2] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 12, pp. 1505–1518, 2003.

[3] J. Wang, M. She, S. Nahavandi, and A. Kouzani, "A review of vision-based gait recognition methods for human identification," in *2010 international conference on digital image computing: techniques and applications*, IEEE, 2010, pp. 320–327.

[4] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[5] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[6] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[7] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices," in *Biometric Recognition: 13th Chinese Conference, CCBR 2018, Urumqi, China, August 11-12, 2018, Proceedings 13*, Springer, 2018, pp. 428–438.

[8] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[10] X. Li, F. Wang, Q. Hu, and C. Leng, "Airface: Lightweight and efficient model for face recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[11] S. Asghari-Esfeden, M. Sznaier, and O. Camps, "Dynamic motion representation for human action recognition," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 557–566.

[12] M. Giordano, P. Mayer, and M. Magno, "A battery-free long-range wireless smart camera for face detection," in *Proceedings of the 8th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, 2020, pp. 29–35.

[13] M. M. Islam, N. Tasnim, and J.-H. Baek, "Human gender classification using transfer learning via pareto frontier cnn networks," *Inventions*, vol. 5, no. 2, p. 16, 2020.

[14] A. J. Paul, P. Mohan, and S. Sehgal, "Rethinking generalization in american sign language prediction for edge devices with extremely low memory footprint," in *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, IEEE, 2020, pp. 147–152.

[15] P. M. S. Ribeiro, A. C. Matos, P. H. Santos, and J. S. Cardoso, "Machine learning improvements to human motion tracking with imus," *Sensors*, vol. 20, no. 21, p. 6383, 2020.

[16] M. Akay, Y. Du, C. L. Sershen, *et al.*, "Deep learning classification of systemic sclerosis skin using the mobilenetv2 model," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 2, pp. 104–110, 2021.

[17] H. Benmeziane, K. E. Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, and N. Wang, "A comprehensive survey on hardware-aware neural architecture search," *arXiv preprint arXiv:2101.09336*, 2021.

[18] S. Mascarenhas and M. Agarwal, "A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification," in *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, IEEE, vol. 1, 2021, pp. 96–99.

[19] P. Mohan, A. J. Paul, and A. Chirania, "A tiny cnn architecture for medical face mask detection for resource-constrained endpoints," in *Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2021*, Springer, 2021, pp. 657–670.

[20] M. de Prado, M. Rusci, A. Capotondi, R. Donze, L. Benini, and N. Pazos, "Robustifying the deployment of tinyml models for autonomous mini-vehicles," *Sensors*, vol. 21, no. 4, p. 1339, 2021.

[21] J. Tao, Y. Gu, J. Sun, Y. Bie, and H. Wang, "Research on vgg16 convolutional neural network feature classification algorithm based on transfer learning," in *2021 2nd China international SAR symposium (CISS)*, IEEE, 2021, pp. 1–3.

[22] M. Wu, B. Jiang, D. Luo, *et al.*, "Learning comprehensive motion representation for action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 2934–2942.

[23] X. Xu, M. Du, H. Guo, J. Chang, and X. Zhao, "Lightweight facenet based on mobilenet," 2021.

[24] W. Budiharto, E. Irwansyah, J. S. Suroso, and A. A. S. Gunawan, "Low-costvision-based face recognition using esp32-cam for tracked robot," *ICIC Express Letters Part B: Applications*, vol. 13, no. 3, pp. 321–327, 2022.

[25] S. Gupta, S. Jain, B. Roy, and A. Deb, "A tinyml approach to human activity recognition," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 2273, 2022, p. 012 025.

[26] S. Islam, J. Deng, S. Zhou, C. Pan, C. Ding, and M. Xie, "Enabling fast deep learning on tiny energy-harvesting iot devices," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2022, pp. 921–926.

[27] B. Sudharsan, S. Salerno, and R. Ranjan, "Tinyml-cam: 80 fps image recognition in 1 kb ram," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 862–864.

[28] Y. Wahyu *et al.*, "A performance evaluation of esp32 camera face recognition for various projects," *Internet of Things and Artificial Intelligence Journal*, vol. 2, no. 1, pp. 10–21, 2022.

[29] L. Yuan, J. Andrews, H. Mu, *et al.*, "Interpretable passive multi-modal sensor fusion for human identification and activity recognition," *Sensors*, vol. 22, no. 15, p. 5787, 2022.

[30] X. Zhang, Z. Xu, and H. Liao, "Human motion tracking and 3d motion track detection technology based on visual information features and machine learning," *Neural Computing and Applications*, vol. 34, no. 15, pp. 12 439–12 451, 2022.

[31] Y. Zhou, H. Zhao, Y. Huang, T. Riedel, M. Hefenbrock, and M. Beigl, "Tinyhar: A lightweight deep learning model designed for human activity recognition," in *Proceedings of the 2022 ACM International Symposium on Wearable Computers*, 2022, pp. 89–93.

[32] N. N. Alajlan and D. M. Ibrahim, "Ddd tinyml: A tinyml-based driver drowsiness detection model using deep learning," *Sensors*, vol. 23, no. 12, p. 5696, 2023.

[33] F. J. Gruber, "Sensor based human activity recognition using edge computing and tinyml in combination with convolutional neural networks," Ph.D. dissertation, University of Applied Sciences, 2023.

[34] R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. López, "Tinyml: Tools, applications, challenges, and future research directions," *arXiv preprint arXiv:2303.13569*, 2023.

[35] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, and S. Han, "Tiny machine learning: Progress and futures [feature]," *IEEE Circuits and Systems Magazine*, vol. 23, no. 3, pp. 8–34, 2023.

[36] J.-J. Lin, C.-K. Hsu, W.-L. Hsu, T.-C. Tsao, F.-C. Wang, and J.-Y. Yen, "Machine learning for human motion intention detection," *Sensors*, vol. 23, no. 16, p. 7203, 2023.

[37] S. Raghuvanshi and S. Dhariwal, "The vgg16 method is a powerful tool for detecting brain tumors using deep learning techniques," *Engineering Proceedings*, vol. 59, no. 1, p. 46, 2023.

[38] A. Sabovic, M. Aernouts, D. Subotic, J. Fontaine, E. De Poorter, and J. Famaey, "Towards energy-aware tinyml on battery-less iot devices," *Internet of Things*, vol. 22, p. 100 736, 2023.

[39] V. Viswanatha, A. Ramachandra, P. T. Hegde, V. Hegde, and V. Sabhahit, "Tinyml-based human and animal movement detection in agriculture fields in india," in *International Conference on Emerging Research in Computing, Information, Communication and Applications*, Springer, 2023, pp. 49–65.

[40] Y. Xiong, S. A. Moqurrab, and A. Ahmad, "Enhancing human motion prediction through joint-based analysis and avi video conversion," 2023.

[41] GeeksforGeeks, *Vgg-16 — cnn model*, https://www.geeksforgeeks.org/vgg-16-cnn-model/, Accessed: 2024-06-05, 2024.

[42] D. Robert, H. Raguet, and L. Landrieu, *Scalable 3d panoptic segmentation with superpoint graph clustering*, 2024. arXiv: 2401.06704 [cs.CV].

[43] D. Categorization and B. Koonce, "Convolutional neural networks with swift for tensorflow,"