**CHAPTER 1**

**INTRODUCTION**

* 1. **PROBLEM INTRODUCTION**

Like the proverbial saying goes, Givers never Lack. Everyone wants to succeed in life, most individuals who wish to know more about the charity giving services or industry and benefits have to visit one of the charities organisations nearest to them or call on personal cell phone to book an appointment. In the recent times, the charity organisation has been growing at a great pace with its peculiar challenges just like any other industry. The main problems faced by charity organisations as identified are as follows:

1. Donors manually book appointments, and the owners of these firms write down same appointments in diary and forget most of them due to the high number of donors.
2. Overwhelming calls from donors who only wish to know more about the services and not necessarily book an appointment.
3. Although, most of these charity organisations have attracted a good number of donors and NGOs, they are yet to achieve the desired number of individuals/donors as originally planned.

Therefore, the aim of this project is to mainly resolve these problems by designing a web-based system to the charity management systems in the charity organisations that includes the following features by just one click on a link:

1. A brief introduction to the charity organisation and a small profile about the organisation owners and volunteers .
2. A page that includes all the services offered by the organisation.
3. Contact details.
4. “Book an appointment now” link.
5. Brief introduction to the charity funding



Figure 1: Coughing in the crows

1. Mobile app that will enable donor to view all these services at their comfort zones and place order and book appointments.
2. The donors should be able to carry out due diligence of the organisations via the web.
   * 1. **MOTIVATION**

The motivation why we started this project is because of the lack of Charity management organizations in India, that is locally meant for Indians, that will reach out to those individuals that really need help from donors and NGOs or even those NGOs That need Help From Other Donors that are willing to help our dear country and don’t know how to go about it or they might even tend to give it to people to distribute and end up not reaching those that the funding was even donated for.



ss

Figure 2: Sneezing in public places

Figure 2 [17] depicting the sneezing in public places, these are the most common ways of spreading a disease

There is no gainsaying that the charity management system is one of the key service industries in the world. In the early stage of industry development, Donors and NGOs were relatively small in scale, mostly small office either found in a mosques or church or occupying small space in an organisation. With the growing concern about one ‘s personal believe, the charity organisation has grown significantly in recent years. Large charity funding chains mushroomed alongside the emergence of a wide variety of charity organisations and technology in the world. Innovation, in an overall social setting, is characterized as any actual article in our current circumstances. For the motivations behind this venture, the emphasis is on a web way to deal with good cause the executives framework. The World Wide Web utilizes the Internet as a vehicle to convey data, including text, illustrations, recordings and sound to anybody with the comparing getting innovation, which today implies a bit of apparatus, or equipment and programming called an internet browser. The Internet permits clients to surf, or investigate, the World Wide Web for data and diversion and has encouraged the improvement of another plan of action, online business. Internet business is the way toward executing business through contact made on the World Wide Web. Inside the business world including the foundation association, has seen extensive change because of web innovation; a completely new arrangement for the scattering of data and for the offer of product has developed to unexpected extents. Web-moved toward business exchanges can be through selling or potentially publicizing products and additionally benefits, learning new aptitude, helping associations among benefactors and NGOs as well as Donors to Donors. This fast innovative move to the web approach is grasped by a few, disappoints others yet influences everybody, intentionally or subliminally. This examination was intended to investigate the web way to deal with good cause the board framework.

* + 1. **PROJECT OBJECTIVE**

The main aims and objectives of this project is to design a web-based app that helps the Donors and The NGO’s Ease their work. Specifically, the aims are to;

1. Design and integrate an automated system to improve the services and decreased the time spent calls and searching for services offered in the donation.
2. Configure a gateway system for online payment to enable individual’s ease of payment from their mobile devices.
3. Design and implement users register page, login and online appointment booking.

Therefore, we propose to build a charity management system for the distribution of donations between charities, giving people the ability to notify about the surplus, and to inform about the poor who need help.

❖ Development of income resources (donation).

❖ Management and distribution of contributions to all the needy and low-income

Families.

❖ Optimum provision and utilization of operational, physical, and human resources.

❖ Organization and maintenance of facilities and family’s data to allow the ease of their access.

❖ Speeding up the practical procedures. Helping decision Makers in their strategic action

Plans.

* + 1. **SCOPE OF THE PROJECT**

This project will be able to supply solutions to charity management firm’s main issues by first enabling individuals to donate online instead of calling, which is seen to be beneficial for both Donors and NGOs. Also, donors can also read about the NGOs history and the variety of services offered and the Helps they gave out all on the web. Secondly, the project will be an easier way to publicizes the Firm by simply being published on the web. By the end of this project, The Charity Organization that will adopt this web approach will hopefully be able to keep track of appointment records, cutting down the number of phone calls received and finally help achieve the desired number of donors. Again, the aim of this project is to encourage the Charity Management system to rely mainly on technology to publicize their firm and communicate with donor/individuals. Finally, the implementation of this project will benefit academic, the Indian society and the world as a whole. This research adds more knowledge to the academic by providing unavailable materials to achieve and aspire in India. Also, a concise report on the implementation of each phase of this project will be developed and further development will be available.

* 1. **RELATED PREVIOUS WORK**

Like earlier mentioned there are a lot of applications and websites that have done similar project but with little impact on the charity organisations. This section will discuss some of these projects and their merits and limitations and present the current project and its advantages and impact on bringing these types of charity management system into our dear country in this helping season.

These are some of the examples of an online donation that helps in donating or getting donations from different sources:

GiveIndia, Atul Satija already builds a charity management website called GiveIndia to serve the need of the non-profit sector. this website is built for charitable organizations to manage their donations and volunteers.

GiveIndia can help with updating databases, monitoring incoming donations, tracking volunteer hours and more. In the recent years, Charities have been using technology to manage their organizations and improve their efficiency. Nowadays, a lot of charities are turning to a charity management website/software to make their jobs easier and more efficient. The number of charities who use these kinds of software has increased by 800% in the last ten years.

Razoo, a popular crowdfunding platform for individuals, teams, or organizations, is a great website for indexing your nonprofit organization to get exposure for your cause. Using Razoo’s nonprofit portal, any registered nonprofit can set up a professionally branded, completely custom charity fundraising page to collect donations at any time.

* 1. **WORK REQUIRED TO BE DONE**

To achieve the objectives of this project, the following tasks need to be completed:

* **Data Collection**: Gather diverse datasets that include images and videos of human movements, facial expressions, and other relevant parameters.
* **Model Development**: Develop and train deep learning models using TensorFlow and other relevant frameworks to analyze the collected data.
* **Integration of OpenPose**: Implement the OpenPose pose estimation model to accurately assess body movements.
* **System Integration**: Combine the different components into a cohesive system that can provide real-time health assessments.
* **Testing and Validation**: Test the system in various settings to ensure accuracy and reliability. Validate the results against established health metrics.
* **Deployment**: Deploy the system in selected environments, such as workplaces, schools, and healthcare facilities, for real-world application and feedback.
  1. **CHALLENGES**

Developing an action-based health prediction system presents several challenges:

* **Data Quality and Diversity:** Ensuring that the collected data is diverse and representative of different populations to avoid biases in the model.
  + Data Collection: Collecting high-quality data that is representative of a diverse population is crucial. Ensuring that the data covers various age groups, ethnicities, genders, and health conditions can be challenging but necessary to avoid biases and ensure the model's generalizability.
  + Data Labeling: Accurate labeling of data for training the models is labor-intensive and requires expertise. Mislabeling can lead to incorrect predictions and reduce the system's reliability.
  + Privacy Concerns: Collecting and using personal health data raises significant privacy and ethical issues. Ensuring compliance with data protection regulations (such as GDPR) and securing informed consent from participants are vital steps in the data collection process.
* **Real-Time Processing**: Achieving real-time analysis and predictions without compromising accuracy.
  + Computational Resources: Real-time processing of health data, especially using deep learning models, requires substantial computational power. Balancing the need for quick, real-time analysis with the limitations of available hardware can be challenging.
  + Latency: Minimizing latency to ensure the system provides timely health assessments is critical. High latency can lead to delayed interventions, reducing the system's effectiveness.
* Integration of Multiple Parameters: Effectively integrating various health parameters to provide a comprehensive assessment.
  + Complexity of Integration: Integrating various health parameters such as body movement, appearance, and potentially mental health indicators adds complexity to the system. Developing robust algorithms that can seamlessly combine these parameters is challenging.
  + Interpreting Correlations: Understanding and accurately interpreting the correlations between different health parameters requires sophisticated modeling and validation techniques. Incorrect interpretations can lead to false health assessments.
* **Privacy and Security**: Ensuring that the system complies with privacy regulations and protects users' personal health data.
  + Data Security: Ensuring the security of health data, both in transit and at rest, is paramount. Implementing robust encryption methods and secure storage solutions is necessary to protect sensitive information from breaches.
  + User Trust: Building user trust in the system's ability to handle their data securely is crucial. Transparency in data handling practices and obtaining necessary certifications can help in gaining user trust.
* **Scalability**: Designing the system to be scalable and adaptable to different environments and use cases.
  + Scaling the System: Designing the system to handle large-scale deployment across various settings, such as workplaces, schools, healthcare facilities, and public spaces, presents scalability challenges. The system must maintain performance and accuracy as it scales.
  + Adaptability: Ensuring that the system can adapt to different environments and user needs is essential. Customizing the system for specific settings without compromising its core functionality can be challenging.
* **Technical and Operational Challenges**
  + Algorithm Robustness: Developing algorithms that can perform reliably under different conditions and with varying data quality is challenging. The algorithms must be robust to noise and capable of handling incomplete or imperfect data.
  + Maintenance and Updates: Regular maintenance and updates of the system are necessary to ensure it stays current with the latest medical knowledge and technological advancements. Managing these updates without disrupting the system's operation is crucial.
  + Interoperability: Ensuring that the system can work seamlessly with existing healthcare infrastructure and other health monitoring devices is important. Interoperability challenges can arise from differences in data formats, communication protocols, and system architectures.
* **Ethical and Social Challenges**
  + Ethical Considerations: The use of AI in health monitoring raises ethical questions regarding the potential for misuse, discrimination, and bias in the algorithms. Addressing these concerns through ethical AI practices is necessary.
  + User Acceptance: Gaining user acceptance and ensuring they are comfortable with using the system involves addressing concerns about privacy, data security, and the accuracy of health predictions. Education and clear communication about the system's benefits and limitations can help in achieving user acceptance.
  + Regulatory Compliance: Navigating the complex regulatory landscape of healthcare technology requires ensuring compliance with various local, national, and international regulations. This involves regular audits and adherence to standards set by healthcare authorities.
  1. **ORGANIZATION OF THE REPORT**

This report is organized into several chapters, each focusing on different aspects of the project:

* **Chapter 1**: Introduction: Provides an overview of the problem, motivation, project objectives, scope, related work, required tasks, challenges, and report organization.
* **Chapter 2**: Literature Review: Discusses previous research and methodologies related to health prediction and monitoring.
* **Chapter 3**: Methodology: Details the methods and technologies used in the project, including data collection, model development, and system integration.
* **Chapter 4**: Implementation: Describes the implementation process, including the development and testing of the system.
* **Chapter 5**: Results and Discussion: Presents the results of the system's performance and discusses their implications.
* **Chapter 6**: Conclusion and Future Work: Summarizes the project's findings and outlines potential directions for future research and development.

This structure ensures a comprehensive and logical presentation of the project's development and outcomes, providing a clear understanding of the action-based health prediction system and its potential impact.

**CHAPTER 2**

**LITERATURE REVIEW**

* 1. **EXISTING SYSTEMS**

In the realm of health monitoring and public safety, existing systems have been developed to address various aspects of health and security. Among these, action detection systems and mask detection systems have gained significant attention, especially in light of recent global health crises. This chapter explores the existing systems in these areas, detailing their methodologies, applications, and limitations.

* + 1. **ACTION DETECTION SYSTEMS**

Action detection systems are designed to recognize and analyze human activities in real-time. These systems have broad applications ranging from surveillance and security to healthcare and sports. The primary objective of action detection systems is to interpret and classify human actions based on video or sensor data.

**Methodologies**

1. **Traditional Computer Vision Techniques:**
   * **Optical Flow**: This method involves tracking the motion of objects in a sequence of frames. It captures the direction and speed of movement, which can be used to infer actions.
   * **Background Subtraction:** By distinguishing moving objects from a static background, this technique helps in isolating and analyzing human actions.
2. **Machine Learning Approaches:**
   * **Support Vector Machines (SVM):** SVMs are used to classify actions by finding the optimal hyperplane that separates different action categories in the feature space.
   * **Random Forests:** These are used to aggregate the results from multiple decision trees to improve the accuracy of action recognition.
3. **Deep Learning Techniques:**
   * **Convolutional Neural Networks (CNNs):** CNNs have revolutionized action detection by automatically learning hierarchical features from raw data. They are particularly effective in recognizing spatial patterns.
   * **Recurrent Neural Networks (RNNs):** RNNs, including Long Short-Term Memory (LSTM) networks, are employed to capture temporal dependencies in sequential data, making them suitable for action detection over time.
   * **Two-Stream Networks:** These networks process spatial and temporal data separately and then combine the results, enhancing the detection of complex actions.

**Applications**

* + **Surveillance and Security:** Action detection systems are used in public and private security to monitor activities and detect suspicious behavior. For instance, identifying loitering, running, or aggressive actions can help in preventing crimes and ensuring public safety.
  + **Healthcare:** In healthcare, these systems monitor patient activities, ensuring they follow prescribed exercises or identifying fall incidents among the elderly.
  + **Sports Analytics:** Coaches and analysts use action detection to study athlete performance, improve training methods, and prevent injuries by analyzing movement patterns.
  + **Human-Computer Interaction (HCI**): Action detection facilitates more natural interactions with devices through gesture recognition and motion-based controls.

**Limitations**

* + **Data Dependency:** High-quality and diverse datasets are required to train effective models. Lack of sufficient training data can lead to poor performance.
  + **Computational Complexity:** Real-time processing of video data requires substantial computational resources, which can be a limiting factor for large-scale deployment.
  + **Environmental Variability**: Changes in lighting, background, and occlusions can significantly affect the accuracy of action detection systems.
  + **Privacy Concerns:** Continuous monitoring and data collection raise privacy issues, especially in surveillance applications.
    1. **MASK DETECTION SYSTEMS**

Mask detection systems have become crucial in public health efforts, particularly during the COVID-19 pandemic. These systems are designed to detect the presence or absence of face masks on individuals in real-time, helping to enforce public health guidelines and reduce the spread of infectious diseases.

**Methodologies**

1. **Traditional Image Processing Techniques:**
   1. **Haar Cascades:** Early mask detection systems used Haar cascades to detect faces and then applied additional classifiers to determine if a mask was present.
   2. **Histogram of Oriented Gradients (HOG):** This method involves feature extraction to detect faces and masks, often combined with SVM for classification.
2. **Deep Learning Techniques:**
   1. **Convolutional Neural Networks (CNNs):** CNNs are extensively used for mask detection due to their ability to automatically learn features from images. Models like ResNet, MobileNet, and InceptionNet have been adapted for this purpose.
   2. **Single Shot Multibox Detector (SSD):** SSD is a popular object detection framework that has been modified for mask detection. It can simultaneously predict bounding boxes and classify whether a face is masked or not.
   3. **You Only Look Once (YOLO):** YOLO models provide fast and accurate object detection, making them suitable for real-time mask detection in crowded environments.
3. **Hybrid Approaches:**
   1. **Ensemble Methods:** Combining multiple models or methods can improve accuracy. For example, using both HOG and CNNs to leverage the strengths of traditional and deep learning techniques.
   2. **Transfer Learning:** Pre-trained models on large datasets can be fine-tuned for mask detection, reducing the need for extensive training data specific to masks.

**Applications**

* **Public Health and Safety:** Mask detection systems are deployed in public spaces, transportation hubs, and commercial establishments to ensure compliance with health guidelines. Automated alerts can be triggered if individuals are detected without masks.
* **Workplace Safety:** In corporate environments, mask detection helps in maintaining a safe workplace by ensuring employees adhere to mask-wearing protocols.
* **Education Institutions:** Schools and universities use these systems to monitor compliance among students and staff, helping to prevent outbreaks.
* **Healthcare Facilities:** Hospitals and clinics use mask detection to safeguard both patients and healthcare workers, ensuring a sterile environment.

**Limitations**

* **False Positives and Negatives:** Mask detection systems can produce errors, especially in cases of partial occlusion, improper mask-wearing, or varied mask designs. False positives (incorrectly detecting a mask) and false negatives (failing to detect a mask) can undermine the system's reliability.
* **Diverse Environments:** Variations in lighting, face orientation, and background can affect the accuracy of mask detection systems.
* **Real-Time Processing:** Achieving real-time detection with high accuracy requires robust computational resources, which can be a challenge in large-scale implementations.
* **Privacy and Ethical Concerns:** The use of face detection technology, even for mask detection, raises privacy issues. There is a need for stringent data protection measures to ensure user privacy is maintained.
  1. **SEARCH STRATEGY OF THE PROJECT**

The study used the PRISMA model, Figure 2 for the selection of studies. For this, a comprehensive search was taken out by the use of Google Scholar and IEEE Xplore. The main keywords used here for selecting the papers were “Health Prediction”, “CNN”, “Sentimental Analysis”, “Emotion Recognition”, “Body movement analysis”, and “Laban movement analysis”, “CNN advancements”, “Skeleton movement of human”, “Disease Recognition”, “OpenVINO”. Figure number 2 depicts the PRISMA model that consist of the sorted resources which are used in carrying out this study. After searching a lot, a total number of 49 studies were selected; duplicate studies were eliminated with the help of the “Find Duplicates” feature of EndNote software. After analysing the process, we retained a set of 49 records. The 3 exclusion methods were used, i.e. (i) studies which are non-AI, (ii) irrelevant articles, and (iii) articles having less data. After applying these criteria’s, 10, 15, and 10 studies were marked as E1, E2 and E3, respectively. The final selection led to studies for this work.

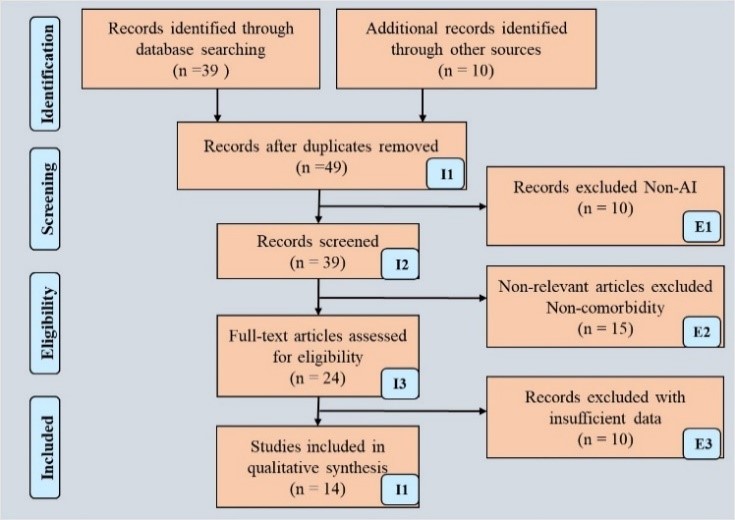


Figure 3: Prisma Model

* 1. **LITERATURE SURVEY**

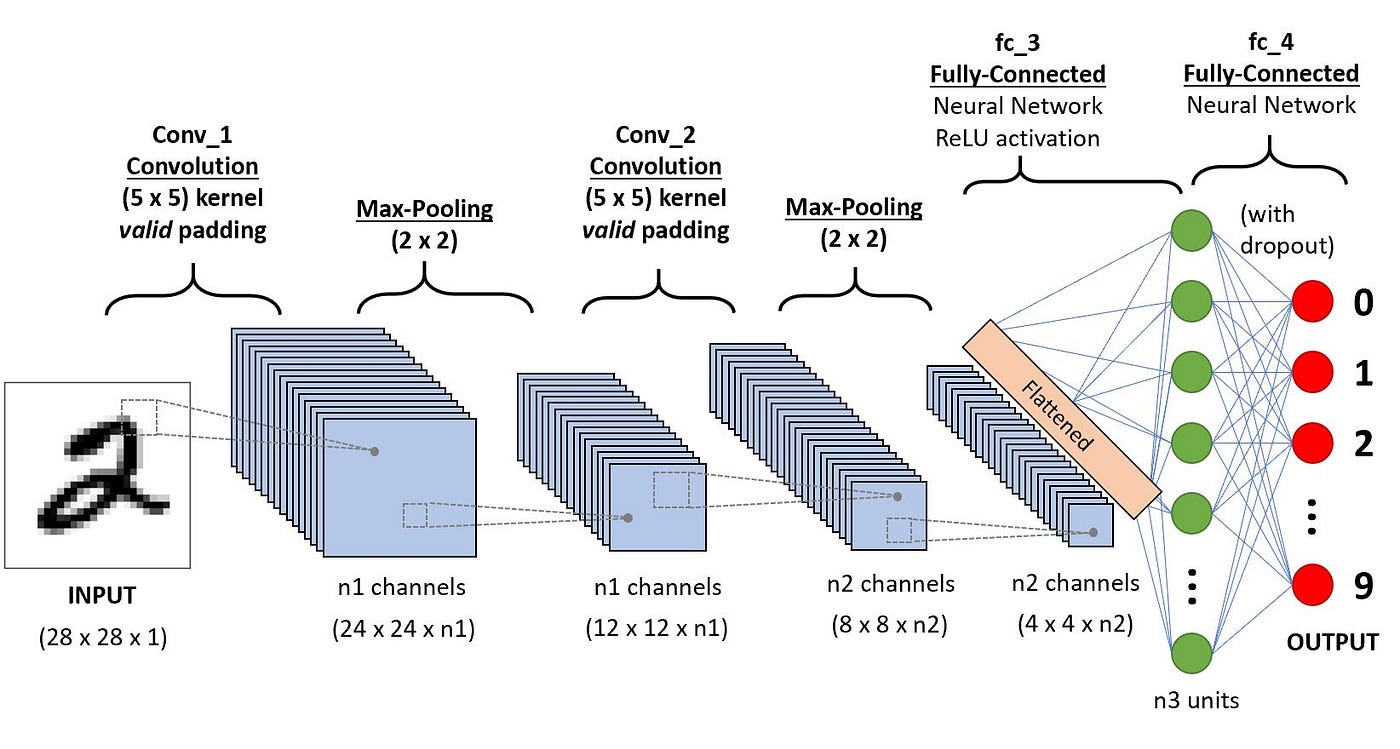


Figure 4: The basic CNN architecture

#### **Sultana, et al. [1]** shed light on the challenges faced by Convolutional Neural Networks (CNNs) due to the limited availability of large training datasets, which hampers their efficiency and performance. Moreover, the absence of standardized techniques and insufficient computational power further restricts their capabilities. The researchers conducted rigorous training and testing of various model variations based on conventional CNN architectures, emphasizing the need for deeper and more extensive networks. While Capsule Networks (CapsNet) show promising results on datasets like MNIST, their full potential is yet to be fully realized and explored.

#### **Alzubaidi, et al. [2]** present an in-depth review focusing on diverse aspects of CNN, ranging from convolutional architectures to applications in areas like flora disease detection and medical image analysis. They highlight CNN's widespread adoption and effectiveness in surpassing human capabilities in tasks such as image classification. The paper underscores the significance of re-used data and parameter application across input layers, illustrating the potential of CNNs in various domains, including medical imaging and object detection.

***Sladojevic, et al. [3]***  provide insights into plant disease recognition using computational intelligence, emphasizing the importance of image pre-processing techniques for optimal feature extraction. They discuss the role of bias in regulating neuron output and highlight the significance of fine-tuning parameters to enhance network accuracy dynamically. The paper elucidates the entire process, including data augmentation, pre-processing, and training functions used for Deep CNNs, offering a comprehensive understanding of disease recognition methodologies.

***Shin, et al. [4]***delve into medical image classification systems, exploring three major approaches to CNN training, including training from scratch, leveraging pre-trained features, and unsupervised pre-training. By employing cross-validation techniques and evaluating different CNN architectures, they demonstrate significant improvements in patch-based classification performance. The authors emphasize the benefits of deep CNN architectures with 7 or 21 layers, particularly in learning from large image datasets for Computer-Aided Detection (CADe) applications.

***Denecke and Deng [5]*** investigate sentiment analysis in the medical field, focusing on analysing facial expressions to glean insights into patient health. They highlight the prevalence of stop words in various medical contexts and discuss the utility of sentiment analysis methods in gathering patient information and detecting critical health indicators during treatment. By analysing sentiment in drug reviews, medical blogs, interviews, and clinical reports, the paper underscores the potential of sentiment analysis in augmenting patient care and treatment outcomes.

***Kidziński, et al. [6]*** explore the predictive capabilities of deep learning models in predicting walking pace, gait characteristics, and knee motion from video datasets. They achieve high correlations between model predictions and ground-truth motion data, demonstrating the potential for advanced smartphones with high-resolution cameras to capture detailed motion data. The paper underscores the importance of leveraging deep learning models for analyzing human motion patterns, offering insights into applications in healthcare, sports science, and rehabilitation.

***Demidovskij*, *et al. [7]*** introduced the utilization of OpenVINO technology inference engine for object recognition through video dataset analysis. The paper outlines current software solutions for evaluating model performance, precision, and accuracy. It also emphasizes the importance of enhancing neural network (NN) operations through computing model optimizations and hardware parallelization. Additionally, the authors provide a detailed overview of the modified documentation for the OpenVINO DL Working environment, aimed at improving NN deployment and quality control. The results obtained from this research not only contribute valuable outputs but also pave the way for further exploration and refinement in this field.

***Hassouneh, et al. [8]*** present a novel algorithm for implementing a real-time emotion recognition system, addressing the complexities arising from facial articulations and EEG signals. This algorithm demonstrates robust performance even under challenging conditions such as low or uneven lighting, subject movement (up to 25° rotation), diverse background images, and varying skin textures. The system's versatility extends to aiding physically disabled individuals and assisting autistic youth in understanding the emotions and sentiments of others. The outputs reveal impressive emotion detection rates of up to 87.25% for EEG signals and 99.81% for facial expressions.

***Patel, et al. [9]***  offer an extensive and systematic survey of state-of-the-art approaches for facial emotion recognition (FER) in images, highlighting various factors that influence the efficacy of these techniques. By identifying current challenges and research gaps, the paper underscores the need for continued investigation in areas such as emotion detection in images and FER in 3D face shape models. Despite recent advancements, FER remains a significant research frontier with ongoing opportunities for innovation and improvement.

***Ahmed, et al. [10]*** Emotion recognition through the detection and analysis of body movements represents an emerging trend in contemporary research. This innovative approach offers numerous advantages, including applications in biometric patient monitoring, biometric security systems, robotics, and motion-controlled gaming, facilitated by sophisticated computational models. Fundamental to this recognition process are the distinctive parameters observed during human locomotion. For instance, during walking, crucial parameters such as the angle of motion in the hand or arm, as well as the dynamics of the upper body region, serve as key indicators for emotion recognition. Similarly, when a person is seated, factors such as spatial extension and elbow angle play significant roles in identifying emotional states. Moreover, it's important to note that the nuances of human motion patterns are influenced by various factors including gender, cultural background, and individual physiological characteristics, contributing to the complexity and richness of the emotion recognition process.

***Rawat and Wang [11]*** This review paper offers an exhaustive examination of image classification methods employing Convolutional Neural Networks (CNNs). It delineates the evolution of CNNs' development into distinct phases: from their early stages, through their pivotal role in the resurgence of Deep Learning (DL), to their ongoing advancements in recent years. The focus lies particularly on the progressive refinement of CNN-based image classification techniques, with attention to various critical parameters such as architectural design, optimization strategies, computational efficiency, regularization mechanisms, and supervision components. Despite previous successes in different epochs of CNN evolution, there is a notable surge in Deep Convolutional Neural Networks (DCCNs) concerning their application in image classification, marking a significant advancement in this field.

***Awais, et al. [12]*** This paper provides insights into various comparative studies focused on IoT-based methods, systems, and frameworks for analysing human emotions that are currently in use globally. Through these studies, we learn about the latest advancements in detecting and interpreting physiological signals to analyse emotions. Utilizing a Long Short-Term Memory (LSTM)-based data-learning model, the system is designed to identify and differentiate a range of emotional expressions such as anger, happiness, amusement, sadness, fear, and more. The model achieves a remarkable level of efficiency and performance, with an accuracy rate exceeding 95%. This high degree of accuracy underscores the model's capability to effectively recognize and categorize human emotions.

***Radaideh, et al. [13]*** In this research paper, the authors have conducted an extensive examination of various sentiment analysis techniques and algorithms. Their findings indicate that sentiment analysis is a highly effective method for real-time applications. Specifically, they determined that the Naïve Bayes algorithm outperforms the Recurrent Neural Network (RNN) in terms of accuracy and precision. The model they propose aims to deliver superior precision, accuracy, and overall performance compared to all previously established models. Furthermore, there is a noticeable positive reception among individuals towards this rapid transition. With a well-crafted strategic plan, any negative feedback can be managed effectively, paving the way for the successful implementation of Industry 5.0.

* 1. **COMPARISON OF VARIOUS SURVEY PAPERS**

Table 1: Comparative study of various survey papers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SN** | **Author/Year** | **Architecture** | **Domain** | **Application** | **Approach/Strategy** |
| **1** | *Sultana, et al. [1] 2018* | Advancement in image classification by using CNN | Non-health care | Image Classification | Proposed an image classification model, AlexNet, demonstrating superior performance compared to other CNN models. |
| **2** | *Alzubaidi, et al. [2] 2021* | Review of DL: Concepts, CNN challenges, architectures, directions, future applications, | Non-health care | Image Classification | Developed a highly efficient and accurate image classification model, showcasing improved performance metrics. |
| **3** | *Sladojevic, et al. [3]* 2016 | Deep NN- recog. of flora diseases by leaf image classification | Health care | Disease Recognition | Introduced a disease recognition model achieving a remarkable accuracy of 96.3% after extensive training iterations and adjustments. |
| **4** | *Shin, et al. [4]* 2016 | Deep CNN for computer-aided detection: CNN dataset characteristic, transfer learning and architectures, | Non-health care | Image Classification | Studied about the various CNN techniques used in the medical field like X-rays, CT scan etc.  Addressed challenges in LN tasks and CADe problems, proposing innovative solutions to overcome these limitations. |
| **5** | *Denecke and Deng [5]* 2015 | Sentiment analysis in medical era: challenges in this and future opportunities | Health care | Emotion recognition and feedback recognition (Good, medium, bad) | Conducted a study on sentiment analysis techniques to analyze textual data, focusing on the proportion of words relevant to personal health data collection. |
| **6** | *Kidziński, et al. [6]* 2020 | Deep NN quantitative motion analysis with the help of camera videos | Health care | Body movement analysis | Proposed a method for analyzing human body movements using video data captured by smartphone-like cameras. |
| **7** | *Demidovskij*, *et al. [7]* 2021 | OpenVINO development toolkit documentation and workbench: Towards analysis give platform for NN inference engine optimization | Non-health care | Training model for the videos | Provided insights into the workings of the OpenVINO development toolkit for processing video inputs. |
| **8** | *Hassouneh, et al. [8]* 2020 | Construction of a live working emotion recog. system using expressions, facial images and EEG | Health care | Emotion Recognition | Presented a technique for constructing a live emotion recognition system using Virtual Markers (VMs), achieving high accuracy rates for facial scans and EEG signals. Having accuracy of 99.81% of facial scans  and 87.25% EEG signals.  \*VMs-Virtual Markers |
| **9** | *Patel, et al. [9]* 2020 | Facial expression and sentiment analysis by using AI techniques: challenges, state-of-the-art and taxonomies, | Health care | Emotion or Sentiment analysis | Conducted a survey on emotion recognition in static images using Facial Expression Recognition (FER) techniques, highlighting the need for further research in this area. |
| **10** | *Ahmed, et al. [10]* 2019 | Emotion identification and recognition from body motion and body action | Health care | Emotion Recognition | Investigated body motion analysis for detecting various physical disabilities, emphasizing the importance of assessing movements like knee extension, hip motion, and hand gestures. |
| **11** | *Rawat and Wang [11] 2017* | Deep convolutional neural networks for image classification: A comprehensive study | Non-health care | Image Classification | Reviewed advancements in Deep Convolutional Neural Networks (DCNNs), evaluating their efficiency, accuracy, and robustness, while addressing associated challenges. |
| **12** | *Awais, et al. [12]* 2020 | LSTM-dependent emotion recognition using the physiological signals for healthcare and distance learning in | Health care | Emotion Recognition | Proposed an LSTM-based Deep Learning (DL) model for emotion recognition within an IoT framework, achieving an overall accuracy of 95%. |
| **13** | *Radaideh, et al. [13]* 2020 | An approach to predict the sentimental analysis by general algorithms & RNN algorithm | Health care | Real time emotion recognition | Conducted a detailed study on sentimental analysis techniques and algorithms, demonstrating superior performance and efficiency compared to previous models. |
| **14** | *Bhavitha, et al. [14]* 2017 | Comprehensive study of the ML techniques in sentiment analysis | Non-health care | Machine learning models | Explored support vector machine (SVM) as a potential technique for emotional recognition within the context of sentimental analysis studies. |

**CHAPTER 3**

**METHODOLOGY**

* 1. **PRODUCT PERSPECTIVE**

The foundation of the product lies in its action detection model, leveraging the OpenPose pose estimation algorithm. This sophisticated algorithm plays a pivotal role in identifying and analyzing the pose of individuals captured within the camera's field of view. By meticulously tracking key body points and articulations in real-time, OpenPose enables precise determination of the actions being performed by the individual at any given moment.

At its core, the product harnesses the power of computer vision and deep learning to interpret the detected poses and infer the corresponding actions. This process involves a series of intricate computations and pattern recognition tasks, allowing the system to discern a diverse range of actions, gestures, and movements exhibited by the subjects in view.

Through continuous analysis of the pose data captured by the camera feed, the system dynamically assesses the ongoing actions of the individuals being monitored. Each detected action is meticulously categorized and evaluated against a predefined set of action labels, enabling the system to provide accurate and timely outputs based on the observed activities.

Moreover, the product's functionality extends beyond mere action detection, incorporating sophisticated decision-making algorithms to interpret the significance and context of detected actions. By contextualizing the observed behaviors within relevant scenarios or environments, the system enhances its ability to deliver meaningful and actionable insights to users.

In addition to real-time action detection, the product offers advanced features such as action classification, temporal analysis, and anomaly detection. These capabilities enable the system to identify patterns, trends, and irregularities in human behavior, empowering users to make informed decisions and take proactive measures in various contexts.

Furthermore, the product's versatility and scalability make it suitable for a wide range of applications across diverse industries and domains. From security surveillance and crowd monitoring to sports analysis and human-computer interaction, the possibilities are virtually limitless. By providing a robust and adaptable solution for action detection and analysis, the product aims to revolutionize the way we perceive and interact with the world around us.

In summary, the product represents a groundbreaking advancement in the field of computer vision and artificial intelligence, offering unparalleled capabilities in action detection and analysis. By harnessing the cutting-edge capabilities of the OpenPose pose estimation algorithm, the system delivers actionable insights and enhances situational awareness in a variety of contexts, ultimately empowering users to make more informed decisions and optimize their workflows.

* + 1. **HARDWARE INTERFAES**

In this project we had built two types of models in which one is completely made from scratch, trained on our own dataset, used mediapipe for action detection which an accuracy of about 88.87%.

* System Configuration:
* Windows 11 64-bit
* 16GB ram
* Nvidia Rtx 3050 6gb GPU
* i5-13450 HX
* Tensorflow=2.10 (integrated GPU using CUDA toolkit version 11.2)

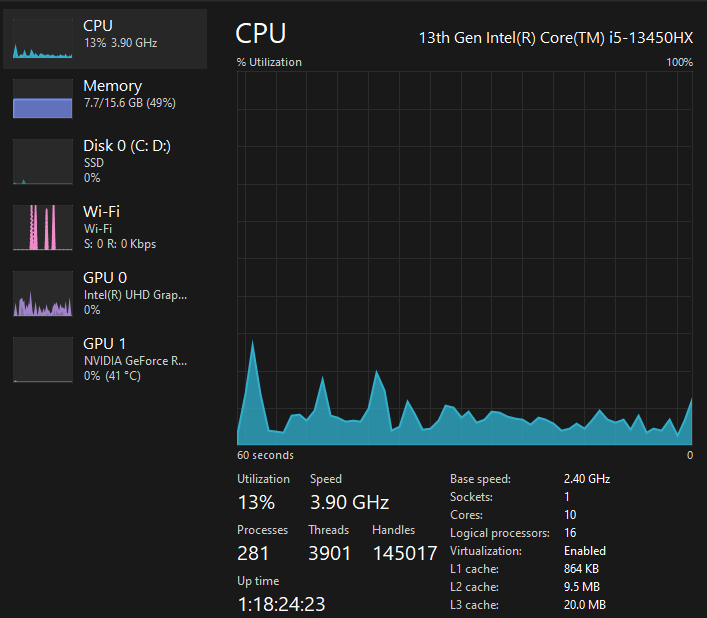


Figure 5: system configurations

Figure 5 showing the system configuration used to run the project. Same configurations for the Linux in which the project get implemented

The second model is based on the pretrained Openpose pose estimation model that we trained on our own dataset with an accuracy of about 95.67%, which was trained on about 10000 frames.

* System Configuration:
* Ubuntu Linux x84-64bit
* 16GB ram
* Nvidia Rtx 3050 6gb GPU
* i5-13450 HX
* Tensorflow-gpu=1.13.1

Now the question arises is why **Linux**?

* **Flexibility and Customization:** Linux provides a high level of flexibility and customization, allowing developers to tailor the operating system environment to their specific requirements. This flexibility is crucial for DNN development, where users may need to install and configure various software libraries, tools, and frameworks to build and train complex neural network models.
* **Compatibility with Open-Source Tools:** Many deep learning frameworks, libraries, and tools are primarily developed and optimized for Linux-based environments. These include popular frameworks like TensorFlow, PyTorch, and Apache MXNet, as well as supporting libraries such as CUDA and cuDNN for GPU acceleration. Using Linux ensures seamless compatibility and integration with these open-source tools, enabling users to leverage the latest advancements in deep learning research and development.
* **Performance and Scalability**: Linux is renowned for its performance and scalability, making it well-suited for running computationally intensive tasks such as training deep neural networks. Linux-based systems can efficiently utilize multi-core processors, distributed computing clusters, and GPU accelerators to accelerate model training and inference. Additionally, Linux offers robust support for containerization technologies like Docker and Kubernetes, facilitating the deployment and scaling of DNN models in cloud and edge computing environments.
* **Community Support:** Linux benefits from a vast and active community of developers, researchers, and enthusiasts who contribute to its ongoing development and maintenance. This vibrant ecosystem provides extensive documentation, tutorials, forums, and repositories of software packages, making it easier for users to troubleshoot issues, seek assistance, and collaborate on deep learning projects.
* **Security and Reliability:** Linux is renowned for its robust security features and reliability, making it a trusted choice for running mission-critical workloads, including deep learning tasks. Linux-based systems offer granular control over system resources, user permissions, and network access, helping to protect sensitive data and ensure the integrity of DNN models and applications.

Overall, Linux's flexibility, compatibility, performance, scalability, community support, and security features make it the preferred choice for developing and deploying deep neural network models in a wide range of applications and industries.

We had tested out the project on multiple models including MobileNet.

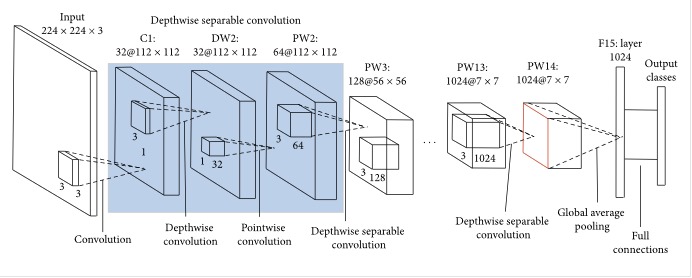


Figure 6: MobileNet Architecture

Figure 6 [15] MobileNet is a family of convolutional neural network architectures tailored for efficient inference on mobile and embedded devices. Developed by Google, MobileNet prioritizes computational efficiency while maintaining competitive performance in various computer vision tasks. Key features include depth wise separable convolutions to reduce parameters, scalability for customization, and pretrained models for transfer learning. MobileNet's lightweight design makes it ideal for on-device inference in mobile apps, IoT devices, and other embedded systems, enabling real-time performance in diverse applications such as image classification, object detection, and semantic segmentation. But the highest accuracy we got is on OpenPose algorithm

OpenPose is a widely-used computer vision library designed for real-time multi-person key point detection and pose estimation. Developed by Carnegie Mellon University, OpenPose offers robust and efficient algorithms for accurately identifying human body key points, such as joints and body parts, from images and videos.

Key features of OpenPose include:

* **Multi-Person Pose Estimation**: OpenPose can detect multiple individuals within an image or video frame simultaneously, enabling the estimation of body poses for each person present in the scene.
* **Real-Time Performance:** OpenPose is optimized for real-time performance, making it suitable for applications requiring low-latency processing, such as gesture recognition, action recognition, and human-computer interaction.
* **High Accuracy**: OpenPose achieves high accuracy in keypoint detection and pose estimation, even in challenging conditions such as occlusions, varying body orientations, and cluttered backgrounds.
* **Modularity and Flexibility:** OpenPose is modular and extensible, allowing developers to integrate it into their existing projects or customize it to suit specific requirements. The library provides APIs for accessing keypoint coordinates, rendering skeletons, and performing further analysis or post-processing.
* **Cross-Platform Support**: OpenPose is compatible with various operating systems, including Windows, Linux, and macOS, making it accessible to a wide range of developers and researchers. It also supports multiple programming languages such as C++, Python, and MATLAB.
* **Pre**-**Trained** **Models**: OpenPose offers pre-trained models trained on large-scale datasets, enabling developers to leverage state-of-the-art pose estimation capabilities without the need for extensive training data or computational resources.
  + 1. **SOFTWARE INTERFACES**

**CUDA Toolkit:**

The CUDA Toolkit is a comprehensive software development platform provided by NVIDIA for accelerating GPU-accelerated computing applications. It includes a suite of tools, libraries, and APIs designed to harness the computational power of NVIDIA GPUs for parallel processing tasks. Key components of the CUDA Toolkit include:

* **CUDA Runtime API**: Provides a set of functions for managing GPU resources, launching parallel kernels, and transferring data between the CPU and GPU.
* **CUDA Driver**: The device driver required to enable CUDA functionality on NVIDIA GPUs, allowing communication between the operating system and CUDA-enabled applications.
* **cuBLAS**: A GPU-accelerated implementation of the Basic Linear Algebra Subprograms (BLAS) library for performing matrix and vector operations efficiently on NVIDIA GPUs.
* **cuDNN**: The CUDA Deep Neural Network library provides optimized primitives and algorithms for deep learning applications, enabling faster training and inference on NVIDIA GPUs.
* **cuFFT**: A GPU-accelerated Fast Fourier Transform (FFT) library for computing FFTs efficiently on NVIDIA GPUs, essential for signal processing and scientific computing tasks.

Overall, the CUDA Toolkit serves as a powerful platform for developing and deploying high-performance GPU-accelerated applications across various domains, including scientific computing, machine learning, computer vision, and more.

**TensorFlow:**

TensorFlow is an open-source machine learning framework developed by Google for building and training machine learning models. It provides a comprehensive ecosystem of tools, libraries, and resources for developing and deploying artificial intelligence applications. Key features of TensorFlow include:

* **Graph-based Computation:** TensorFlow models are represented as computational graphs, where nodes represent mathematical operations and edges represent data flow between operations. This enables efficient parallel execution and optimization of machine learning workflows.
* **Flexible Architecture**: TensorFlow offers a flexible and modular architecture that supports a wide range of machine learning tasks, including deep learning, reinforcement learning, and probabilistic modeling. It provides high-level APIs for building and training models, as well as lower-level APIs for fine-grained control and customization.
* **Scalability**: TensorFlow supports distributed training across multiple devices and machines, allowing users to scale their machine learning workloads to large datasets and compute clusters. It also provides tools for deploying models to production environments, including TensorFlow Serving for serving models over HTTP endpoints.
* **Integration**: TensorFlow seamlessly integrates with other popular machine learning frameworks and libraries, such as Keras, for building and training deep learning models. It also supports interoperability with programming languages like Python and C++.

Overall, TensorFlow is a powerful and versatile framework for building and deploying machine learning models, with applications ranging from natural language processing and computer vision to healthcare and finance.

**TensorFlow GPU:**

TensorFlow GPU is a version of TensorFlow optimized for GPU-accelerated computing. It leverages the computational power of NVIDIA GPUs to accelerate training and inference of machine learning models, enabling faster execution and higher throughput compared to CPU-only implementations. TensorFlow GPU utilizes CUDA and cuDNN libraries for GPU-accelerated computations, allowing users to harness the parallel processing capabilities of NVIDIA GPUs for training large-scale deep learning models efficiently.

By leveraging TensorFlow GPU, developers can significantly reduce training times and improve model performance for a wide range of machine learning tasks, including image classification, object detection, and natural language processing. TensorFlow GPU is compatible with a variety of NVIDIA GPU architectures, from entry-level GPUs to high-end data center accelerators, providing scalability and flexibility for different computing environments and use cases.

**OpenCV:**

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library designed to provide a common infrastructure for building computer vision applications. It offers a wide range of functionalities for image and video processing, including feature detection, object tracking, camera calibration, and machine learning algorithms.

Key features of OpenCV include:

* **Image Processing**: OpenCV provides a comprehensive set of tools and algorithms for performing various image processing tasks, such as filtering, edge detection, image transformation, and morphological operations.
* **Feature Detection and Description**: OpenCV includes algorithms for detecting and describing key features in images, such as corners, blobs, and key points, which are essential for tasks like object recognition, tracking, and matching.
* **Object Detection and Recognition**: OpenCV offers pre-trained models and algorithms for detecting and recognizing objects in images and videos using techniques such as Haar cascades, HOG (Histogram of Oriented Gradients), and deep learning-based approaches.
* **Camera Calibration**: OpenCV provides tools for calibrating camera parameters and estimating camera

**VScode:**

Visual Studio Code (VS Code) is a lightweight, open-source code editor developed by Microsoft. It is renowned for its versatility, performance, and extensive ecosystem of extensions, making it a popular choice among developers for various programming languages and platforms. Here's a detailed description of its features and functionalities:

* **Cross-Platform Support**: VS Code is compatible with major operating systems, including Windows, macOS, and Linux, ensuring a consistent development experience across different platforms. This cross-platform support allows developers to seamlessly switch between environments without compromising productivity.
* **Intuitive User Interface**: VS Code features a clean and intuitive user interface designed to maximize productivity and streamline the coding experience. It offers customizable layouts, themes, and keyboard shortcuts, enabling users to tailor the editor to their preferences and workflow.
* **Integrated Terminal**: VS Code includes a built-in terminal emulator that allows developers to run command-line tasks, execute scripts, and interact with their development environment without leaving the editor. This integrated terminal enhances workflow efficiency by eliminating the need to switch between multiple applications.
* **Version Control Integration**: VS Code seamlessly integrates with version control systems such as Git, providing essential features for managing source code repositories directly within the editor. It offers visual diffs, commit history, branching, and merging capabilities, empowering developers to collaborate effectively and track changes in their projects.
* **Extensibility**: One of the standout features of VS Code is its extensive ecosystem of extensions, which enable users to customize and extend the editor's functionality to suit their specific needs. These extensions cover a wide range of categories, including language support, debugging, code linters, and productivity tools, allowing users to tailor their development environment to their workflow.
* **Task Automation:** VS Code allows developers to define and execute tasks directly within the editor using task runners such as Gulp, Grunt, or npm scripts. This built-in task automation feature simplifies common development tasks such as building, testing, and deploying applications, enhancing productivity and reducing manual effort.

In summary, Visual Studio Code is a versatile and feature-rich code editor that provides developers with a powerful and customizable development environment. With its intuitive interface, extensive ecosystem of extensions, and built-in tools for editing, debugging, and collaboration, VS Code empowers developers to write, debug, and maintain code with efficiency and ease.

* + 1. **REQUIREMENTS**

The fundamental prerequisites for the project include the following components:

* **Python Environment**: Python version 3.6 serves as the foundation due to its compatibility with TensorFlow-GPU version 1.13.1, a critical requirement for our project. This version of Python is chosen based on its proven compatibility and optimal performance with TensorFlow, ensuring smooth execution of deep learning tasks.
* **Computational Unit**: A robust computational unit, whether a CPU or GPU, is essential for efficient processing of machine learning algorithms. Preferably, hardware from the NVIDIA RTX series paired with an HX series CPU offers superior performance and compatibility with deep learning frameworks.
* **Development Environment**: Setting up a dedicated development environment is crucial for managing dependencies and ensuring project reproducibility. Utilizing Conda environments, particularly configured with TensorFlow-GPU version 1.13.1, provides a controlled and isolated environment conducive to development and experimentation.
* **Version Control with Git**: Git serves as the cornerstone for version control, facilitating collaboration, code management, and project tracking. Utilizing Git allows for seamless cloning of data and repositories, enabling efficient project management and team collaboration throughout the development lifecycle.
* **Essential Libraries and Frameworks:** Incorporating essential libraries and frameworks such as TensorFlow, PyTorch, OpenCV, NumPy, and SciPy is imperative for comprehensive support in video data processing, feature extraction, model training, and inference tasks. These libraries offer a wide array of tools and algorithms essential for implementing machine learning solutions effectively.

By adhering to these foundational requirements, we establish a robust groundwork for the project, ensuring compatibility, performance, and scalability throughout the development process. Additionally, maintaining a balance between hardware specifications, development environment configuration, and library integration optimizes productivity and facilitates seamless execution of machine learning workflows.

* 1. **PRODUCT FUNCTIONS**

The action recognition project aims to fulfill several key functions essential for its successful operation and utility. These functions include:

* **Real**-**time** **Action** **Detection**: The primary function of the project is to detect human actions in real-time video streams captured by surveillance cameras. Leveraging advanced computer vision algorithms and deep learning models, the system analyzes video frames to identify and classify various human actions accurately and efficiently.
* **Pose** **Estimation**: Integral to action recognition is pose estimation, which involves detecting and tracking the skeletal key points of individuals in video frames. By accurately estimating the poses of human subjects, the system can extract relevant spatial and temporal features necessary for action classification and analysis.
* **Action** **Classification**: Once poses are estimated, the system classifies the observed actions into predefined categories or classes. Using machine learning techniques such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), the system assigns probabilities to different action classes, enabling the identification of specific activities or behaviors exhibited by individuals.
* **Alert Generation**: In addition to action classification, the system generates alerts or notifications in real-time when certain predefined actions or events of interest are detected. These alerts may trigger immediate responses or interventions, such as notifying security personnel or activating surveillance measures, to address potential security threats or safety concerns.
  1. **CONSTRAINTS**

Despite its capabilities, the action recognition project operates within certain constraints that may impact its performance or functionality. These constraints include:

* **Hardware** **Limitations**: The effectiveness of the system may be limited by the hardware resources available, including the processing power of the CPU or GPU, memory constraints, and input/output bandwidth. Insufficient hardware resources can affect the system's ability to process video streams in real-time or handle large-scale deployments.
* **Environmental** **Factors**: Environmental conditions such as lighting conditions, occlusions, and background clutter can pose challenges to accurate action recognition. Variations in illumination, shadows, and object occlusions may affect the system's ability to detect and classify actions reliably, particularly in complex or dynamic environments.
* **Training** **Data** **Availability**: The performance of the action recognition models relies heavily on the availability and quality of training data. Limited or biased training data may result in suboptimal model performance, leading to errors or inaccuracies in action detection and classification.
* **Model** **Robustness** **and** **Generalization**: While trained models may exhibit high accuracy on the training data, their robustness and generalization capabilities in real-world scenarios may be limited. Adverse conditions, variations in appearance or context, and unseen actions or scenarios may challenge the system's ability to generalize effectively beyond the training data.
  1. **ASSUMPTIONS AND DEPENDENCIES**

The action recognition project operates under certain assumptions and dependencies that influence its design, implementation, and usage. These assumptions and dependencies include:

* **Assumption of Homogeneous Environments**: The project assumes relatively homogeneous environments with consistent lighting, camera viewpoints, and background settings. Deviations from these assumptions may require additional preprocessing or calibration steps to ensure accurate action recognition.
* **Dependency on External Libraries and Frameworks**: The project relies on external libraries and frameworks such as TensorFlow, OpenCV, and NumPy for core functionality. Changes or updates to these dependencies may impact the project's compatibility, performance, or behavior, necessitating periodic maintenance and version management.
* **Assumption** **of** **Stationary** **Cameras**: The project assumes stationary surveillance cameras with fixed viewpoints and perspectives. Moving or unstable cameras may introduce motion blur, perspective distortions, or occlusions that complicate action recognition and pose estimation tasks.
* **Dependence** **on** **Training** **Data** **Quality**: The effectiveness of the action recognition models is contingent upon the quality, diversity, and representativeness of the training data. High-quality, well-annotated training datasets are essential for training accurate and robust models capable of generalizing to unseen scenarios and variations.

By acknowledging and addressing these constraints, assumptions, and dependencies, the action recognition project can be designed, implemented, and deployed effectively, maximizing its utility and performance in real-world applications.

* 1. **FLOW OF THE IMPLEMENTATION**

There are 5 files which we make for this project

src/s1\_get\_skeletons\_from\_training\_imgs.py

src/s2\_put\_skeleton\_txts\_to\_a\_single\_txt.py

src/s3\_preprocess\_features.py

src/s4\_train.py

src/s5\_test.py

Firstly, the videos get converted into the frames as on frames that would be easier to run any training algorithm precisely. More than 16000 frames got extracted.   
Each frame is named as image\_filename\_format: "{:05d}.jpg".

Using the naming convention of the frames a separate txt file is created in which all the valid frames (between these frames the action is performed) are numbered in the format.

cough\_1

52 59

72 79

Sneez\_2

54 62

In which the first number represents the starting index and the end one is representing the ending index.

**s1\_get\_skeletons\_from\_training\_imgs.py** this file is used to extract the skeletons from all the valid frames using OpenPose. After this all the skeleton files are converted to the text and stored in the single txt file using this code file **s2\_put\_skeleton\_txts\_to\_a\_single\_txt.py**. After this the data is pre-processed and training is performed on the processed data. The file **s5\_test.py** is used to run the project.

**CHAPTER 4**

**SYSTEM DESIGN AND IMPLEMENTATION**

* 1. **ARCHITCTURE DIAGRAM**

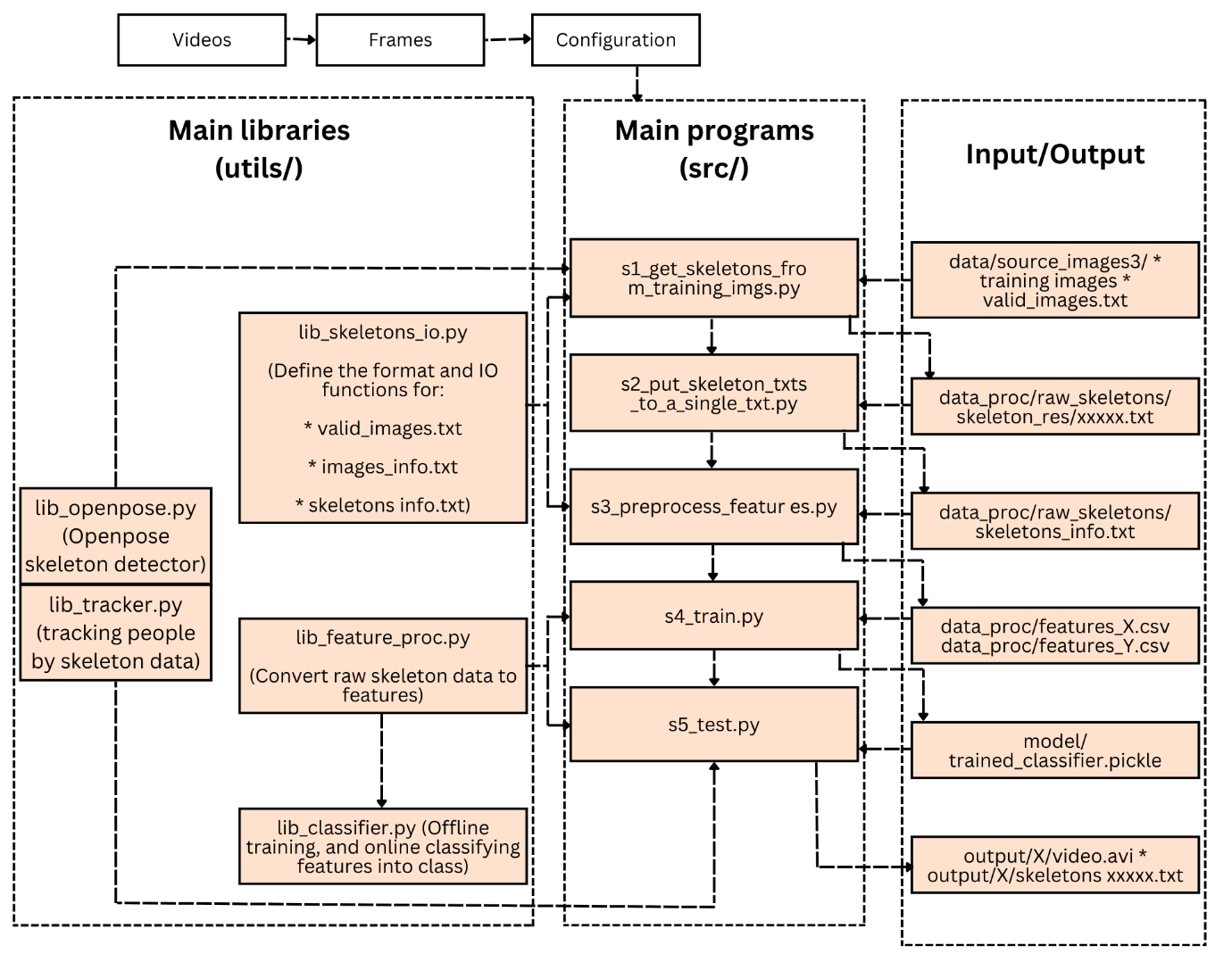
****

Figure 7: The Architecture of the project

The architecture showed in Figure 7 depicting the complete flow of the project, how It is working and from where it is obtaining all the data.

* 1. **DATASET USED**

There are two distinct datasets utilized for training two different models. The initial dataset, sourced from GitHub, comprises video samples encompassing three actions: coughing, sneezing, and expressing tiredness. Each action category comprises 120 videos, leading to a total frame count exceeding 100,000 after frame extraction. However, a notable challenge with this dataset was its inherent quality issues, characterized by excessive blurriness and low frame-per-second (FPS) rates.

Despite preprocessing efforts using CMU OpenPose for feature extraction, the resultant dataset exhibited suboptimal performance during model training. Specifically, the accuracy achieved on this dataset peaked at 72.69%, falling below the desired threshold for application within the healthcare domain. The training progression and performance metrics for this dataset are visually depicted in Figure 8 of the accompanying documentation.

Recognizing the limitations imposed by the dataset's quality and characteristics, further exploration and potential enhancements are warranted to improve model efficacy and real-world applicability, particularly within critical healthcare contexts. Additional strategies may include data augmentation techniques, refinement of preprocessing methodologies, or exploration of alternative feature extraction approaches to mitigate the impact of low-quality input data on model performance.

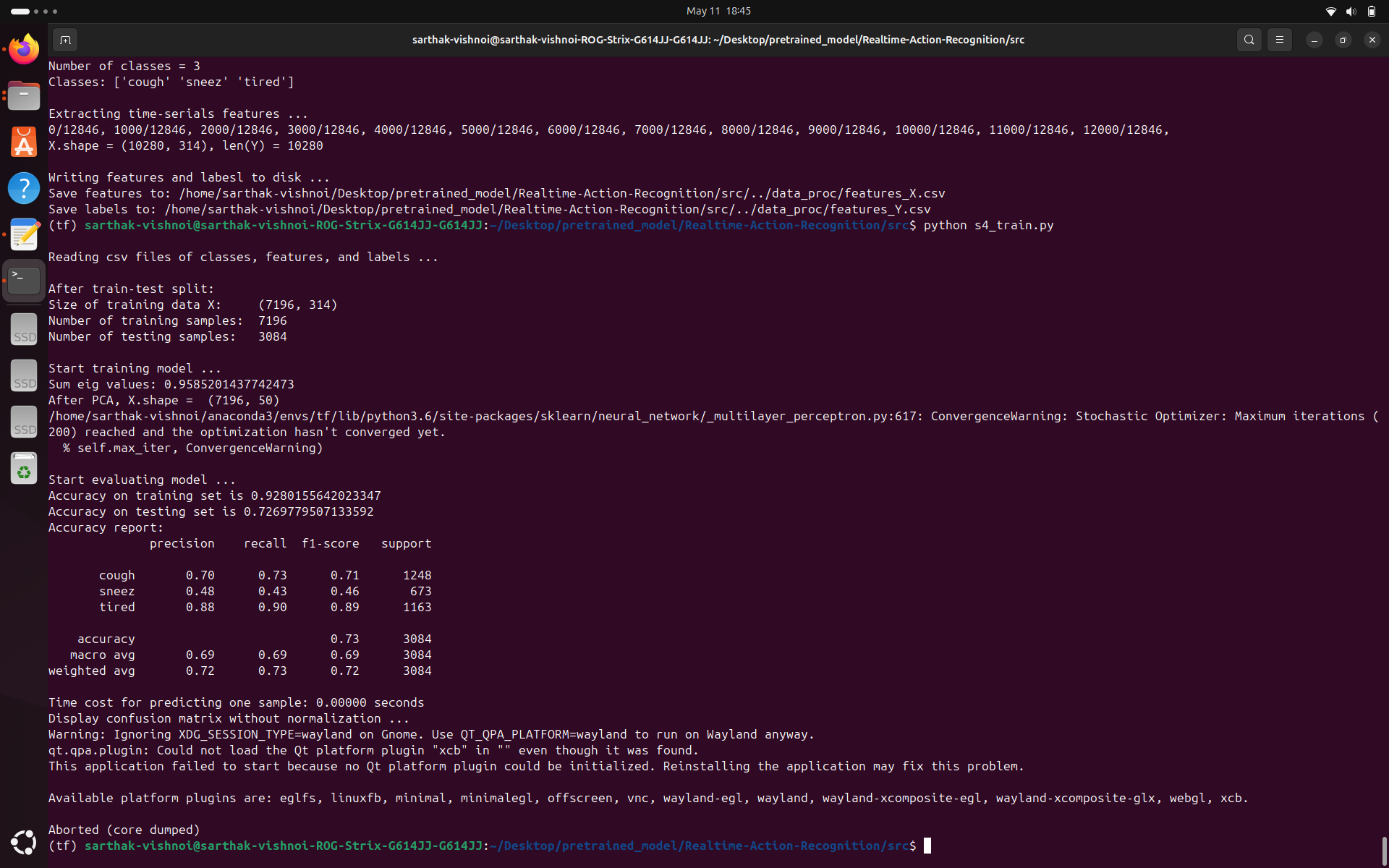


Figure 8: Training Data of Old Dataset

The second dataset utilized for model training is an internally recorded dataset, distinguished by its superior quality compared to the GitHub-sourced dataset. Recorded at a resolution of 1080p and a smooth frame rate of 30 frames per second (FPS), this dataset offers enhanced clarity and temporal fidelity, crucial for robust model learning.

Comprising four distinct classes—coughing, sneezing, expressing tiredness, and a baseline "normal" state—this dataset provides a comprehensive representation of common human actions and states relevant to healthcare monitoring and analysis.

Following the training regimen on this refined dataset, significant improvements in model accuracy were observed. Compared to the initial dataset, the accuracy metrics experienced a notable increase, reflecting the efficacy of leveraging high-quality input data for model learning.

The enhanced accuracy attained on this dataset underscores the importance of data quality and its pivotal role in influencing model performance, particularly in critical domains such as healthcare. Moving forward, continued exploration of methodologies for dataset curation, acquisition, and preprocessing will remain essential to further refine model performance and facilitate its deployment in real-world healthcare scenarios.

The converted dataset into the skeleton frames is shown below in Figure 9.

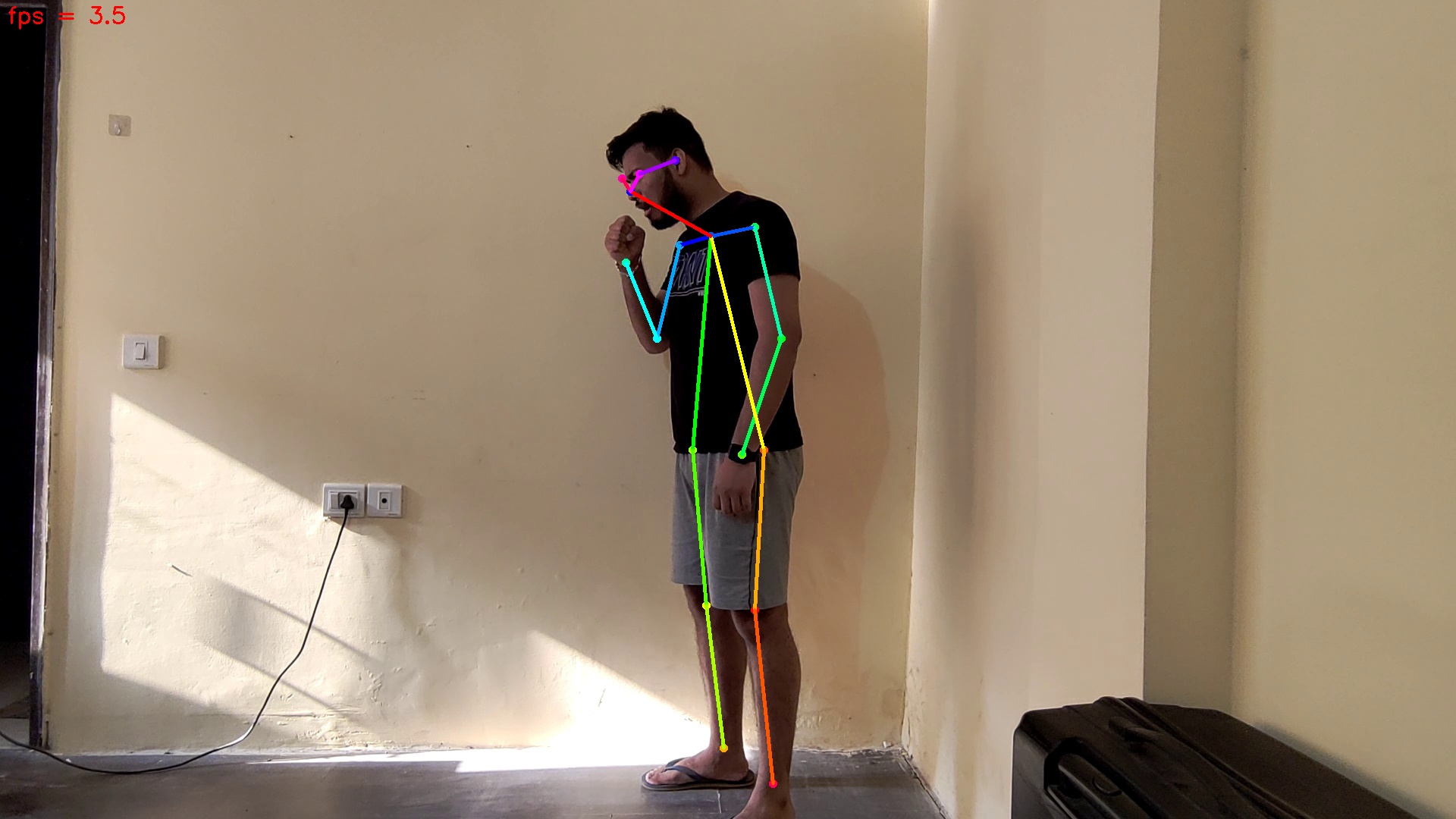


Figure 9: The skeleton using OpenPose

sample skeleton txt file made in the data\_proc folder to predict the action.

[[1, 5, 142, "cough", "cough\_1/00241.jpg", 0.5213414634146342, 0.12839673913043478, 0.5640243902439024, 0.1589673913043478, 0.5304878048780488, 0.16202445652173914, 0.5121951219512195, 0.23233695652173914, 0.5182926829268293, 0.17730978260869565, 0.600609756097561, 0.14979619565217392, 0.6280487804878049, 0.24150815217391305, 0.6067073170731707, 0.3179347826086956, 0.5426829268292683, 0.3026494565217391, 0.5579268292682927, 0.4127038043478261, 0.5701219512195121, 0.5227581521739131, 0.5914634146341463, 0.3118206521739131, 0.5853658536585366, 0.4188179347826087, 0.5914634146341463, 0.5380434782608696, 0.5182926829268293, 0.11922554347826086, 0.5304878048780488, 0.11616847826086957, 0.0, 0.0, 0.5609756097560976, 0.11005434782608696]

* 1. **WORKFLOW**

The project's workflow algorithm is designed to facilitate the automated analysis and classification of human actions or states from image or video data. This comprehensive process encompasses several key stages, each aimed at extracting meaningful features, reducing dimensionality, and ultimately classifying individuals based on their observed behaviors. Let's delve deeper into each component of this algorithm to understand its significance and impact within the context of the project.

* **Obtain Joint Positions**: This initial step involves leveraging OpenPose, a sophisticated computer vision library, to accurately detect and localize the positions of anatomical joints within the provided image or video frames. By identifying key points such as wrists, elbows, shoulders, knees, and ankles, OpenPose enables precise tracking and analysis of human movement patterns.
* **Person Tracking**: Once the joint positions are obtained, the algorithm employs a robust tracking mechanism, exemplified by the Euclidean distance-based matching algorithm encapsulated within the Tracker class. This tracking algorithm associates individuals across consecutive frames by comparing their joint positions, thereby facilitating the seamless monitoring of human activity over time.
* **Impute Missing Joints**: In scenarios where certain joint positions are not directly observable or are occluded in the current frame, the FeatureGenerator class within lib\_feature\_proc.py steps in to estimate these missing joint positions. By leveraging the relative positions of joints from the previous frame, this imputation process ensures continuity and completeness in the extracted feature set.
* **Data Augmentation**: To enhance the robustness and generalization capability of the model, controlled noise is introduced to the (x, y) coordinates of joint positions. This data augmentation strategy diversifies the training dataset, enabling the model to better handle variations in input data and improve its performance on unseen instances.
* **Feature Extraction**: Employing a sliding window approach with a duration of 0.5 seconds, features such as body velocity, normalized joint positions, and joint velocities are extracted from the tracked joint positions. This temporal aggregation strategy captures the dynamic aspects of human movement and behavior, facilitating more comprehensive feature representation.
* **Dimensionality Reduction**: With a potentially large feature space resulting from the extraction process, Principal Component Analysis (PCA) is applied to reduce the dimensionality of the feature vectors to 80 dimensions. This reduction not only aids computational efficiency but also helps mitigate the curse of dimensionality, potentially enhancing the model's generalization performance.
* **Classification**: The reduced-dimensional feature vectors are then fed into a Deep Neural Network (DNN) classifier, comprising three layers with 50 neurons each. This classifier is trained to classify individuals based on their observed behaviors, with the flexibility to seamlessly switch to alternative classifiers as per the project's requirements.
* **Temporal Smoothing**: To stabilize the predictions over consecutive frames and mitigate temporal noise, a mean filter is applied to smooth the prediction scores. This smoothing mechanism enhances the consistency and reliability of the model's predictions over time, especially in scenarios with varying input data quality or environmental conditions.
* **Labeling**: Finally, individuals exceeding a predetermined prediction score threshold of 0.8 are annotated with a label indicating the predicted class. This labeling process enables easy interpretation and visualization of the model's output, facilitating downstream analysis and decision-making tasks.

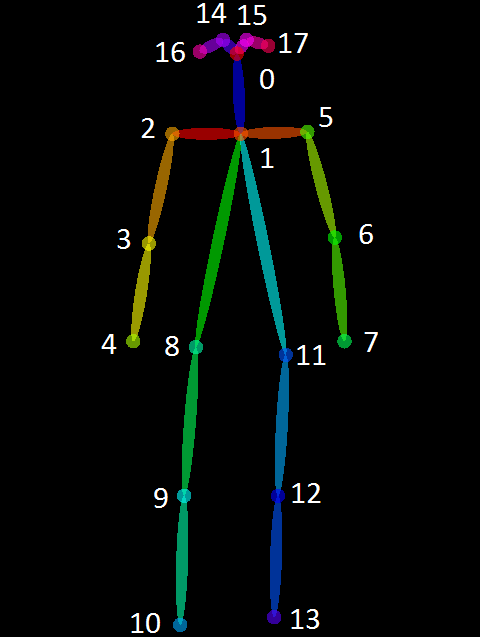


Figure 10: Order of the Joints

In summary, the project's workflow algorithm seamlessly integrates advanced computer vision, machine learning, and signal processing techniques to enable automated analysis and classification of human actions or states from image or video data. By leveraging a combination of feature extraction, dimensionality reduction, and classification methodologies, the algorithm empowers researchers and practitioners to gain deeper insights into human behavior, with potential applications spanning healthcare, security, and human-computer interaction domains. Figure 10 shows the order of the joints.

* 1. **TESTING**

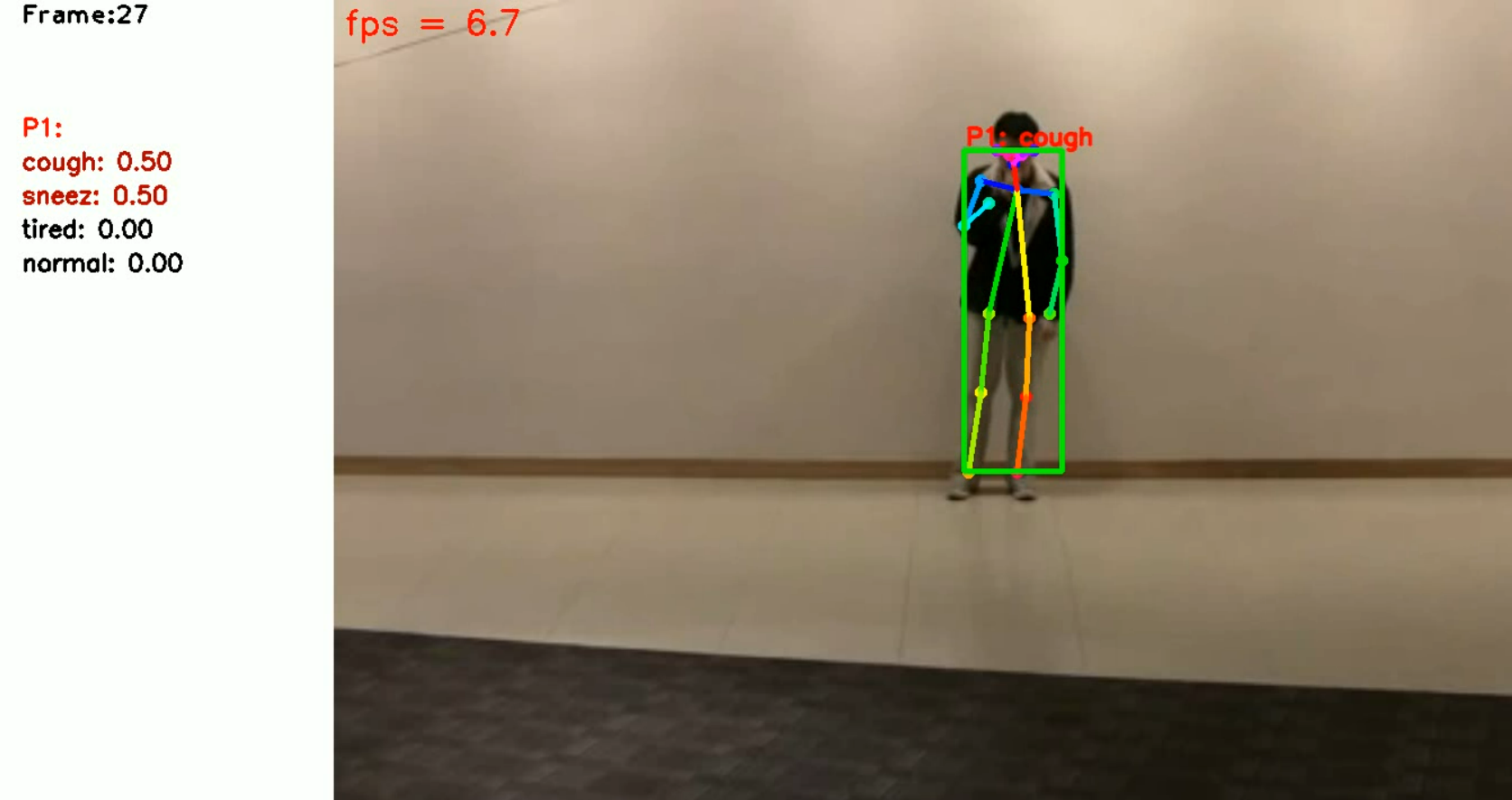
****

Figure 11: Video tested for cough

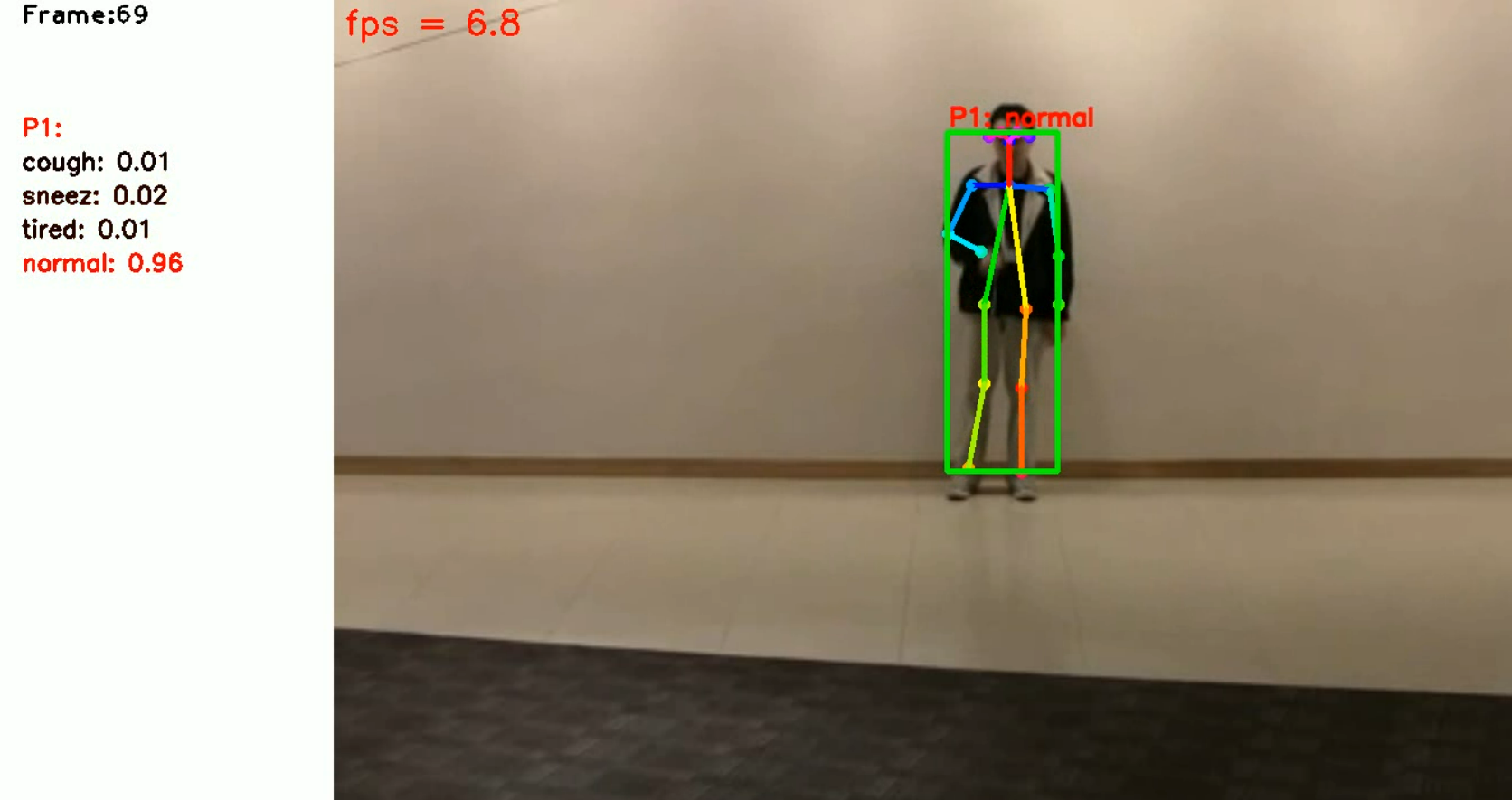
****

Figure 12: Video tested for normal postures

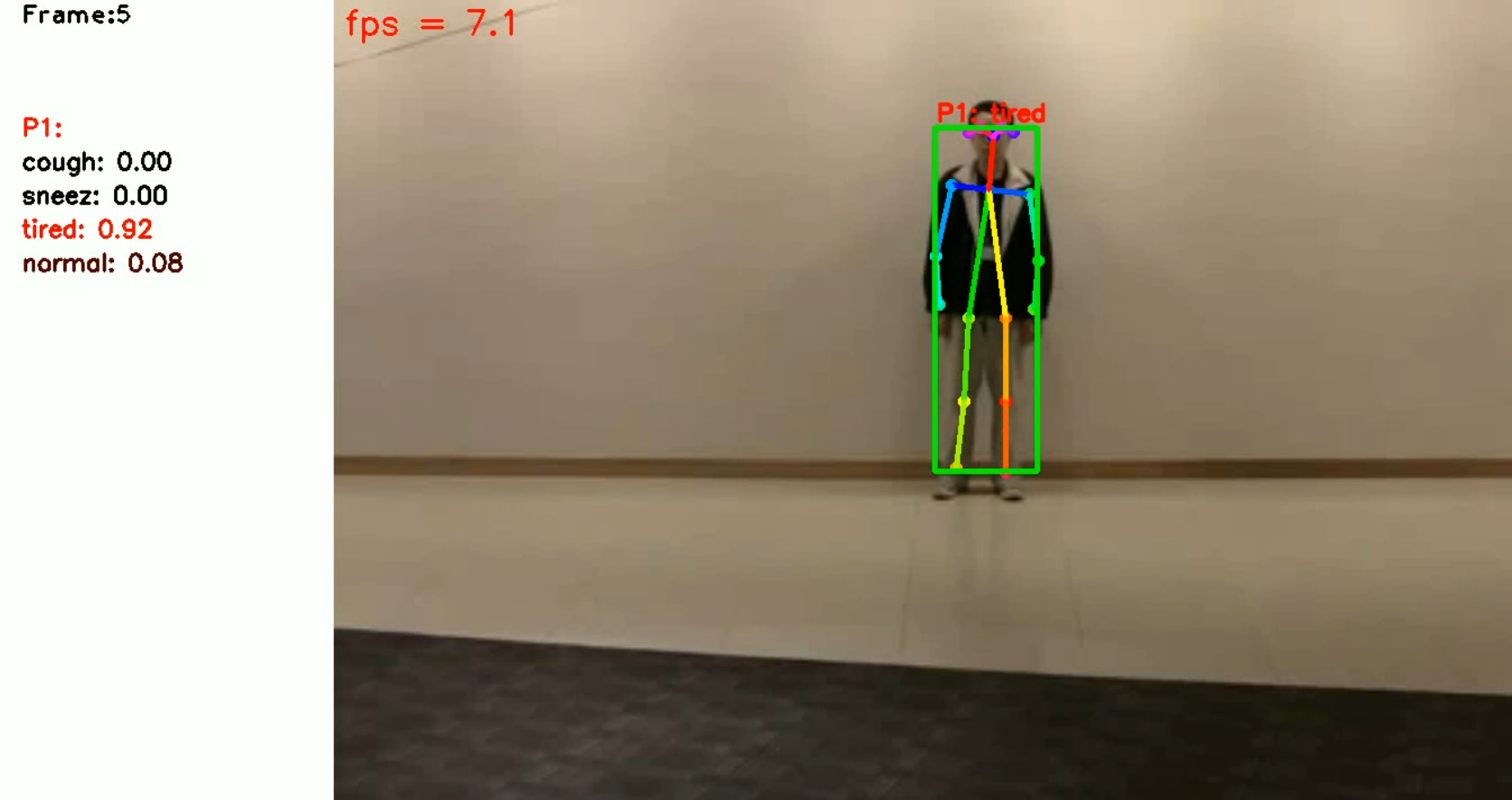
****

Figure 13: Video tested for a slightly bent posture

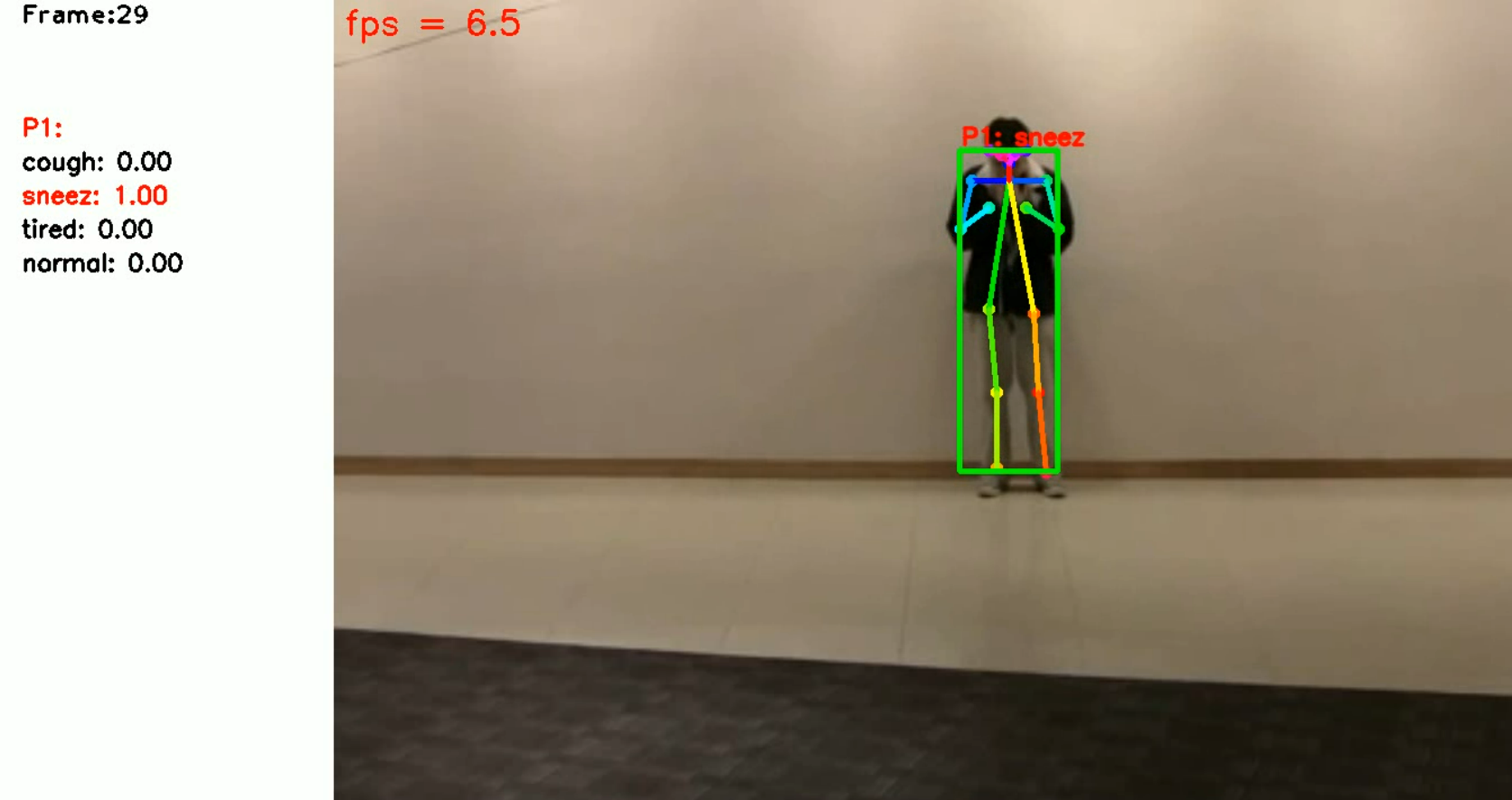
****

Figure 14: Video tested for sneeze

As we can there are four videos’ snapshots tested for each of the action the model is detecting. Figure 11, Figure 12, Figure 13, Figure 14 respectively shows the action performed by the person in the frame.

**CHAPTER 5**

**RESULT AND DISCUSSION**

* 1. **TECHNOLOGY USED**

**OpenPose Pose Estimation Library:**

OpenPose is a powerful library for real-time multi-person keypoint detection, which serves as the core technology for extracting skeletal keypoints from images or videos. It provides accurate and efficient estimation of human poses, enabling the identification of key body joints and their spatial relationships.

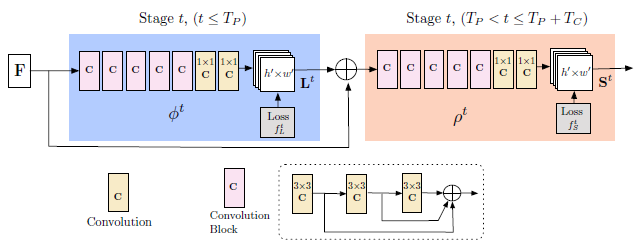


Figure 15: multilayer CNN openpose architecture

Figure 15 [18] shows the multilayer CNN architecture of the OpenPose model

The initial stages forecast the Part Affinity Fields, which subsequently refine 𝐿𝑡 derived from the base network F's feature maps.

The subsequent stages utilize the output Part Affinity Fields from the earlier layers to enhance the accuracy of the confidence map predictions.

The final (confidence maps) 𝑆 and Part (Affinity Fields) 𝐿 are then input into a greedy algorithm for further processing.

**Loss functions:**

An L2-loss function is employed to compute the discrepancy between the predicted confidence maps and Part Affinity Fields and their corresponding ground truth maps and fields.

Here, represents the ground truth Part Affinity Fields, denotes the ground truth part confidence map, and 𝑊 is a binary mask with 𝑊(𝑝)=0 when the annotation is missing at pixel 𝑝. This approach is used to avoid additional loss that might be caused by these missing annotations.

**Confidence Maps:**

The Confidence maps for each person k and each body part j is defined by:

It follows a Gaussian curve with gradual changes, where the spread of the peak is controlled by sigma. The predicted peak of the network is obtained by aggregating the individual confidence maps using a max operator.

**Python 3.6:**

Python serves as the primary programming language for developing the action detection system. Python's simplicity, versatility, and rich ecosystem of libraries make it an ideal choice for implementing machine learning and computer vision algorithms, as well as for building and deploying complex systems.

**TensorFlow GPU 1.13.1:**

TensorFlow GPU version 1.13.1 is utilized to harness the computational power of Graphics Processing Units (GPUs) for accelerated deep learning training and inference tasks. TensorFlow provides high-level APIs for building, training, and deploying deep neural networks, making it a popular choice for developing machine learning models, including those for action detection.

**NumPy and SciPy:**

NumPy and SciPy are fundamental libraries for numerical computing and scientific computing in Python. They offer a wide range of mathematical functions and data manipulation tools, which are essential for processing and analyzing the output of OpenPose, as well as for preparing data for training deep learning models.

**OpenCV (Open-Source Computer Vision Library):**

OpenCV is a versatile library for computer vision tasks, offering a wide range of functionalities for image and video processing, including reading, writing, and manipulating images and videos. It complements OpenPose by providing tools for preprocessing input data, such as resizing, cropping, and applying filters.

**Matplotlib and Seaborn:**

Matplotlib and Seaborn are visualization libraries in Python, used for creating plots, charts, and visualizations to analyze and interpret data. They facilitate the visualization of pose estimations, model predictions, evaluation metrics, and other relevant information, aiding in the debugging and validation of the action detection system.

**Scikit-learn:**

Scikit-learn is a machine learning library in Python that provides a wide range of algorithms for classification, regression, clustering, and dimensionality reduction. While TensorFlow is used for deep learning-based action detection, Scikit-learn may be employed for traditional machine learning tasks, such as feature engineering, model evaluation, and ensemble learning.

**Git and GitHub:**

Git is a version control system used for tracking changes in source code and collaborating with team members. GitHub serves as a hosting platform for Git repositories, enabling seamless collaboration, code sharing, and version control management throughout the development lifecycle of the action detection system.

* + 1. **WHY THESE TECHNOLOGIES ARE USED**

The selection of these technologies was driven by their specific capabilities and advantages, aligning closely with the requirements and objectives of the action detection system project:

OpenPose was chosen as it offers state-of-the-art multi-person keypoint detection, enabling accurate estimation of human poses in real-time. This capability forms the cornerstone of our action detection system, providing the essential input data for recognizing and analyzing various human actions.

Python 3.6 emerged as the primary programming language due to its simplicity, readability, and extensive ecosystem of libraries. These factors make Python well-suited for rapid prototyping and development of complex systems like ours. Additionally, Python's popularity ensures ample community support and readily available resources for tackling challenges throughout the project lifecycle.

TensorFlow GPU 1.13.1 was selected to leverage the computational power of GPUs for accelerated deep learning tasks. By harnessing GPU resources, TensorFlow GPU significantly reduces training times for deep neural networks, making it ideal for processing large volumes of data efficiently. This version was chosen for its stability and compatibility with other components of the system.

NumPy and SciPy played critical roles in data manipulation and scientific computing tasks. These libraries provide powerful tools for handling multidimensional arrays, performing complex mathematical operations, and implementing algorithms essential for processing OpenPose output and preparing data for model training. Their efficiency and versatility make them indispensable for our system's data processing pipeline.

OpenCV's comprehensive suite of computer vision functionalities proved invaluable for image and video processing tasks. From reading and writing images to performing advanced transformations and filtering operations, OpenCV streamlines the preprocessing of input data, enhancing the quality and consistency of features extracted from raw video frames.

Matplotlib and Seaborn were instrumental in visualizing data generated throughout the action detection pipeline. These libraries enable us to create informative plots, charts, and visualizations for analyzing pose estimations, model predictions, and evaluation metrics. By visualizing data, we gain deeper insights into the system's performance and can make informed decisions to improve its accuracy and efficiency.

Scikit-learn augmented the system's capabilities with traditional machine learning algorithms for tasks such as feature engineering and model evaluation. While TensorFlow handles deep learning tasks, Scikit-learn provides additional tools for exploring alternative approaches, fine-tuning models, and evaluating performance metrics, thus enriching the system's predictive capabilities.

Git and GitHub facilitated seamless collaboration and version control throughout the development process. By using Git for version control and GitHub for hosting repositories, we ensured code integrity, streamlined collaboration among team members, and maintained a comprehensive record of changes. This approach fosters transparency, accountability, and efficient workflow management, essential for the success of a complex project like ours.

* + 1. **HOW TO IMPROVE**
* **Model Architecture Optimization:**
  + Streamline and optimize the architecture of the action detection model to reduce computational complexity and memory usage.
  + Experiment with different network architectures, such as shallower networks or network pruning techniques, to achieve a balance between accuracy and speed.
  + Utilize efficient model components, such as depthwise separable convolutions or lightweight attention mechanisms, to reduce the number of parameters and computations.
* **Quantization and Compression:**
  + Apply quantization techniques to reduce the precision of model weights and activations, thereby reducing memory footprint and inference time without significantly compromising accuracy.
  + Explore model compression methods, such as knowledge distillation or network pruning, to remove redundant parameters and make the model more compact and efficient.
* **Hardware Acceleration**:
  + Utilize specialized hardware accelerators, such as GPUs, TPUs, or dedicated inference chips, to speed up the inference process and improve real-time performance.
  + Optimize the model for deployment on specific hardware platforms, leveraging platform-specific optimizations and libraries for maximum efficiency.
* **Data Augmentation and Preprocessing:**
  + Augment training data with diverse transformations, such as rotation, scaling, or jittering, to increase the model's robustness to variations in input data.
  + Preprocess input data efficiently, using techniques such as resizing, cropping, or normalization, to reduce computational overhead during inference while preserving essential information.
* **Temporal Consistency and Smoothing:**
  + Incorporate temporal consistency constraints into the model to ensure smooth and coherent predictions across consecutive frames.
  + Apply temporal smoothing techniques, such as averaging or filtering prediction scores over a short window of frames, to reduce noise and improve the stability of predictions over time.
* **Multi-Scale and Multi-Modal Fusion:**
  + Exploit multi-scale representations of input data to capture both global context and fine-grained details, enhancing the model's ability to detect actions at different spatial resolutions.
  + Fuse information from multiple modalities, such as RGB images, depth maps, or optical flow, to leverage complementary cues and improve the robustness of action detection in challenging conditions.
* **Incremental Learning and Adaptation:**
  + Implement techniques for incremental learning and adaptation, allowing the model to continuously improve and adapt to changes in the environment or the task requirements over time.
  + Fine-tune the model periodically on new or updated data to incorporate the latest information and maintain optimal performance in dynamic settings.
* **Efficient Post-Processing and Filtering:**
  + Develop efficient post-processing algorithms to refine and filter model predictions, removing spurious detections or false positives while retaining relevant action instances.
  + Utilize domain-specific knowledge and heuristics to guide post-processing decisions and improve the overall reliability and accuracy of action detection results.

By incorporating these strategies and techniques, it's possible to significantly enhance the performance of a real-time action detection model, making it more accurate, efficient, and suitable for deployment in various applications and environments.

* 1. **RESULTS SNAPSHOTS**

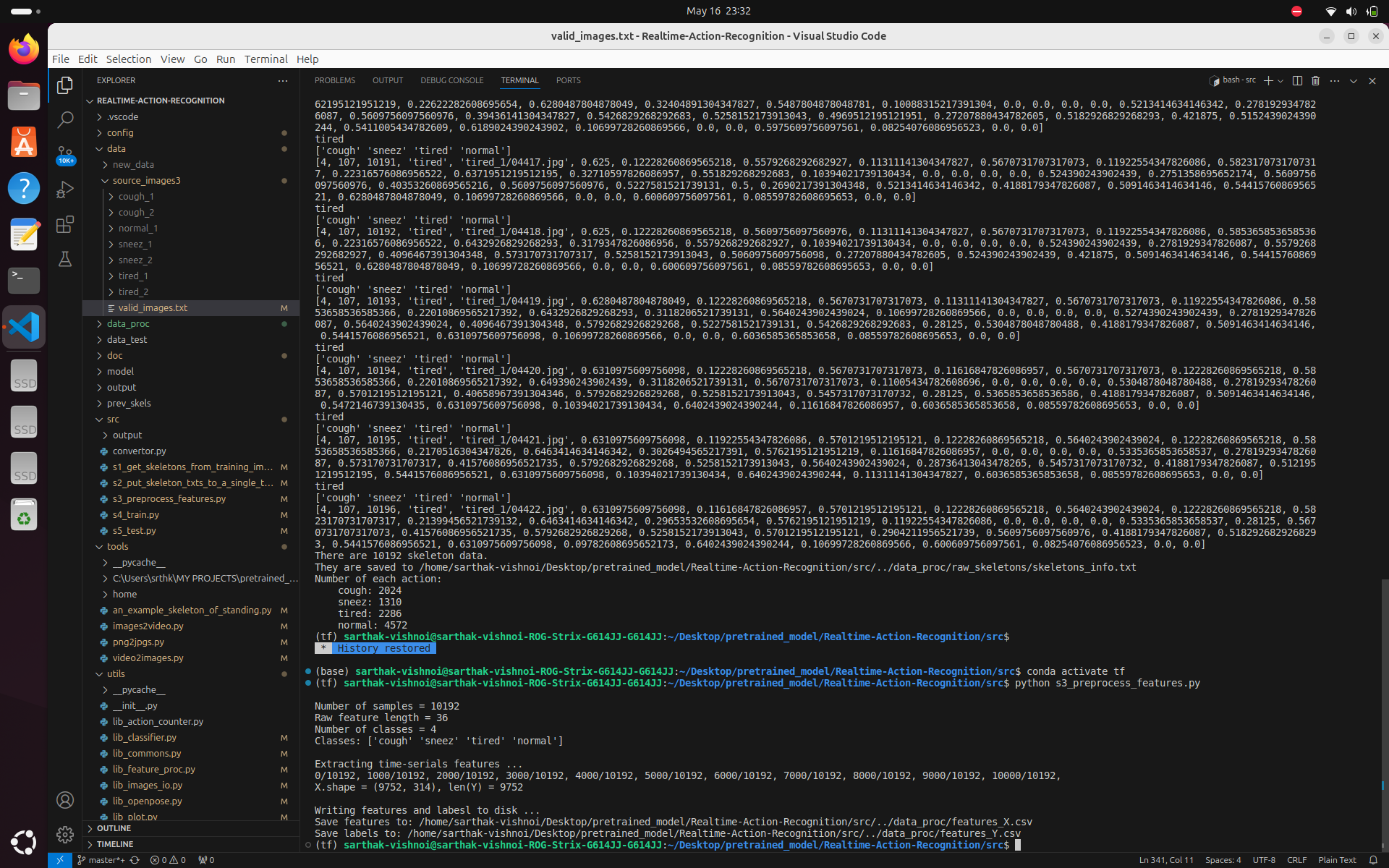
****

Figure 16: The preprocessing results

In Figure 16, we can see that the dataset consists of 10,192 samples. These samples are exclusively the frames in which the person in the video is actively performing the specified action. All other frames, where no action occurs, have been excluded. This careful selection process ensures that the dataset is comprised solely of frames that are relevant for the training process.

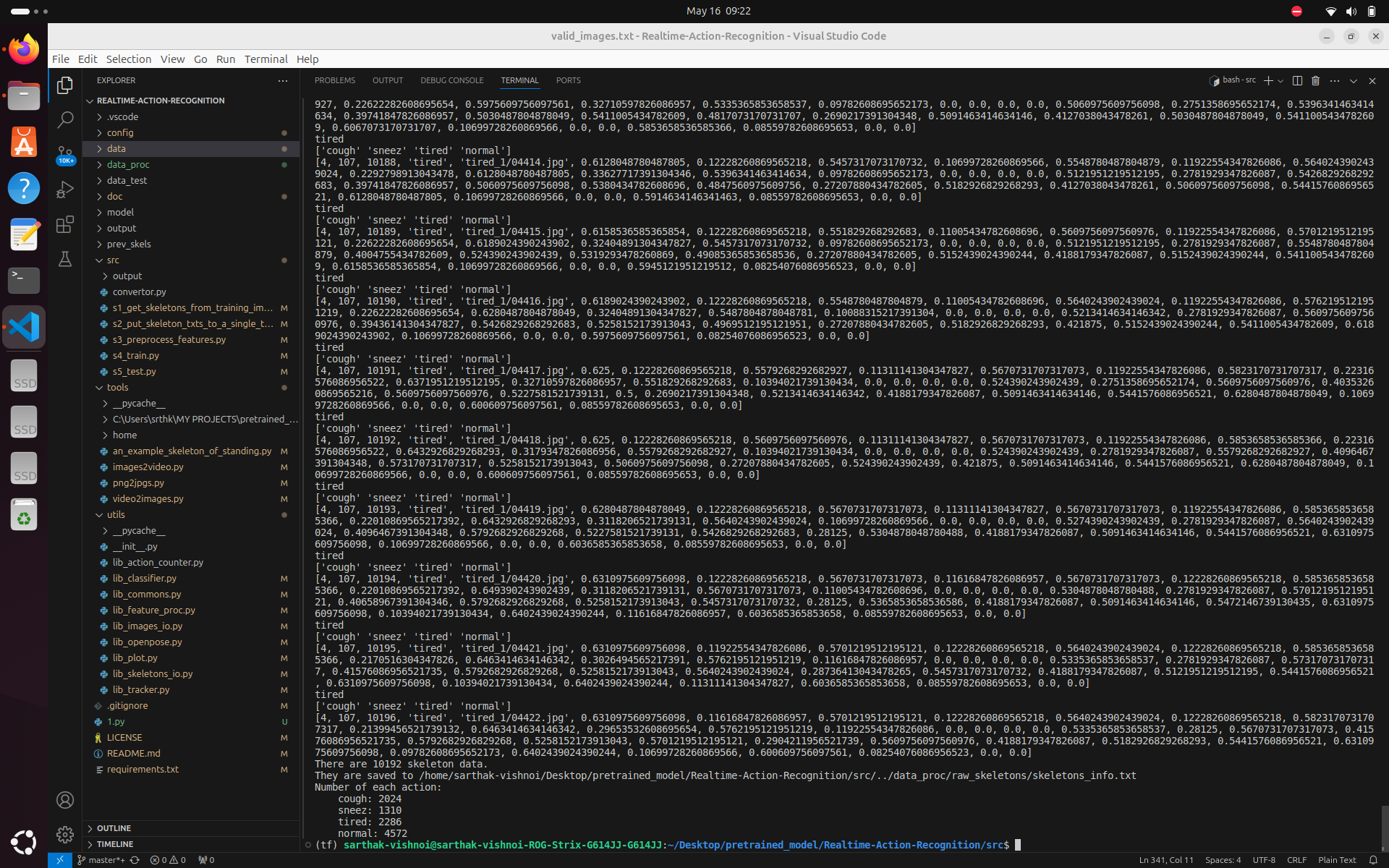
****

Figure 17: Counting frames from each class

By filtering out the non-action frames, the dataset becomes more focused and effective for training purposes. This method is applied to each action category, ensuring that only the necessary images are included. This targeted approach is crucial because it enhances the model's ability to learn from pertinent data, improving both the accuracy and efficiency of the training process. The validation process further confirms that only the required images are utilized, ensuring a high-quality dataset that is optimal for model training.

Figure 17 shoes the number of frames each class is getting for its training and Figure 18 is the final training results of the model

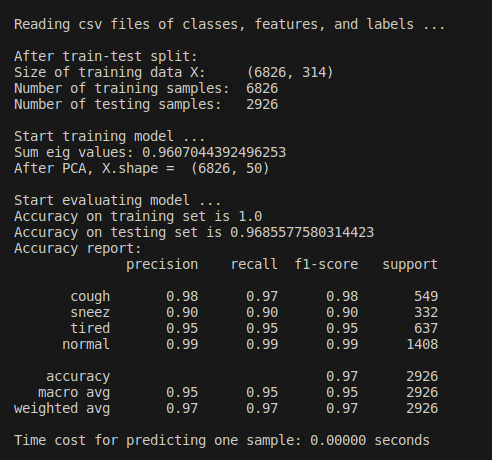
****

Figure 18: Showing the training results

* 1. **EVALUATION MATRICES**

**Precision, Recall, F1 Score, and Support**

These are common metrics used in the evaluation of classification models in machine learning and statistics.

* **Precision** is the ratio of correctly predicted positive observations to the total predicted positives. It answers the question: What proportion of positive identifications was actually correct?

Precision=True Positives (TP) / ((True Positives (TP) + False Positives (FP))

High precision indicates that an algorithm returned substantially more relevant results than irrelevant ones. Precision evaluates the exactness of the model (how many selected items are relevant).

* **Recall (Sensitivity)**

Recall is the ratio of correctly predicted positive observations to all the observations in the actual class. It answers the question: What proportion of actual positives was identified correctly?

Recall = True Positives (TP) / (True Positives (TP) + False Negatives (FN))

High recall indicates that an algorithm returned most of the relevant results. Recall evaluates the completeness of the model (how many relevant items are selected).

* **F1 Score**

The F1 Score is the weighted average of Precision and Recall. It considers both the precision and the recall of the test to compute the score. The F1 Score is especially useful when you need a balance between Precision and Recall, and when there is an uneven class distribution.

F1 Score = 2\*((Precision \* Recall)/(Precision + Recall))​

The F1 Score ranges from 0 to 1, with 1 being the best possible score indicating perfect precision and recall. F1 Score provides a single metric that balances both precision and recall.

* **Support**

Support is the number of actual occurrences of the class in the dataset. It gives context to the metric scores by indicating how many instances of each class were present in the dataset. Support helps understand the distribution of the dataset across different classes.

These metrics are crucial for understanding the performance of classification models, especially when dealing with imbalanced datasets.

Table 2: Evaluation matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
| **Cough** | **0.98** | **0.97** | **0.98** | **549** |
| **Sneeze** | **0.90** | **0.90** | **0.90** | **332** |
| **Tired** | **0.95** | **0.95** | **0.95** | **637** |
| **Normal** | **0.99** | **0.99** | **0.99** | **1408** |
|  |  |  |  |  |
| **Accuracy** |  |  | **0.97** | **2926** |
| **Macro avg** | **0.95** | **0.95** | **0.95** | **2926** |
| **Weighted Avg** | **0.97** | **0.97** | **0.97** | **2926** |

**CHAPTER 6**

**CONCLUSION AND FUTURE SCOPE**

* 1. **CONCLUSION**

It is critical to develop a real-time health prediction system that prioritizes physical health assessment, especially as new diseases continue to emerge and health problems continue to increase. The proposed system leverages various data on human physiology and behavior and aims to produce accurate health predictions with a particular focus on physical health.

The most sophisticated element of the system is its ability to read body language and data from wearable devices such as smartwatches to determine a person's physical condition. These devices can track vital signs, activity levels, sleep patterns, and other important indicators that can help determine a person's overall health.

AI-powered predictive models can also be used to personalize health assessments to suit each patient's specific characteristics and circumstances. Through continuous monitoring and analysis, the system can adjust and improve its forecasts over time, ensuring accuracy and reliability.

This technology can also be helpful in disease surveillance and population health management. Anonymized data from multiple users can be used together to identify outbreaks, identify broader health trends, and guide public health responses.

In summary, significant progress has been made in proactive health management through the development of a real-time health prediction system that focuses on physical health assessment. The system has the potential to completely transform the way people track and manage their health using artificial intelligence and advanced technology, ultimately improving outcomes and overall well-being.

* 1. **FUTURE SCOPE AND WORK SUGGESTIONS**

Wearable device integration. By integrating data from wearable health monitoring devices such as fitness trackers and smartwatches, real-time physiological data can be obtained for more accurate health predictions. Examples include heart rate variability, sleep habits, physical activity, and other factors.

**Genomic data analysis:** Genomic data analysis based on a person's genetic makeup enables personalized health predictions. By incorporating genetic data, prevention and treatment plans can be tailored to provide insights into a person's propensity to develop particular diseases or conditions.

**Environmental factors:** By incorporating environmental data such as pollution levels, air quality and geographical location, better knowledge can be gained about variables that influence health status. This will enable predictive models to take into account the impact of the environment on health.

**Behavioral and lifestyle analysis:** More accurately predict health outcomes by considering lifestyle factors such as social interactions, stress levels, and dietary and exercise habits. Integrating behavioral data enables individual coaching and comprehensive health assessments. AI and ML are currently being used to develop algorithms for telemedicine and remote monitoring. Developing predictive models is one way to address this problem. This allows health problems to be identified early and treated quickly, reducing the need for doctor visits.

**Electronic Health Record (EHR) Integration:** Integrate AI-powered health predictive models into EHR systems and leverage historical patient data for predictive analytics. This allows healthcare providers to proactively identify and treat patients who may develop certain diseases in the future.

Ethical Concerns and Privacy: Considers ethical and privacy considerations related to the collection and use of personal health information. To maintain trust and protect patient privacy, you must implement strict security measures in accordance with regulations such as GDPR and HIPAA.

Continuous improvement and further development of the model can be achieved through the use of iterative learning and feedback loops. Predictive models take into account user input, new data sources, and advances in AI technology to keep them accurate and up-to-date.

**REFERENCES**

[1] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network," in *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, 2018: IEEE, pp. 122-129.

[2] L. Alzubaidi *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *Journal of big Data,* vol. 8, no. 1, pp. 1-74, 2021.

[3] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational intelligence and neuroscience,* vol. 2016, 2016.

[4] H.-C. Shin *et al.*, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging,* vol. 35, no. 5, pp. 1285-1298, 2016.

[5] K. Denecke and Y. Deng, "Sentiment analysis in medical settings: New opportunities and challenges," *Artificial intelligence in medicine,* vol. 64, no. 1, pp. 17-27, 2015.

[6] Ł. Kidziński, B. Yang, J. L. Hicks, A. Rajagopal, S. L. Delp, and M. H. Schwartz, "Deep neural networks enable quantitative movement analysis using single-camera videos," *Nature communications,* vol. 11, no. 1, pp. 1-10, 2020.

[7] A. Demidovskij *et al.*, "OpenVINO deep learning workbench: Towards analytical platform for neural networks inference optimization," in *Journal of physics: Conference series*, 2021, vol. 1828, no. 1: IOP Publishing, p. 01 2012.

[8] A. Hassouneh, A. Mutawa, and M. Murugappan, "Development of a real-time emotion recognition system using facial expressions and EEG based on machine learning and deep neural network methods," *Informatics in Medicine Unlocked,* vol. 20, p. 100372, 2020.

[9] K. Patel *et al.*, "Facial sentiment analysis using AI techniques: state-of-the-art, taxonomies, and challenges," *IEEE Access,* vol. 8, pp. 90495-90519, 2020.

[10] F. Ahmed, A. H. Bari, and M. L. Gavrilova, "Emotion recognition from body movement," *IEEE Access,* vol. 8, pp. 11761-11781, 2019.

[11] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation,* vol. 29, no. 9, pp. 2352-2449, 2017.

[12] M. Awais *et al.*, "LSTM-based emotion detection using physiological signals: IoT framework for healthcare and distance learning in COVID-19," *IEEE Internet of Things Journal,* vol. 8, no. 23, pp. 16863-16871, 2020.

[13] A. Radaideh, F. Dweiri, and M. Obaidat, "A novel approach to predict the real time sentimental analysis by naive bayes & rnn algorithm during the covid pandemic in uae," in *2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, 2020: IEEE, pp. 1-5.

[14] B. Bhavitha, A. P. Rodrigues, and N. N. Chiplunkar, "Comparative study of machine learning techniques in sentimental analysis," in *2017 International conference on inventive communication and computational technologies (ICICCT)*, 2017: IEEE, pp. 216-221.

[15] https://in.images.search.yahoo.com/search/images;\_ylt=Awrx\_e0BJUZmZccMetu9HAx.;\_ylu=c2VjA3NlYXJjaARzbGsDYnV0dG9u;\_ylc=X1MDMjExNDcyMzAwNQRfcgMyBGZyA21jYWZlZQRmcjIDcDpzLHY6aSxtOnNiLXRvcARncHJpZAMxS2ZMaF9NbFNNMllSU1J0b3lSZWxBBG5fcnNsdAMwBG5fc3VnZwMxMARvcmlnaW4DaW4uaW1hZ2VzLnNlYXJjaC55YWhvby5jb20EcG9zAzAEcHFzdHIDBHBxc3RybAMwBHFzdHJsAzExBHF1ZXJ5A21vYmlsZSUyMG5ldCUyMAR0X3N0bXADMTcxNTg3MzA1Mg--?p=mobile+net+&fr=mcafee&fr2=p%3As%2Cv%3Ai%2Cm%3Asb-dtop&ei=UTF-8&x=wrt&type=E210IN826G0#id=5&iurl=https%3A%2F%2Fwikidocs.net%2Fimages%2Fpage%2F164799%2FMobile\_Fig\_01.png&action=click

[16] https://media.istockphoto.com/id/1272117163/photo/sick-man-on-train-cough-and-make-other-people-feel-worry-about-virus-spreading.jpg?s=612x612&w=0&k=20&c=rVvpCsjTbYCqX461UGLjP92I3RL9W9n1gRoA7DhYx8k=

[17] https://media.istockphoto.com/id/1297166259/photo/senior-man-coughing-into-elbow-while-commuting-by-bus.jpg?s=612x612&w=0&k=20&c=kLLmV50MgSzpC\_NYwjUPVTYw8IdhiAX2DHC56a7B0PE=

[18] https://www.geeksforgeeks.org/openpose-human-pose-estimation-method/