Name: XUENING WAN
Matriculation number: S2195023
MEng Honours Project Phase 1 Report
< Design of Facial and Action Recognition Algorithm suitable for Remote Health Monitoring to the Elder People>
6 August 2024

Mission Statement

Project Title: ***Design of Facial and Action Recognition Algorithm suitable for Remote Health Monitoring to the Elder People***

Student: ***Xuening Wan***

Supervisor: ***Dr. Gary Wells***

Project Definition

Nowadays NHS is extremely understaffed, and it is estimated to be much more severe in the future. Thus, some procedures or methods can be improved to tackle on this. Remote monitoring is a sensible method that the NHS has already taken into use. While reducing unnecessary travel for appointments, it still supports the provision of immediate health situations for patients without relying on the availability of the practitioner or patient at the same time. The aim of the project is to design an algorithm that can be used for remote monitoring. It will be able to detect unusual behaviors that patients have (e.g. painful face, abnormal posture, or gait) and notify caregivers in the first moment (e.g. a doctor or nurse from the NHS). In order to accomplish this, different recognition algorithms will be needed (posture recognition, face recognition, facial expression recognition and gait recognition). Each algorithm will be developed individually and then combined into the final algorithm. The algorithm will then be tested at different camera angles, so its reliability at different scenarios can be revealed.

Main Tasks

- *To design an algorithm for remote monitoring.*
- *Recognize the patients.*
- *Distinguish abnormal postures or gaits from the patient's normal ones.*
- *Recognize abnormal facial expressions.*

- *Notify the professionals when anything goes wrong.*

Scope for Extension

- *Understand the patient's daily routine.*
- *Reliability of results at different camera angles.*
- *Recognize any dangers from surroundings*

Background Knowledge

- *Python*
- *Posture Recognition Algorithms*
- *Facial Expression Recognition Algorithms*
- *Gait Recognition Algorithms*

Resources

- *What do you need to complete the project?*
- *Camera*
- *A Computer for writing the script and executing the program*
- *Online Resources*

Location

- *No specific access permission is required.*

References

1. *X. Jiang, Z. Hu, S. Wang, and Y. Zhang, "A survey on Artificial Intelligence in posture recognition," Computer Modeling in Engineering & Sciences, vol. 137, no. 1, pp. 35–82, 2023. doi:10.32604/cmes.2023.027676.*

2. *"Face recognition using artificial intelligence," GeeksforGeeks, https://www.geeksforgeeks.org/face-recognition-using-artificial-intelligence/ (accessed Feb. 7, 2024).*

3. *X. Guo, Y. Zhang, S. Lu, and Z. Lu, "Facial expression recognition: A Review," Multimedia Tools and Applications, Aug. 2023. doi:10.1007/s11042-023-15982-x*

4. *A. Parashar, A. Parashar, A. F. Abate, R. S. Shekhawat, and I. Rida, "Real-time gait biometrics for surveillance applications: A Review," Image and Vision Computing, vol. 138, p. 104784, Oct. 2023. doi:10.1016/j.imavis.2023.104784*

5. *NHS choices, https://transform.england.nhs.uk/blogs/role-remote-monitoring-future-nhs/ (accessed Feb. 7, 2024).*

6. *"Remote Health Monitoring," Learn, https://learn.nes.nhs.scot/49680 (accessed Feb. 14, 2024).*

7. *"NHS staff shortages in England could exceed 570,000 by 2036, leaked document warns," The Guardian, https://www.theguardian.com/society/2023/mar/26/nhs-england-staff-shortages-could-exceed-570000-by-2036-study-finds (accessed Feb. 14, 2024).*

The supervisor and student are satisfied that this project is suitable for performance and assessment in accordance with the guidelines of the course documentation.

Signed

Student: .....

Supervisor: .....

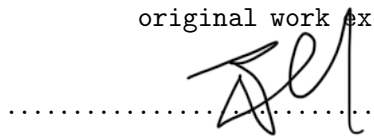
Date: 06.08.2024

Abstract

Given the trend of higher life-expectancy and an aging population in UK, remote health monitoring can be ideal to reduce the workloads of healthcare providers while ensuring continuous monitoring of elderly individuals. This thesis investigates the design and implementation of a visual recognition algorithm designed for robotic entities, such as robotic dogs, to monitor the health of elderly individuals. The algorithm integrates action recognition and facial recognition to evaluate and interpret the physical (e.g. falling or walking) and emotional states (e.g. sad, happy) of individuals. Through further enhancement and modification in phase two, the algorithm shall provide comprehensive assessments of an individual's status, which aims to contribute to remote health monitoring systems by timely notification of abnormal behaviors to alleviate the burden on healthcare providers.

Declaration of Originality

I declare that this thesis is my
original work except where stated.



.....

Statement of Achievement

This project represents a significant achievement in the integration of advanced facial recognition and action recognition technologies. The key accomplishment of this report is the successful development and implementation of an algorithm that seamlessly combines facial recognition (including facial verification and facial expression recognition through DeepFace), with action recognition (using yolo_slowfast).

Throughout the project, I independently developed and tested the algorithm, incorporating valuable suggestions from my supervisor. The core achievements include:

- **Facial Recognition Integration**: Successfully implemented DeepFace for both facial verification and facial expression recognition, ensuring accurate and reliable identification and analysis of facial features.
- **Action Recognition Integration**: Utilized the YOLO model for object detection and the SlowFast model for action recognition, creating a robust system capable of detecting and understanding actions within video sequences.
- **Algorithm Development**: Designed and developed a comprehensive algorithm that integrates facial and action recognition capabilities, enabling the system to perform complex tasks such as identifying individuals and recognizing their actions and expressions in real-time.

Contents

List of Figures

List of Tables

Glossary

AP@50 Average Precision at Intersection over Union of 50%.

CNN Convolutional Neural Network.

CSP-Darknet53 Cross Stage Partial Darknet53.

CSP-PAN Cross Stage Partial Path Aggregation Network.

GAN Generative Adversarial Network.

LSTM Long Short-Term Memory.

MAE Mean Absolute Error.

mAP mean Average Precision.

ONS Office for National Statistics.

R3D Recurrent 3D Convolutional Neural Network.

RLE Remaining Life Expectancy.

RPM Remote Patient Monitoring.

SORT Simple Online Realtime Tracking.

SPPF Spatial Pyramid Pooling Fast.

UN United Nations.

WHO World Health Organization.

Chapter 1

Introduction and Background

1.1 Background

The National Health Service (NHS) in the UK is currently facing significant strain due to increased life expectancy and an aging population [1]. This demographic shift indicates healthcare providers would face escalating workloads and resource constraints. Remote health monitoring has emerged as a viable solution through its continuous health supervision [2]. This approach ensures immediate updates on patients' health statuses and allows timely interventions without the need for frequent in-person appointments or monitors [3].

1.2 Project Overview

This thesis explores the design and implementation of a visual recognition algorithm designed for robotic entities, such as robotic dogs, to assist the health monitoring of elderly individuals. The proposed algorithm integrates action recognition and facial recognition to comprehensively evaluate and interpret both the physical (e.g., falling, walking) and emotional (e.g., sad, happy) states of individuals. Each subsystem will be developed individually and subsequently integrated into a unified framework. By combining these components, the algorithm aims to provide thorough assessments of an individual's status, facilitating timely notifications of abnormal behaviors to caregivers and thereby reducing the burden on healthcare providers.

1.3 Algorithm Design

This section outlines the specific methodology employed in developing the visual recognition algorithm. The project leverages advanced tools such as DeepFace [4] for facial recognition and yolo_slowfast [5] for pose estimation to ensure comprehensive and reliable assessments of health status. DeepFace offers robust capabilities in recognizing facial features and emotions, while yolo_slowfast excels in object detection and pose estimation. The preliminary system design focuses on integrating these tools into a cohesive framework that assesses both physical and emotional states. Initial development emphasizes establishing the fundamental components, with plans to expand and refine these elements in subsequent phases..

1.4 Research Impact

The significance of this research lies in its potential to improve remote health monitoring systems, which offers a reliable and efficient solution to protect the well-being of older individuals. By providing healthcare professionals with timely alerts regarding abnormal behaviors, this technology can contribute to improved patient outcomes and a reduction in the workload of healthcare providers.

Chapter 2

Literature Review

2.1 Demographics and its potential impact

2.1.1 Ageing and Health

The **WHO** (World Health Organization) provides detailed explanations of ageing and corresponding impacts [6]. Aging causes a decrease in physical and mental performance with an increased risk of disease and death. From a biological perspective, it is a result of the impact of the accumulated damage on different molecular and cellular aspects over time. It is also often associated with life transitions such as retirement. Today, most people can live for more than 60 years. One in six people in the world will be 60 years or older by 2030, which is equivalent to 1.4 billion (1 billion in 2020). By 2050, this number will increase to 2.1 billion, and those aged 80 years or older are expected to reach 426 million.

A longer life can imply further opportunities and abundant time to pursue new activities. However, a variety of health conditions limit the possibilities. Elderly people can experience one or multiple conditions such as hearing loss, cataracts, and neck pain. Healthy aging is thus important, and the General Assembly of the United Nations (**UN**) has declared 2021-2030 the UN Decade of Healthy Ageing. It is a global partnership led by WHO to promote longer and healthier lives. It focuses on four areas:

1. changing the way we think, feel and act about age and ageism;
2. developing communities in ways that build the capabilities of older people;
3. providing integrated, person-centred care and primary health services for older people;
4. providing quality long-term care for older people in need.

2.1.2 UK's trends on population

The mid-2023 population estimates for England and Wales [7] from the Office for National Statistics (**ONS**) suggests that 58% of local authorities estimated to have a higher number of deaths than births, where only 26% had more deaths than births 10 years ago. This reflects the effects of decreasing fertility rates and an ageing population. In the year to mid-2023, the number of people aged 65 years or over has increased by 1.4%, and it has increased by about 110,000 (4.5%) for ages 75-79. The pyramid chart (Figure 2.1) from the article indicates there are significantly more woman who has aged 90 years or over. At the same time, the age composition varies with regions. The median age for the population in Wales at mid-2023 was 42.8, where it was only 40.4 in England. Another article [8] from ONS also suggests the ageing trend in Great Britain (Figure 2.2), and indicates a different perspective of ageing population (prospective population ageing). 65 is traditionally considered as the start of older age, as health measures have improved over time at every chronological age, the age 70 could be considered the new 65 in terms of health and life expectancy. By 2017, a 70-year-old had a remaining life expectancy (**RLE**) similar to a 65-year-old's in 1997. These factors should be considered when designing the elderly people's health aid plan. For example, if the number of free remote health monitoring devices is limited, people with shorter RLE should have first access towards the products, where regions that have more eligible people should be allocated more devices. To accommodate the higher number of female aged 90 or over than male, and address problems such as biological differences, it may be necessary to develop and manufacture specific versions of health monitoring devices tailored to their needs.

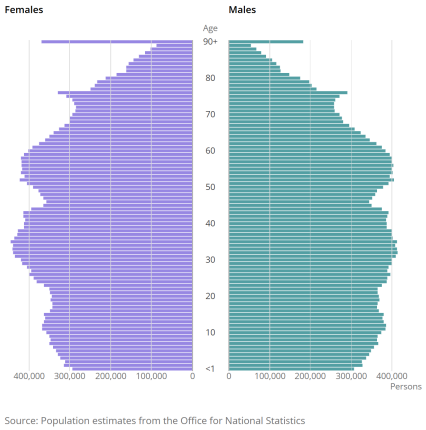


Figure 2.1: population pyramid in mid-2023

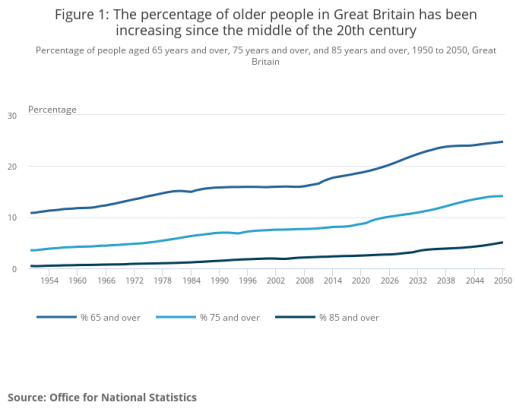


Figure 2.2: The percentage of older people in Great Britain has been increasing since the middle of the 20th century

2.2 Remote Health Monitoring

Remote health monitoring systems have gained traction due to their potential to provide real-time health data, reduce hospital visits, and enhance patient outcomes. The systematic review published by Tan et al. in npj Digital Medicine [9] investigates the impact of Remote Patient Monitoring (**RPM**) interventions on various outcomes, focusing on safety, adherence, quality of life, and cost-effectiveness. RPM technologies examined include communication tools, smartphone apps, and wearables. Key findings from the review indicate that RPM can improve patient safety, adherence to treatments, and mobility. Additionally, RPM interventions were found to reduce hospital readmissions and overall healthcare costs. The effectiveness of RPM was also acknowledged as non-pharmacological tool for monitoring both acute and chronic conditions of elder people. However, the mixed results on quality-of-life improvements suggest that while some patients experience increased empowerment and satisfaction, others may feel burdened by the constant monitoring. Economic evaluations indicate that while RPM can reduce overall healthcare costs, the initial setup and maintenance costs may be high. Implementation barriers such as technological literacy, data privacy concerns, and resistance from healthcare providers are also key considerations.

The integration of advanced technologies has been pivotal in the development of effective remote monitoring solutions. Schütz et al. [10] indicates a large portion of digital measures for health relates to mobile technologies that may require interactions such as smart watches. The effectiveness of them may decrease on elder people due to four reasons:

1. Older adults tend to be less attracted to novel technologies;
2. monitoring duration may become very long or even unlimited;
3. Many older adults tend to fear being seen as frail if they wear a device;
4. Not feasible for seniors with potential memory issues, wearing and maintaining devices may not be feasible.

A zero-interaction digital exhaust was then designed, which refers to the passive, continuous collection of data from non-contact sensors installed in an older adult's living environment. A set of 94 hypothesis-driven base measures were introduced, and a total of 1268 digital measures using aggregation and frequency analysis were further derived. The study demonstrates that these measures can effectively assess aging-related health outcomes like fall risk, frailty, and mild cognitive impairment using machine learning models, thus providing an alternative to wearable sensors and aiding in early intervention and precision medicine. These findings provide an idea when designing the remote monitoring system of this project. The entity chosen for monitoring

should make the patient willing to be within the monitoring area, where a cute robotic dog is a great example. The whole monitoring process should be as non-interactive as possible, unless the interaction is necessary and urgent(e.g. checking if the patient is just lying or unconscious). Continuous and timely monitoring should be available, and the patient should not need to set anything for the system.

2.3 Facial Recognition

Facial recognition technology has advanced significantly in recent years, driven by improvements in deep learning algorithms and the availability of large datasets. According to the review by Taskiran et al. [11], the history of Facial Recognition could be traced back to 1950s and 1960s, where research of automatic face recognition is considered happening in early 70s. Further iterations and innovations happened, and deep learning based methods achieved popularity after AlexNet won great success in the 2012 ImageNet competition. Deep learning based methods normally consist of three stages:

1. **Face Pre-processing**:
   (a) **One-to-Many Augmentation**: Generates images in various poses from a single image using data augmentation, 3D face models, or convolutional neural network (**CNN**) models to create 2D images in different poses.
   (b) **Many-to-One Normalization**: Aims to generate canonical views of faces from multiple angles using Stacked Auto encoders, CNNs, or **GAN**s.
   (c) **Illumination Robust Pre-processing**: Methods to remove shadows while retaining identity-related information have been proposed.
2. **Deep Feature Extraction**: Network Architectures such as typical CNN architectures like AlexNet, VGGNet, GoogleNet, ResNet, and SENet, as well as networks for multi-task learning. Loss Functions are used to improve feature discriminability, loss functions like Euclidean-distance-based loss, triplet loss, angular/cosine-margin-based loss, and variations of soft-max loss are used. One-Shot Face Recognition addresses the challenge of having only a single image of a subject using methods like intermediate deep attribute representations, regularization functions, and GANs for data synthesis.
3. **Face Matching**: Use of cosine distance or other metric learning methods to match faces after deep feature extraction.

2.3.1 DeepFace

DeepFace [4] has been chosen for this project. It is a comprehensive and lightweight tool for facial analysis that handles all four common stages of a modern face recognition pipeline: detection, alignment, representation, and verification. It integrates a series of advanced models to identify and verify faces as well as analyze facial attributes with a high degree of accuracy under various conditions. According to experiments conducted by the DeepFace team, many models successfully achieved or surpassed the level of human accuracy (97.53%) on the same dataset, with FaceNet-512d demonstrating the highest performance. The flexibility of DeepFace in wrapping various advanced models allows users to select the most appropriate model based on their specific needs.

Facial Recognition

According to Serengil et al. [12],the initial stages of a face recognition pipeline involve detection and alignment of faces, where OpenCV is used to detect frontal faces and eyes, but it does not provide face alignment functions. Face alignment is crucial as it significantly improves the accuracy of face recognition models, raising it from 98.87% to 99.63% according to Google researchers. This improvement is achieved by preprocessing, without modifying classification models. To implement alignment, trigonometric methods are used to calculate the rotation angle needed to align faces based on the positions of the eyes detected by OpenCV. By constructing a right-angled triangle using the eye positions, the cosine rule (Formula 2.1) is applied to find the necessary rotation angle, which is then used to rotate the base image, ensuring consistent face orientation for the recognition model.

$$\cos A = \frac{b^2 + c^2 - a^2}{2bc} \tag{2.1}$$

Face recognition models are essentially CNNs that represent face images as vectors. The CNNs are initially trained to classify identities, these CNNs use their early layers to extract raw facial feature vectors, enabling the verification of previously unseen faces. The represented vectors are then fed into the verification module, where similarities are assessed using metrics such as cosine similarity, Euclidean distance, or its L2 form. The Euclidean distance between two vectors is defined as the length of the line connecting them (Formula 2.2). Cosine similarity (Formula 2.3), derived from the Euclidean dot product formula, measures the angle between two vectors, with values ranging from -1 to +1. This can be converted to cosine distance (Formula 2.4), aligning the interpretation of distances in both metrics. Optimum threshold is determined through the feed of positive and negative instances to a decision tree algorithm. Overall, FaceNet exhibits the highest accuracy, and the Euclidean L2 offers the best robustness compare to other

distance metrics.

The framework handles all four stages seamlessly in the background. This framework constructs a face recognition model using TensorFlow and Keras and downloads pre-trained weights for these models due to their large file sizes. It performs face detection and alignment, converts image pairs into vectors using the selected face recognition model, computes distances between these vectors based on the chosen metric, and finally determines whether the pair represents the same person using an optimal threshold. All these processes are managed automatically. In addition, it provides an out-of-the-box function for finding a face in a large scale data set. Representations are pre-extracted and stored due to costs. When calling the find function, only the embedding of the target image, and distances for all alternatives in the database needs to be found, which significantly speeds up the process.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.2}$$

$$\cos\theta = \frac{\sum_{i=0}^{n} x_i y_i}{\sqrt{\sum_{i=0}^{n} x_i^2}\sqrt{\sum_{i=0}^{n} y_i^2}} \tag{2.3}$$

$$D_c(x, y) = 1 - \cos\theta \tag{2.4}$$

Serengil et al. further evaluated and optimized facial recognition pipelines by systematically assessing the performance of various configurations in 2024 [13]. This study utilized the LightFace framework to perform an exhaustive series of experiments, evaluating the co-usability and performance impact of different modules within the facial recognition pipeline, encompassing nine state-of-the-art facial recognition models, six cutting-edge face detectors, three distance metrics, and two alignment modes. The research underscored the significant influence of detection and alignment stages on overall accuracy and robustness, with detection enhancing performance by up to 40% and alignment contributing up to a 17% improvement. Using metrics such as the Area Under the Curve (AUC) and Receiver Operating Characteristic (ROC) curves, the study provided a comprehensive evaluation, identifying FaceNet-512d as the most robust model with an accuracy of 98.4%, closely approaching human-level performance. Additionally, the study offered practical guidelines for selecting and configuring facial recognition models based on specific use cases and datasets, thereby providing a valuable resource for practitioners aiming to implement efficient and accurate facial recognition systems. By building on the LightFace framework, the 2024 benchmark study not only assessed the performance of existing models but also provided actionable insights for improving facial recognition systems, underscoring the importance of modular, flexible frameworks in advancing the field and setting a new standard for future

research and development in facial recognition technology.

Facial Attribute Analysis

A facial attribute analysis framework was developed by Serengil et al. for age, gender, emotion, and race/ethnicity prediction tasks [14]. It is equipped with pre-processing functionalities similar as in Facial Recognition module mentioned previously, and pre-trained models used for face recognition were trained specifically for facial attribute analysis. Different models were used for the four predictions respectively (e.g. FairFace dataset was used to train the race prediction model). The age, gender and race prediction models were built on the base VGG-Face model. The input layer shape is same as the expected size of VGG-Face (224×224×3) for all three models. The output layers were customized based on different models. For example, the output layer of the gender prediction model consists of two nodes corresponding to "man" and "woman", where the age prediction model's output layer contains 101 nodes corresponding to ages 0 to 100. The emotion model was not built on the base VGG-Face model due to its lower resolution and lack of RGB channels. For age prediction, a **MAE** of 4.65 was obtained with the test set. The gender prediction model has 97.44% accuracy on the same test set with age prediction. The race/ethnicity model has significantly lower performance (68% accuracy), and the emotion model is the worst (57.42% accuracy). This could be due to the fact that the rest two tasks are harder to fulfill.

2.3.2 Relation to Health Monitoring

In the context of health monitoring, facial recognition can provide valuable insights into the emotional states of individuals. Studies have demonstrated the potential of facial expression analysis in healthcare. For example, Guo et al. [15] conducted a comprehensive review of developments in emotion recognition technology over the past decade, and investigated trends and real-world effects of emotion recognition technology. The results justifies its positive impacts such as facilitated remote emotion recognition and enhancement of accuracy of medical diagnosis. Additionally, an exploration of use of facial expression analysis based artificial intelligence (AI) on pain intensity evaluation was carried out by Fontaine et al.[16], the results validated its feasibility, especially for people not able to report appropriately their pain by themselves.

2.4 Action Recognition

Action recognition involves determining the spatial configuration of a person's body parts and is a crucial component of activity recognition and fall detection systems. PySlowFast, a state-of-the-

art framework developed by Facebook AI Research, has demonstrated superior performance in video understanding tasks. By leveraging a combination of slow and fast pathways, PySlowFast captures detailed spatial and temporal information, making it highly effective for analyzing human movements [17]. In this thesis, the integrated project yolo_slowfast [5] was used, which combines the strength of YOLOv5 [18] and Pyslofast.

2.4.1 YOLO_SlowFast

The yolo_slowfast project aims to achieve real-time action detection by integrating YOLOv5 with the SlowFast network. This integration replaces Faster R-CNN with YOLOv5, combining the strengths of YOLOv5 for object detection and the SlowFast network for action recognition, resulting in a processing speed of 24.2 FPS on an RTX 2080Ti GPU.

YOLOv5

YOLO is known for its real-time object detection capabilities, which provide high accuracy and speed in detecting objects within video frames [19]. YOLOv5 is a newer version of YOLO family. Similar to previous versions, it consists of three main parts in its archeticture [20]:

1. **Backbone**: A pre-trained network to extract rich feature representation for images, which is designed using the New **CSP-Darknet53** structure. This structure tackles the problem of redundant gradient information.

2. **Neck**: The neck extracts feature pyramids and helps generalization to objects of different sizes and scales. In YOLOv5, **SPPF** and New **CSP-PAN** structures are utilized.

3. **Head**: Same head is used as YOLOv3 and YOLOv4. The head is responsible for final stage processes. The use of three convolution layers predicts the location of the bounding boxes, objectness scores and the objects classes.

YOLO is significantly faster than R-CNN and Fast R-CNN, making it suitable for real-time applications [5]. When video is decomposed into multiple frames and YOLOv5 is executed for object detection frame by frame, the generated object tracking box will move with the object (Figure 2.3).

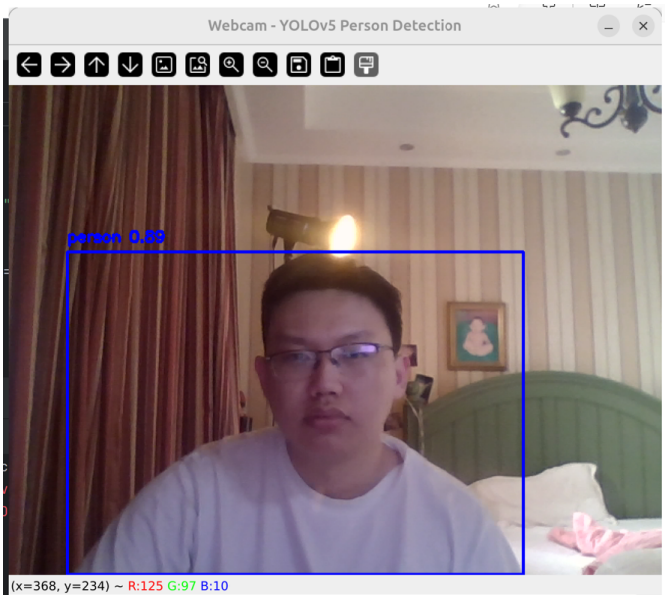


Figure 2.3: Example of object detection using YOLOv5

PySlowFast

PySlowFast [17] is a high-performance and light-weight video understanding pytorch codebase developed by Facebook AI Research for state-of-the-art video backbones for video understanding research on different tasks. It includes implementations of SlowFast networks and other video models, along with tools for data loading, training, and evaluation. The SlowFast network (Figure 2.4) is designed to capture both fast and slow motion in video streams, making it highly effective for recognizing actions over varying temporal resolutions [21]. To achieve behavior detection, video sequences and the information from the detection box are fed into the behavior classification model to obtain behavior categories for each detection box. The behavior classification model includes a slow path that operates at a low frame rate to capture spatial semantics, and a fast path that operates at a high frame rate to capture motion with fine temporal resolution. The fast path is lightweight, reducing channel capacity while learning useful temporal information for video recognition. Experiments were conducted on both action classification and action recognition. The action classification experiment utilizes three datasets without ImageNet pre-training, which are Kinetics-400 (240k training videos and 20k validation videos in 400 human action categories), Kinetics-600 (392k training videos and 30k validation videos in 600 classes), and Charades (9.8k training videos and 1.8k validation videos in 157 classes in a multi-label classification setting of longer activities spanning 30 seconds on average) respectively. For Kinetics-400, the best model provides 2.1% higher accuracy than previous state-of-the-art.

In comparison with those also without ImageNet pre-training, all results are substantially better. This model is 5.9% higher than the previous best result (73.9%). Experiments with ImageNet pretraining for SlowFast networks shows that similar performance (0.3%) for both the pre-trained and the train from scratch (random initialization) variants. Results using the other two datasets also outperform previous best results with lower costs. When it comes to Action Recognition, AVA 2.1 dataset was chosen. It has 211k training and 57k validation video segments, and the standard protocol of evaluating on 60 classes was followed. A person detection model trained with Detectron was adopted. It is a Faster R-CNN with a ResNeXt-101-FPN backbone, and was pre-trained on ImageNet and the COCO human keypoint images. The fine-tuned detector produces 93.9 **AP@50** on the AVA validation set. The network weights were initialized from the Kinetics-400 classification models, where 14k iterations were trained. With these, SlowFast model has a **mAP** of 26.3, which is 5.6 mAP higher than the previous best one with similar settings, and 7.3 mAP higher than that with no optical flow. The final ensemble ranked first in the AVA action detection challenge 2019 achieved with 34.3 mAP accuracy on the test set.

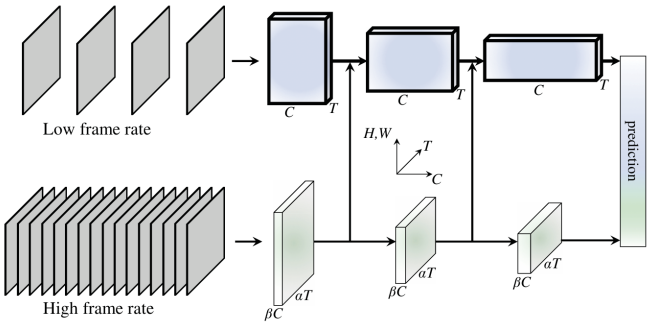


Figure 2.4: SlowFast Network

DeepSort

To further enhance the system, DeepSORT [22] is incorporated for object tracking. It is a computer vision tracking algorithm extended from the **SORT** (Simple Online Realtime Tracking) algorithm, and is used for for tracking objects while assigning an ID to each object. With deep learning introduced using an added appearance descriptor to reduce identity switches, tracking using DeepSORT is more efficient then SORT. DeepSORT enables the system to maintain consistent identities of detected objects across video frames, which is crucial for assigning accurate action labels to these objects as they move. This combination allows for the simultaneous detection of objects and recognition of actions with high accuracy and efficiency.

2.4.2 Relation to Health Monitoring

Several studies have utilized action estimation for health monitoring, demonstrating its potential in various applications. For instance, Kim et al. [23] developed a reliable strategy that integrates machine learning and fully homomorphic encryption to ensure information confidentiality and secure data processing. Homomorphic encryption allows computations to be carried out on encrypted data without needing to decrypt it first, thus preserving privacy. The fine-tuned Fast-HEARshows excellent performance with a low memory usage. In the context secure inference, the throughput is 3.1 times the throughput-optimized nGraph-HE2, and its speed is 613 times faster than the latency-optimized LoLa on average. The space usage is 97.8%–98.5% less than nGraph-HE2 and LoLa, which indicates a significantly smaller memory usage.

Additionally, Gao et al. [24] proposed a novel deep learning architecture named the recurrent 3D convolutional neural network (**R3D**) and validated its effectiveness for remote smart healthcare monitoring. The R3D model serves 3D convolutional network entries as an input to the Long Short-Term Memory (**LSTM**) architecture. Short-term spatial-temporal features are obtained using 3D convolutional network. These features are aggregated by LSTM for the extraction of long-range spatial-temporal information, conveying a high-level of abstraction of human actions. Although simpler than C3D network, the proposed R3D method demonstrates higher accuracy with much faster computation speed (85.7% accuracy with 427 fps for R3D, and 82.3 accuracy with 313 fps for C3D).

These studies highlight significant advances in action estimation technologies for health monitoring. By integrating action recognition with facial recognition, reliable and real-time monitoring solutions could be achieved for elder's remote health monitoring.

2.5 Integration of Pose Estimation and Facial Recognition

The integration of action recognition and facial recognition into a unified framework can offer comprehensive assessments of an individual's health status. In this project, the camera footage will be captured as video clips for further analysis using yolo_slowfast. The processed outputs will then be subsequently displayed, while deepFace conducting real-time analysis on them. This approach combines physical activity monitoring with emotional state analysis, providing a holistic view of a patient's well-being. The developed system would finally be capable of monitoring elderly people in a daily life context. At the end of Phase two, the system aims to detect abnormal behaviors or actions such as falls, identify negative emotions, and alert caregivers to potential health issues timely.

Chapter 3

Facial Recognition

Facial recognition plays a pivotal role in this project, as facial expressions are a natural and direct means for humans to communicate their emotions and intentions [25]. These expressions are key components of non-verbal communication. Accurately recognizing and interpreting these expressions is essential for monitoring the emotional states of elderly individuals within a health monitoring system. In this project, DeepFace [4] is employed to detect faces and subsequently perform face verification and attributes analysis.

3.1 Preliminary Design of Facial Recognition

Face detection was done using OpenCV's Haar CascadesDeepface pre-trained model[26]. DeepFace was then used for both facial recognition [12][13] and facial attribute analysis purposes [14]. The designed algorithm for facial recognition in this project will analyze captured video frames from camera using Deepface, and display analyzed information in real time. It is divided into six sections, each accompanied by corresponding flowcharts to illustrate the process:

1. **Initialization**: This section encompasses the fundamental setup procedures, including loading the face detection model, loading known faces, and initializing video capture. These steps establish the foundational elements required for subsequent operations. The face detection model is pre-trained (Haar Cascade Classifier) and loaded into memory to expedite the detection process during runtime.

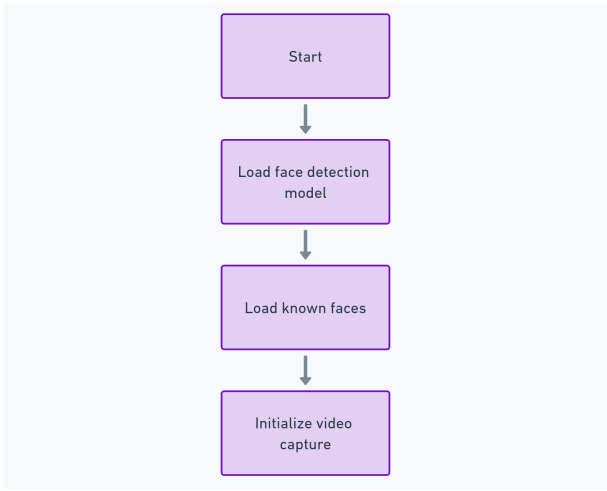


Figure 3.1: Facial Recognition: Initialization Flowchart

2. **Video Capture**: In this section, the algorithm verifies the successful opening of the video device. If the video device fails to open, the algorithm branches into error handling. If successful, the algorithm proceeds to read frames from the video feed. This ensures that real-time video input is consistently available for processing.

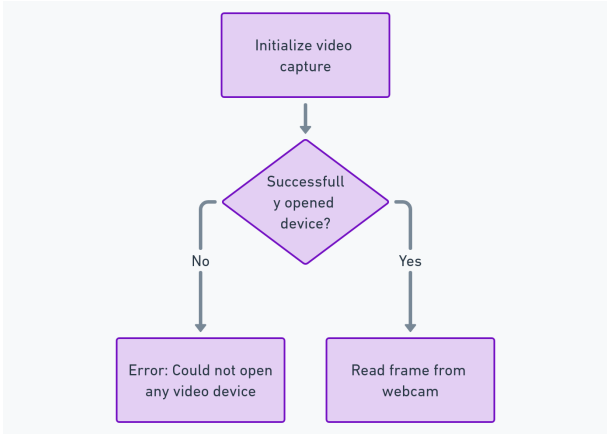


Figure 3.2: Facial Recognition: Video Capture Flowchart

3. **Frame Processing**: This section converts the captured frame to grayscale. Converting the frame to grayscale simplifies the detection process by reducing computational complexity. The detection process involves identifying regions of interest (faces) in the frame, which are then analyzed for facial features and attributes.

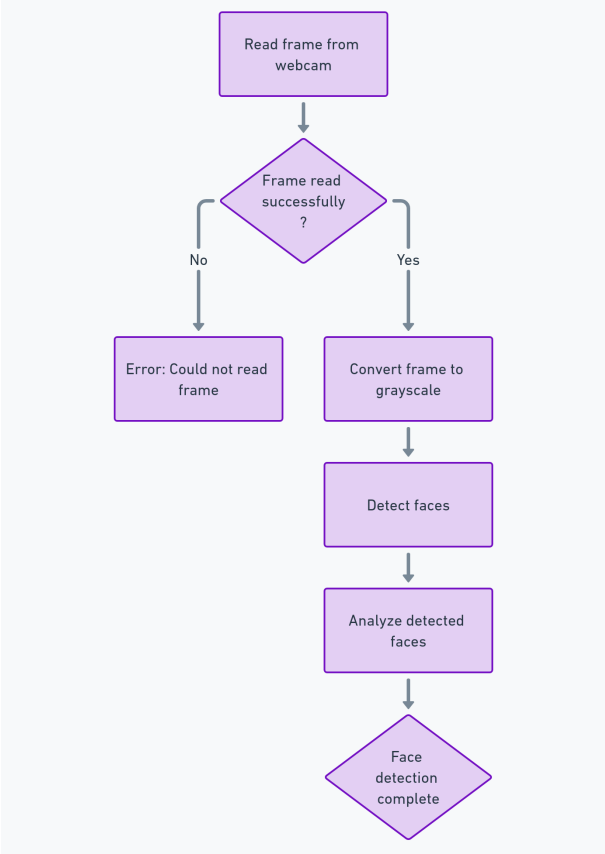


Figure 3.3: Facial Recognition: Frame Processing Flowchart

4. **Face Analysis**: This section carries out post-detection tasks, such as drawing bounding boxes around detected faces, displaying labels, and managing face windows based on inactivity. These steps ensure that the system maintains real-time and efficient face tracking. Bounding boxes visually highlight detected faces, while labels provide additional information such as the identity of known individuals or detected attributes (e.g., age, gender, emotion).

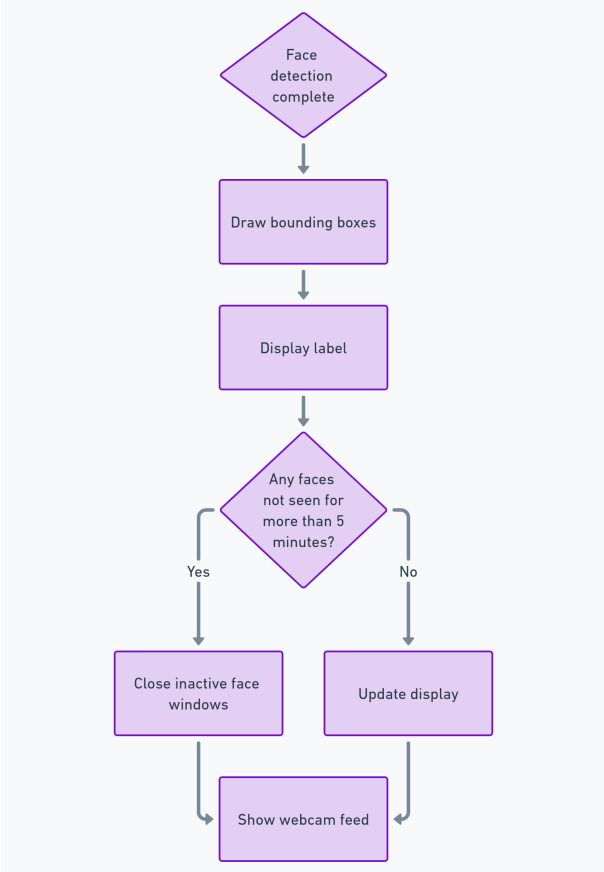


Figure 3.4: Facial Recognition: Face Analysis Flowchart

5. **Results Processing**: This section elaborates on the analysis outcomes, including processing results, calculating average values, and determining if a detected face matches known individuals. This facilitates the identification and information display process. Matching detected faces with known faces involves comparing facial features and attributes to a pre-stored database of known individuals.

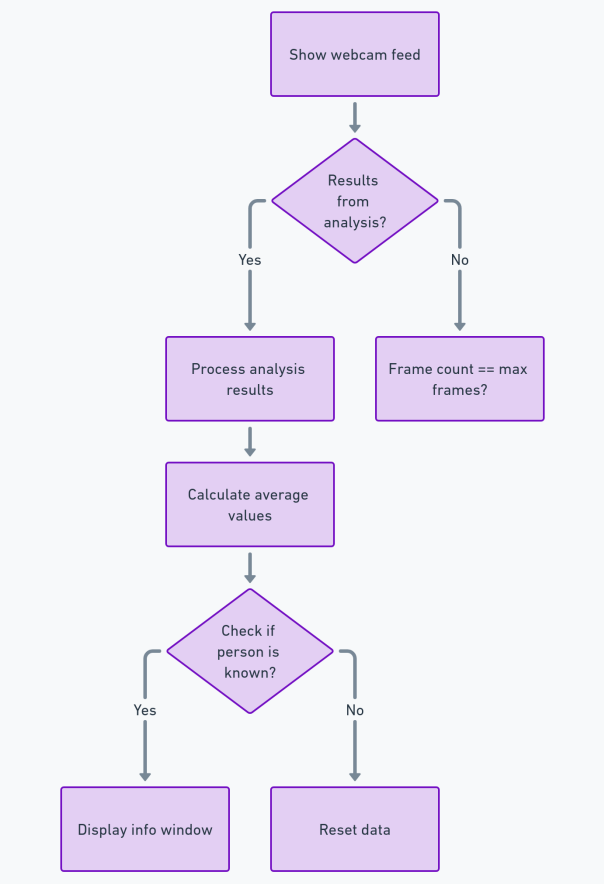


Figure 3.5: Facial Recognition: Result Processing Flowchart

6. **Exit Condition**: When a quit command is issued, the algorithm ensures an orderly shutdown by closing the video feed, releasing allocated memory, and saving any necessary data. This ensures that the system can be restarted or reconfigured without issues.

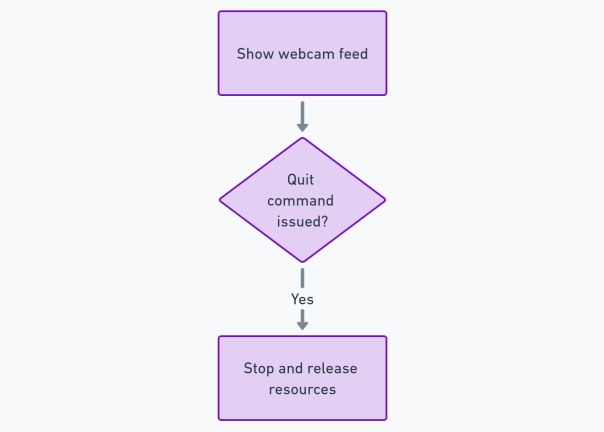


Figure 3.6: Facial Recognition: Exit Condition Flowchart

According to Figure 3.7, the designed algorithm successfully captures video from the camera and displays real-time footage in a separate window. The system accurately locates and frames the person's face, with relevant information displayed nearby. This real-time feedback loop is crucial for continuous monitoring and timely interventions. Through testing, the face detection mechanism is limited by face angles, where side faces can hardly be detected. Meanwhile, successfully detected side faces would have incorrectly analyzed outputs (Figure 3.8). This is due to the lacking information of side faces compared to straight faces. However, the system's ability to detect, analyze, and verify faces still highlights its potential for application in elder health monitoring, and the emerged problems will be tackled during Phasr two of this project. In phase Two, while tests on all four attributes are planned, their dataset scale may be limited depending on the available resources. Race/ethnicity and emotion prediction will be paid more attention to, due to the fact they perform worse previously (Chapter 2.3.1 Facial Attribute Analysis).

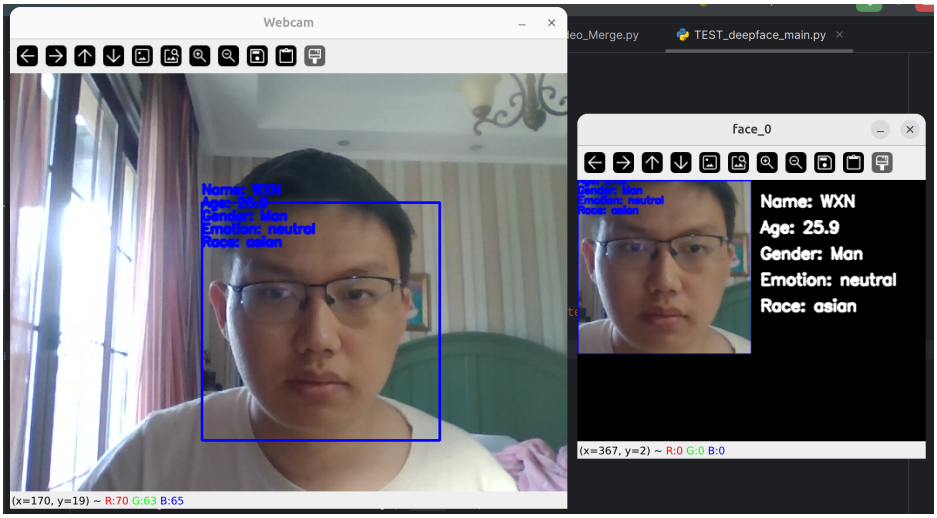


Figure 3.7: Example of Facial Recognition Output

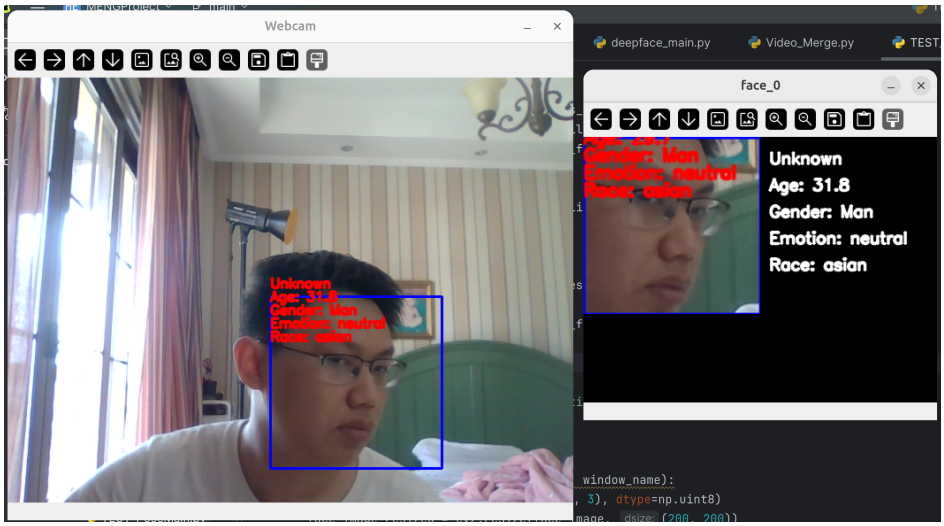


Figure 3.8: Example of Incorrect Facial Recognition Output

Chapter 4

Action Recognition

Action recognition is another critical component of this project, enabling the system to interpret and respond to the physical activities of elderly individuals. Accurate recognition of actions can help monitor daily activities and detect abnormal behaviors that can indicate health problems or emergencies. For this project, the yolo_slowfast [5] GitHub project is employed to achieve real-time action recognition.

4.1 Preliminary Design of Action Recognition

The `yolo_slowfast` algorithm can be called directly for use by specifying the input source. Although this algorithm has the capability to capture video from a camera, it was designed to output the result as a video to the designated directory. The output freezes around each second due to the SlowFast processing time if real-time video is displayed. Therefore, modifications were made to the algorithm. A separate capturing function (explained in Chapter [add chapter number]) records and stores camera footage as one-second video clips. The modified algorithm (five sections) then processes all the video clips in time order and stores the results as videos in designated directory created based on time and date for later use:

1. **Initialization**: The script begins by initializing various components necessary for video processing. It starts with creating an instance of MyVideoCapture, a custom class designed to handle video capture from a specified source. Next, it loads the YOLOv5 model for object detection, configuring the model with specific confidence and IoU thresholds, as well as the maximum number of detections allowed. Following this, the SlowFast model for action recognition is loaded and set to evaluation mode, ensuring it is ready for inference. Lastly, the DeepSORT tracker is initialized using a specified checkpoint, which will be used later for tracking detected objects across video frames.

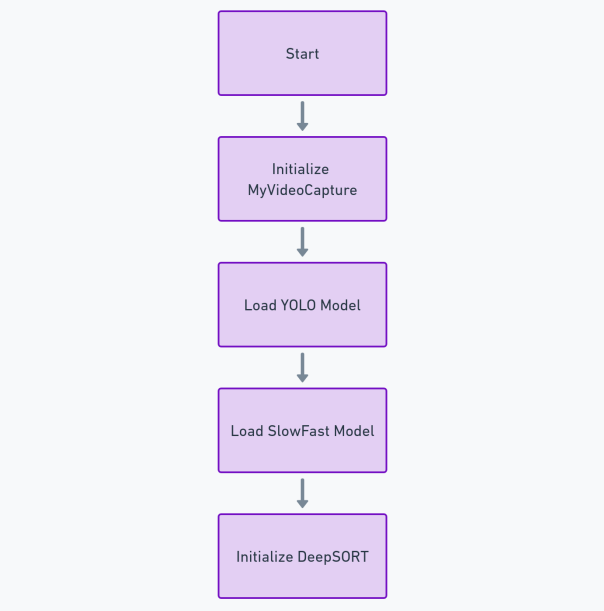


Figure 4.1: Action Recognition: Initialization Flowchart

2. **Directory and Video Handling**: In this section, the script focuses on finding input video directory and process the video files. It identifies video directories based on the current date and time, looking for the most up-to-date input-video directory. Once the directory is identified, the script checks if it contains video files. If no matching directory are found based on the current time and date, the script exits. If directories are found, it proceeds to process the video files within. The script then checks for new video files in the identified directory that have not yet been processed. If no new video files are found, the script waits for a short period before retrying. If new video files are found, the script moves on to process each new video file individually.

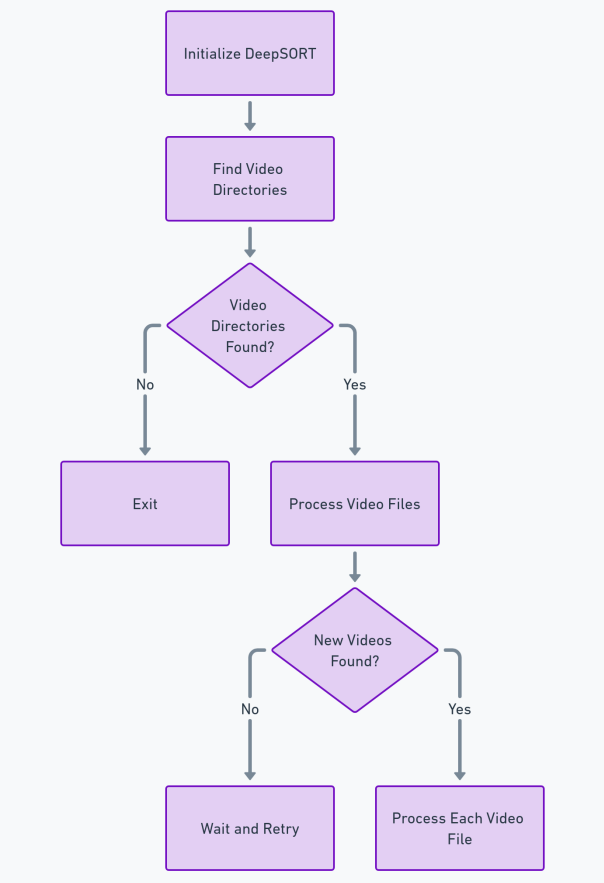


Figure 4.2: Action Recognition: Directory and Video Handling Flowchart

3. **Frame Processing**: For each new video file, the script initiates the frame capture and processing sequence. It uses the MyVideoCapture instance to capture frames from the video. Once frames are captured, the YOLOv5 model is applied to detect objects within those frames. Detected objects are then tracked using the DeepSORT tracker, which updates the tracks of objects across consecutive frames. The script checks if 25 frames have been processed. If fewer than 25 frames have been processed, it continues capturing additional frames. If 25 frames have been processed, the script transforms the captured video clip and the detected bounding boxes, preparing them for input to the SlowFast model.

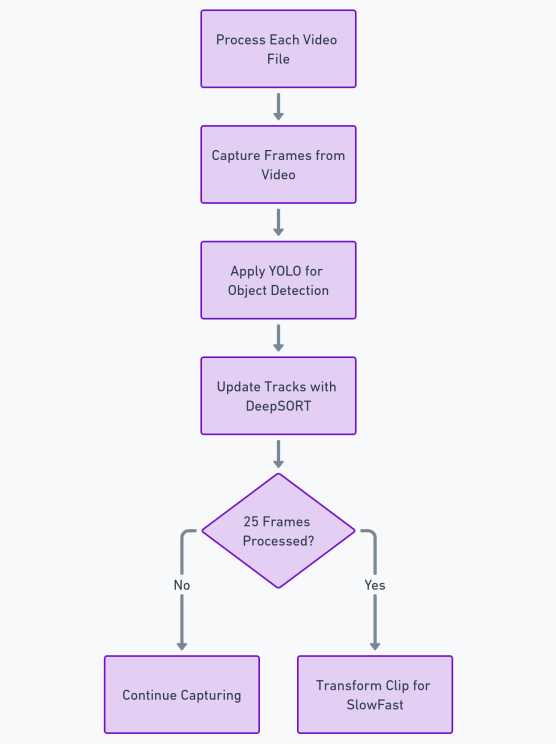


Figure 4.3: Action Recognition: Frame Capture and Processing Flowchart

4. **Action Recognition and Result Overlay**: After transforming the video clip and detected bounding boxes, the script uses the SlowFast model to recognize actions within the video clip. The results of the action recognition, including bounding boxes and labels, are then overlaid on the video frames. This visual overlay helps in understanding which actions were recognized and where they occurred within the video. The script ensures that these results are clearly visible on the processed video frames.

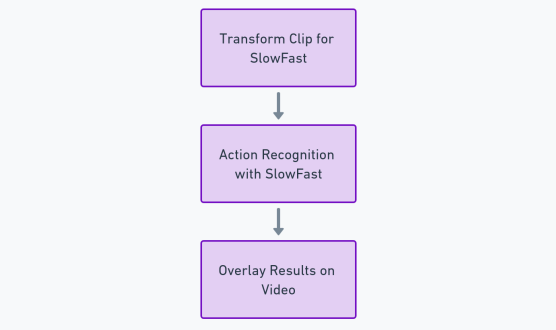


Figure 4.4: Action Recognition: Action Recognition and Result Overlay Flowchart

5. **Video Saving and Looping**: The processed videoes are saved to the designated output directory. The script then concludes the current processing loop. After saving the video, the script checks again if there are any new video files that need to be processed. If no new video files are found, the script waits for a short period before retrying the search for new files. If new video files are identified, the script continues to process each new video file, repeating the frame capture, processing, action recognition, and saving steps as needed. This loop ensures that all new video files are continuously processed and saved.

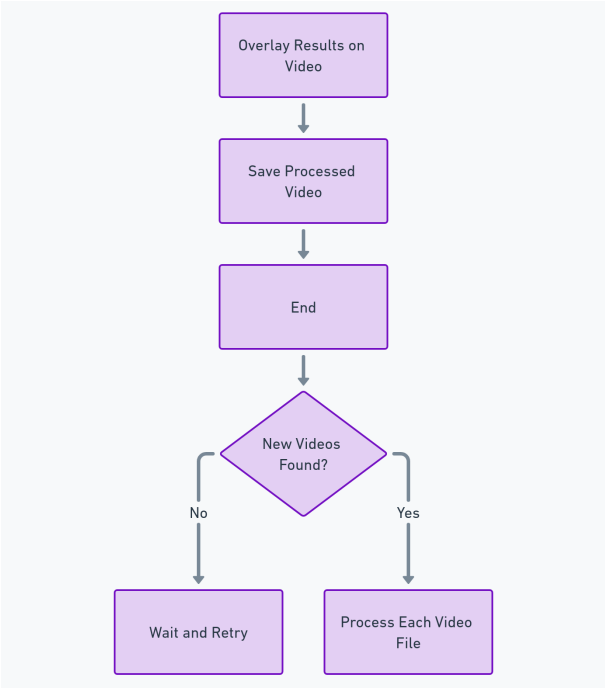


Figure 4.5: Action Recognition: Video Saving and Looping Flowchart

The designed algorithm successfully processes one-second input videos and stores the output videos in the designated directory (Figure 4.6). Detected action (sit) is displayed with relevant information, which is crucial for monitoring and timely intervention. This implementation demonstrates the feasibility and effectiveness of the modified algorithm, highlighting its potential for application in elder health monitoring. Although subsection of Chapter 2.4.1 PySlowFast indicates SlowFast's superior performance over previous models, further experiments and evaluations will be conducted on factors such as accuracy on Phase Two.

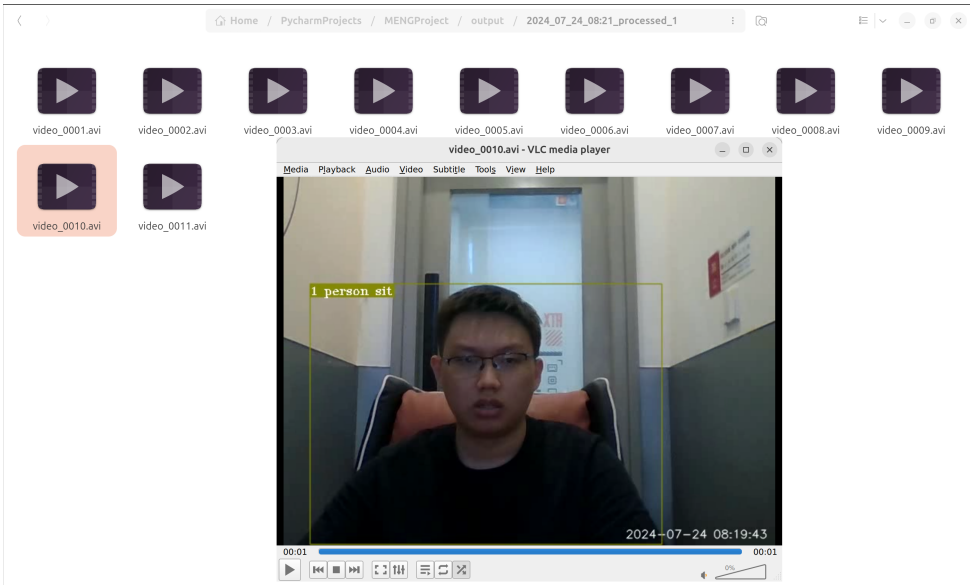


Figure 4.6: Example of Action Recognition Results

Chapter 5

Integration and Further Enhancement

5.1 Input videos Process

A separate script (Figure 5.2) is written that automates the process of recording videos from a webcam, annotating them with timestamps, and running the yolo_slowfast algorithm for video analysis. It starts by creating a unique output directory based on the current timestamp to store the recorded videos (Figure 5.1). The camera footage is then continuously captured as one-second video clips, where each clip is annotated with the current timestamp. These video files are then queued for further processing. Meanwhile, yolo_slowfast is executed to analyze the captured videos. This process runs in separate threads to ensure simultaneous video recording and analysis, with a queue mechanism to manage the flow of recorded video files.

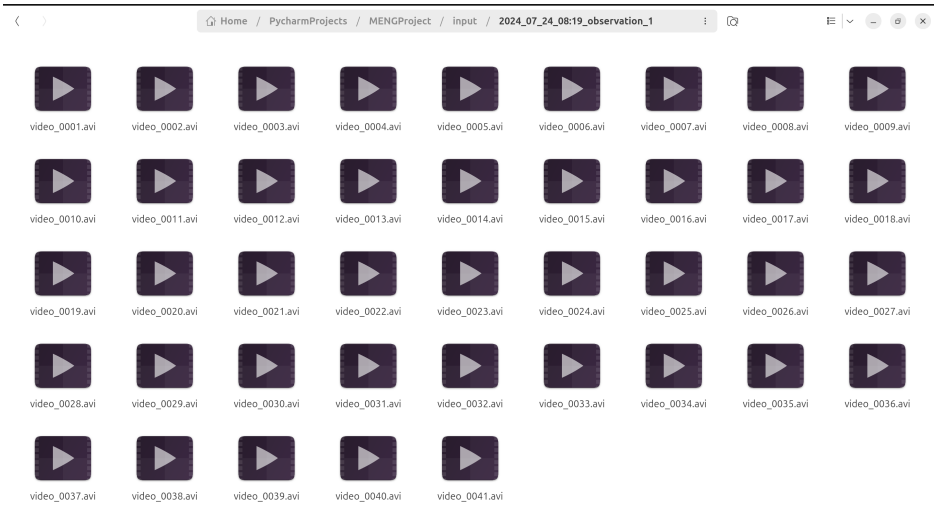


Figure 5.1: Example of Input video files stored at the created directory

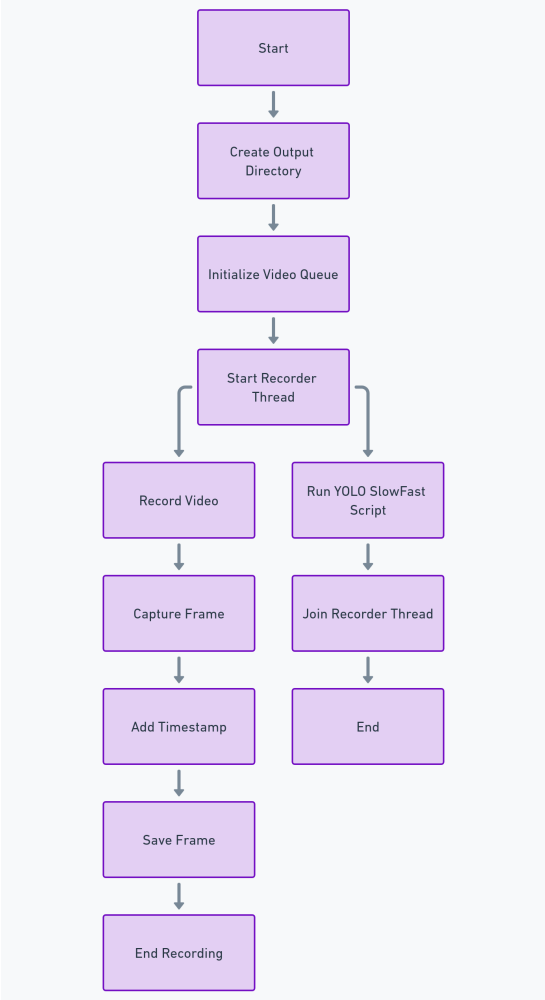


Figure 5.2: Video Recording and Processing Flowchart

5.2 Merging Processed Videos

According to Chapter 4, the processed videos are stored as 1-second clips in the designated directory. This brings inconvenience to the user (e.g. health carer) when accessing them. Thus, another algorithm is designed that turns on a video player, which continuously displays the processed output videos in time order. When there are no subsequent videos in the directory, it is paused until new videos are processed and stored in that directory. In order to implement both facial recognition and action recognition, the previously designed DeepFace algorithm is also merged and applied directly on the video displayed by the video player. The descriptions for the flowcharts of the four sections are provided below:

1. **Initialization**: The initialization and setup section involves preparing the environment for the video player and face analysis. The process begins with carrying out necessary initial setup procedures. This includes loading any configurations or initial parameters required for the application. Following the initialization, it loads images of known individuals from a specified directory. These images are used later for face recognition. Once the known faces are loaded, a Tkinter window is created and configured, and will be used to display the video and analysis results. Finally, a separate thread is initiated dedicated to playing the video files. This thread ensures that video playback runs smoothly without blocking the main application.

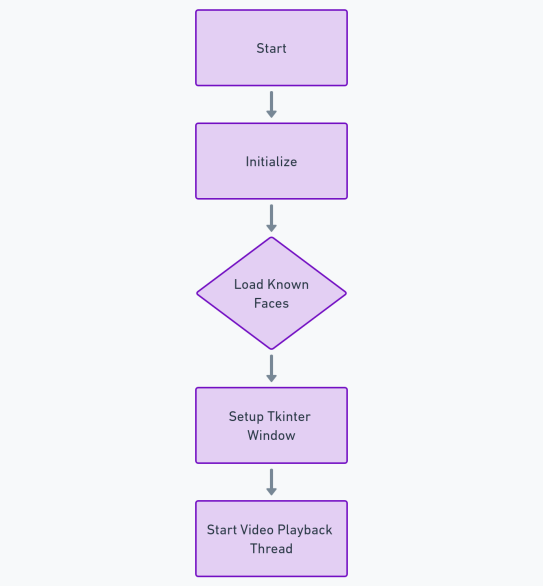


Figure 5.3: Video Merge and DeepFace Integration: Initialization Flowchart

2. **Video Playback Thread**: This thread is the core of the video playback functionality, running continuously until the user closes the window. It starts with checking the latest video directory to ensure that the most recent video files are loaded. The next step scans the base directory for new video files, updating the list of videos to be played. Once the latest video files are loaded, it sorts and prepares these files for playback. For video playing, video frames are played one by one. During playback, OpenCV is used to detect faces within each video frame. If a new face is detected, it is added to the face queue for analysis in the following step. The application then checks if enough frames have been analyzed for each face. If so, the label for the face is locked, displaying detailed information about the face. The next "Display Frame in Tkinter" step ensures that each video frame,

along with any detected face information, is displayed in the Tkinter window. This loop continues until the user decides to close the window. If the window is closed, the process stops.

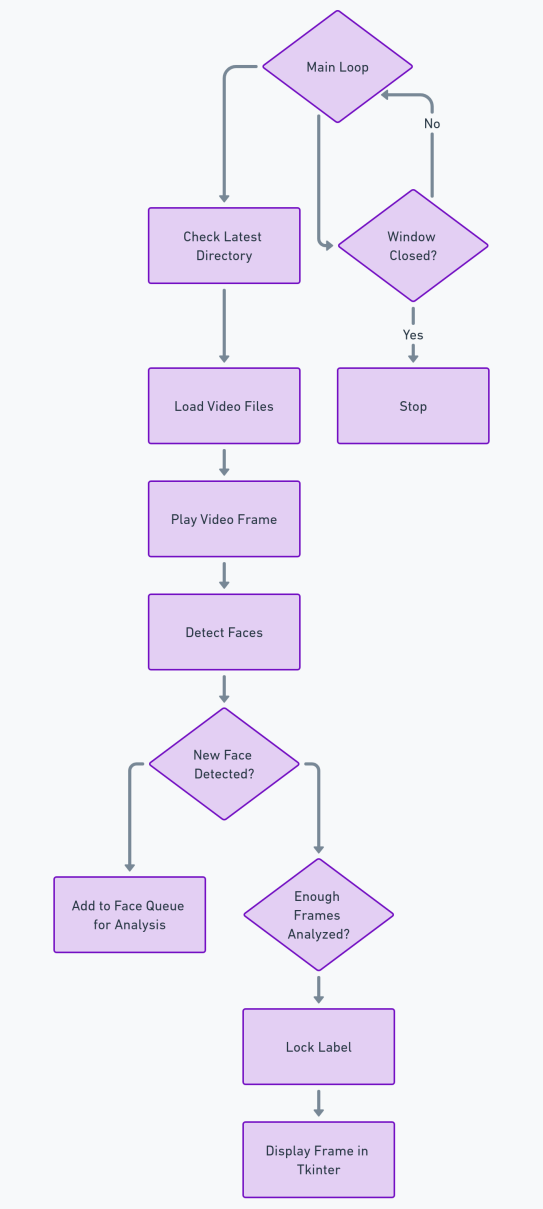


Figure 5.4: Video Merge and DeepFace Integration: Video Playback Flowchart

3. **Face Analysis Thread**: The face analysis thread operates concurrently with the main video playback loop. This thread is responsible for analyzing faces detected in the video

frames. When a face is added to the face queue for analysis, it ensures that a separate thread processes these faces. This separation allows the main video playback to continue smoothly. It then utilizes deepFace to analyze face attributes. The results are then added to the result queue, which is subsequently processed to update the face data and display relevant information on the video frames.

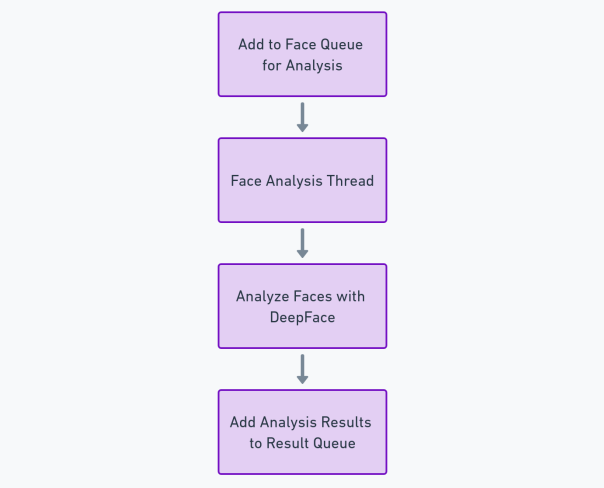


Figure 5.5: Video Merge and DeepFace Integration: Face Analysis Thread Flowchart

4. **Face Data Management**: The process begins detecting faces within each video frame. It then updates the data for each detected face, storing attributes such as age, gender, emotion, and race. The next step averages these attributes over multiple frames to provide accurate results. This helps to smooth out any anomalies and provides a more reliable estimate of the face attributes. The detected face are checked for matches with any known faces loaded during initialization. If a match is found, it locks a label for the known face, displaying relevant information such as the name of the individual. If no match is found, it locks a label indicating that the face is unknown. The final step displays the labeled video frmaes in the Tkinter window. The loop continues to check if the window has been closed, and if it is true, the process ends.

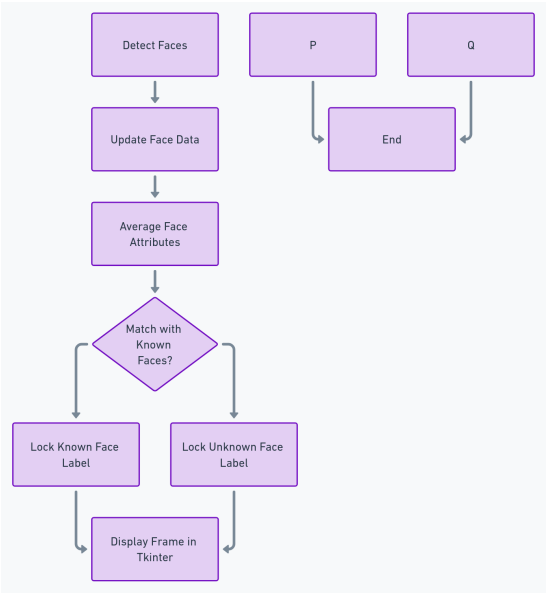


Figure 5.6: Video Merge and DeepFace Integration: Face Data Management Flowchart

From figure 5.7, on the video stream of output videos processed by yolo_slowfast, the face is successfully detected with framework and analyzed attributes displayed. This demonstrates the successful execution of the script, equipping the system with the ability required for real-time display.

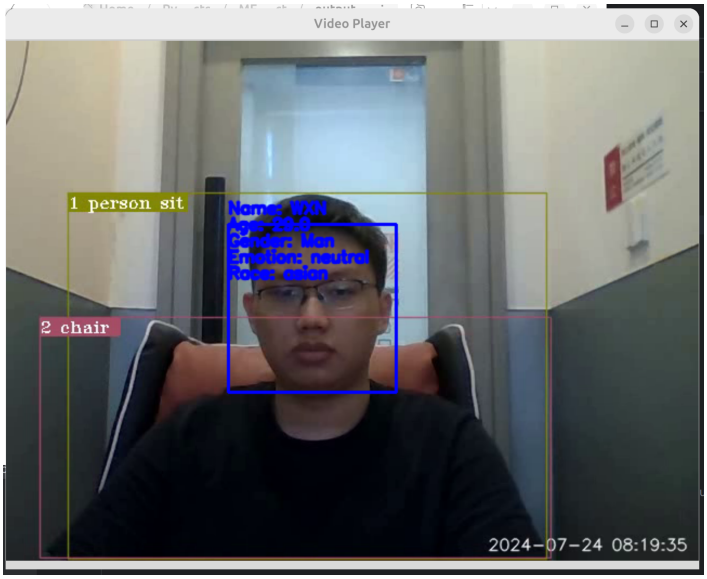


Figure 5.7: Example of Real-time Output Display

5.3 GUI Interface

A simple graphical user interface (GUI) (Figure 5.8) is also created as an initial preparation for user interaction, where the tkinter library [27] is used to manage the execution and termination of the designed algorithms (Chapter 5.1 and Chapter 5.2). When the "Execute YOLO5 Script" button is clicked, it starts both scripts in separate threads to allow them to run concurrently. This is achieved through the subprocess module, which launches each script as a subprocess, and the threading module, which ensures that they run in parallel without blocking the GUI. The "End Script" button is designed to safely terminate these subprocesses if they are still running, resetting their references to ensure that the processes are properly cleaned up. The GUI features a main window titled "Execute YOLO5 Script GUI" with a size of 1600x800 pixels and includes two large buttons for user interaction. This setup provides an intuitive interface for users to control the execution flow of the specified Python scripts.

This algorithm successfully displays a video player window (similar to Figure 5.7), where the displayed footage is approximately 95 seconds behind the actual time. This high lag might be caused by many reasons. An obvious one is the startup time for calling all algorithms, which significantly delays the display of the video. Additionally, the processing time for the algorithms, combined with potential memory usage issues, likely contributes to this high lag. These challenges will be thoroughly examined and addressed in Phase Two. Overall, this system integrates real-time video capture, object detection, action recognition, face recognition as well as facial atrribute analysis, demonstrating the feasibility and potential of using advanced algorithms for elder health monitoring.

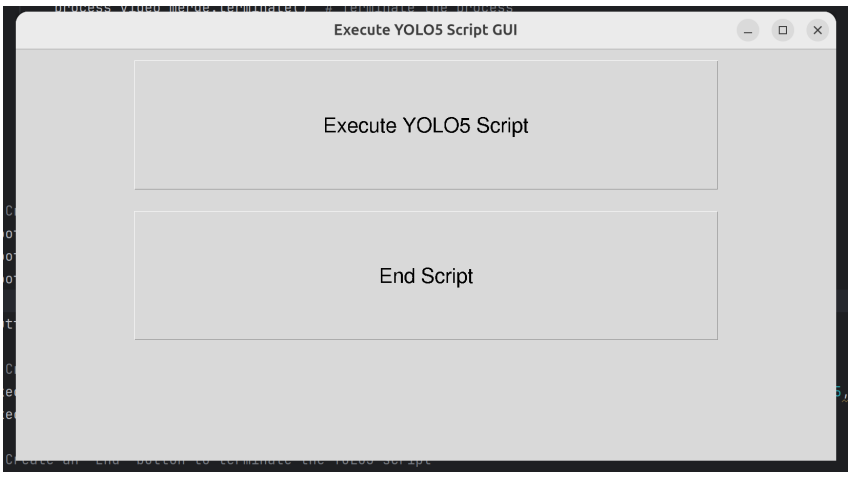


Figure 5.8: Basic GUI Interface

Chapter 6

Plan for Phase two

6.1 Severity Matrix and Alert function

The primary goal of phase two is the design and implementation of a severity matrix, followed by an alert function based on its output. The matrix will read in detected person or people's processed action and facial information, and determine if someone is in danger or poses a danger to their surroundings.

The severity matrix will categorize different combinations of actions and facial expressions into various risk levels. For instance, actions such as fighting or holding one's chest, combined with facial expressions of disgust or fear, would indicate a high-risk situation requiring immediate intervention. Conversely, neutral actions and expressions will be categorized as low risk. Below (Table 6.1) is an example of matrix, although the emotion prediction from deepFace has seven emotions, other emotions might be added or existing emotions might be deleted based on specific needs in phase two. Similarly, action labels might be amended and models would trained to accomplish a best result for the severity matrix.

Based on the risk assessment, the system will trigger an alert function to notify caregivers or relevant authorities. This alert function will provide detailed information about the detected risk, including the type of action, facial expression, and the severity of the situation. The photo captured when the event happened will also be displayed to assist further judgement of severity. The alerts will also be customized to include visual and audio notifications, ensuring timely and effective responses to potential dangers.

Facial / Action	Neutral	Happy	Surprise	Sad	Angry	Disgust	Fear
Stand/Sit/Lie							
Dance							
Quarrel							
Fight							
prolonged immobility							
Holding Chest							

Table 6.1: Severity Matrix for Action and Facial Recognition

The implementation of this severity matrix and alert function involves several key steps:

1. **Data Collection and Annotation:** Prepare dataset of actions and facial expressions, annotated with corresponding risk levels.
2. **Matrix Design:** Explore possible and suitable combinations of actions and facial expressions to form the matrix.
3. **Alert System Development:** Ensure the designed system can effectively communicate risk levels to caregivers or authorities, and minimize incorrect alerts (false positives).

6.2 Enhance Algorithm Robustness

Another key aspect is to test and improve the robustness of facial and action recognition algorithms. This involves solving existing issues mentioned in previous chapters, as well as testing and improving the algorithms ability to handle various real-world conditions such as different lighting, occlusions, and camera angles. The goal is to ensure that the system remains accurate and reliable under diverse environmental conditions, as well as identify any weaknesses and improve the system's overall performance.

6.3 Optimization of Real-Time Processing

According to Chapter 5.3), a latency of around 95 seconds exists. Thus, optimization of real-time processing capability is also crucial. This includes reducing computational load and improving processing speed to ensure that the system can operate efficiently on available hardware. It is crucial for timely detection and notification of abnormal behaviors.

6.4 User Interface Improvement

The graphical user interface (GUI) will also be improved. The current basic GUI will be refined to improve usability and functionality for caregivers. One example can be the caregiver can determine when remote monitoring starts through the GUI. The goal is to create an intuitive and user-friendly interface that facilitates easy interaction with the system.

6.5 Deployment Testing

If applicable, real-word deployment will also be conducted for further testing and identification of areas need improvement. Different entities (e.g. robotic dogs or birds, or camera itself) may be chosen as the monitoring device, while their suitability will be ranked.

Chapter 7

Conclusion

7.1 Summary of Achievements

The project successfully integrated facial recognition and action recognition algorithms, demonstrating an initial design of a comprehensive monitoring system for elderly care. This integration facilitates the simultaneous evaluation of physical and emotional states, providing a holistic view of an individual's well-being. Key achievements include:

1. **Algorithm Integration**: The project combined state-of-the-art tools such as DeepFace for facial recognition and YOLO_SlowFast for action recognition. This integration enabled the system to detect and analyze both facial expressions and physical actions in real-time, which allows for a more nuanced understanding of an individual's condition, which is essential for providing timely and appropriate care.

2. **Preliminary Design and Testing**: Algorithms were successfully developed as individuals first, proving their effectiveness and feasibility in the project. The algorithms were then merged as a whole system, where intial tests validated the system's functionality, demonstrating its ability to detect and interpret facial expressions and actions simultaneously. These tests highlighted the system's potential for real-world applications in remote health monitoring for the elder people.

Overall, the prototype serves as a foundational step towards a more sophisticated and effective remote health monitoring system, with significant potential to enhance elder care.

7.2 Future Work

The next phase of the project will focus on several key areas for further development and optimization of the system:

1. **Severity Matrix and Alert function**: A severity matrix will be designed to evaluate the criticality of detected conditions and a corresponding alert system will notify caregivers of potential issues promptly. This will assist caregivers in prioritizing their responses and providing timely interventions.

2. **Enhance Algorithm Robustness**: Further optimization and testing over the whole system will also be conducted to ensure the reliability and accuracy of the system across different conditions. This will involve existing problems fixing, continuous updates and improvements based on the testing results.

3. **Real-Time Processing**: Real-time processing capability will be optimized.

4. **User Interface Improvement**: More functions and refinements will be carried out on the GUI for better user friendliness.

5. **Deployment Testing**: Finally, real-world deployment and entity choice may be determined if enough resources could be allocated and utilized.

By the end of phase two, the system aims to provide a robust and reliable method for monitoring and assessing the well-being of individuals, particularly in elder care settings. This will enhance the ability of caregivers to respond promptly to emergencies and ensure the safety and well-being of those under their care. Through continuous innovation and optimization, the project aims to provide a new solution for remote health monitoring.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Gary Wells, for his invaluable guidance and support throughout this project. His insights and suggestions have been instrumental in the successful development of this thesis. I would also like to thank my family and friends for their continuous encouragement and support.

References

[1] Vernon, M.: *Supporting older people to live well at home*, Accessed: 2024-06-18, 2016. [Online]. Available: https://www.england.nhs.uk/blog/martin-vernon/.

[2] England, N.: *The role of remote monitoring in the future of the nhs*, Accessed: 2024-06-18, 2020. [Online]. Available: https://transform.england.nhs.uk/blogs/role-remote-monitoring-future-nhs/.

[3] Scotland, N.: *Nhs scotland learning resources*, Accessed: 2024-06-18, 2024. [Online]. Available: https://learn.nes.nhs.scot/49680.

[4] Serengil, S. I.: *Deepface: A lightweight face recognition and facial attribute analysis (age, gender, emotion) library for python*, Accessed: 2024-06-23, 2024. [Online]. Available: https://github.com/serengil/deepface.

[5] Fan, W.: *A realtime action detection frame work based on pytorchvideo*, 2021. [Online]. Available: https://github.com/wufan-tb/yolo_slowfast.

[6] Organization, W. H.: *Ageing and health*, https://www.who.int/news-room/fact-sheets/detail/ageing-and-health, Accessed: 2024-08-05, 2021.

[7] National Statistics (ONS), O. for: *Population estimates for england and wales: Mid-2023*, Statistical bulletin, ONS website, Released 15 July 2024, 2024. [Online]. Available: https://www.ons.gov.uk/releases/populationestimatesforenglandandwalesmid2023.

[8] National Statistics (ONS), O. for: *Living longer: Is age 70 the new age 65?* Statistical bulletin, ONS website, Released 19 November 2019, 2019. [Online]. Available: https://www.ons.gov.uk/peoplepopulationandcommunity/birthsdeathsandmarriages/ageing/articles/livinglongerisage70thenewage65/2019-11-19.

[9] Tan, S., Sumner, J., Wang, Y. *et al.*, 'A systematic review of the impacts of remote patient monitoring (rpm) interventions on safety, adherence, quality-of-life and cost-related outcomes,' *npj Digital Medicine*, vol. 7, p. 192, 2024. DOI: 10.1038/s41746-024-01182-w.

[10] Schütz, N., Knobel, S., Botros, A. *et al.*, 'A systems approach towards remote health-monitoring in older adults: Introducing a zero-interaction digital exhaust,' *npj Digital Medicine*, vol. 5, p. 116, 2022. DOI: 10.1038/s41746-022-00657-y.

[11] Taskiran, M., Kahraman, N. and Erdem, C. E., 'Face recognition: Past, present and future (a review),' *Digital Signal Processing*, vol. 106, p. 102 809, 2020, ISSN: 1051-2004. DOI: 10.1016/j.dsp.2020.102809. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200420301548.

[12] Serengil, S. I. and Ozpinar, A.: 'Lightface: A hybrid deep face recognition framework,' in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, pp. 23–27. DOI: 10.1109/ASYU50717.2020.9259802. [Online]. Available: https://ieeexplore.ieee.org/document/9259802.

[13] Serengil, S. and Ozpinar, A., 'A benchmark of facial recognition pipelines and co-usability performances of modules,' *Journal of Information Technologies*, vol. 17, no. 2, pp. 95–107, 2024. DOI: 10.17671/gazibtd.1399077. [Online]. Available: https://dergipark.org.tr/en/pub/gazibtd/issue/84331/1399077.

[14] Serengil, S. I. and Ozpinar, A.: 'Hyperextended lightface: A facial attribute analysis framework,' in *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, IEEE, pp. 1–4. DOI: 10.1109/ICEET53442.2021.9659697. [Online]. Available: https://ieeexplore.ieee.org/document/9659697.

[15] Guo, R., Guo, H., Wang, L. *et al.*, 'Development and application of emotion recognition technology — a systematic literature review,' *BMC Psychology*, vol. 12, p. 95, 2024. DOI: 10.1186/s40359-024-01581-4.

[16] Fontaine, D., Vielzeuf, V., Genestier, P. *et al.*, 'Artificial intelligence to evaluate postoperative pain based on facial expression recognition,' *European Journal of Pain*, vol. 26, no. 6, pp. 1282–1291, 2022. DOI: 10.1002/ejp.1948. [Online]. Available: https://doi.org/10.1002/ejp.1948.

[17] Fan, H., Li, Y., Xiong, B., Lo, W.-Y. and Feichtenhofer, C.: *Pyslowfast*, https://github.com/facebookresearch/slowfast, 2020.

[18] Jocher, G.: *YOLOv5 by Ultralytics*, version 7.0, May 2020. DOI: 10.5281/zenodo.3908559. [Online]. Available: https://github.com/ultralytics/yolov5.

[19] Ultralytics: *Yolo: A brief history*, Ultralytics website, 2024. [Online]. Available: https://docs.ultralytics.com/#yolo-a-brief-history.

[20] Cherifi, I.: *Yolo v5 model architecture [explained]*, OpenGenus website, 2024. [Online]. Available: https://iq.opengenus.org/yolov5/.

[21] Feichtenhofer, C., Fan, H., Malik, J. and He, K., 'Slowfast networks for video recognition,' *arXiv preprint arXiv:1812.03982*, 2019. [Online]. Available: https://arxiv.org/pdf/1812.03982.

[22] Mallick, S.: *Understanding multiple object tracking using deepsort*, https://learnopencv.com/understanding-multiple-object-tracking-using-deepsort/, Accessed: 2024-07-27, 2024.

[23] Kim, M., Jiang, X., Lauter, K. *et al.*, 'Secure human action recognition by encrypted neural network inference,' *Nature Communications*, vol. 13, p. 4799, 2022. DOI: 10.1038/s41467-022-32168-5.

[24] Gao, Y., Xiang, X., Xiong, N. *et al.*, 'Human action monitoring for healthcare based on deep learning,' *IEEE Access*, vol. 6, pp. 52 277–52 285, 2018. DOI: 10.1109/ACCESS.2018.2869790.

[25] Revina, I. M. and Emmanuel, W. R. S., 'A survey on human face expression recognition techniques,' *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 6, pp. 619–628, 2021, ISSN: 1319-1578. DOI: 10.1016/j.jksuci.2018.09.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157818303379.

[26] OpenCV: *Cascade classifier*, https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html, Accessed: 2024-07-27, 2024.

[27] Foundation, P. S.: *Tkinter — python interface to tcl/tk*, https://docs.python.org/3/library/tkinter.html, Accessed: 2024-07-27, 2024.