A Hybrid Approach to Determine Patients Critical Situation using Expression & Posture with Convolutional Neural Network & Blazepose Algorithm

by

Abrar Islam
17101217
Orbin Ahmed Acanto
17201023
Mehzabin Islam Drishty
18101315
Samonty Zaman
19201102
Jaber Ahmed
19201131

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

Department of Computer Science and Engineering
BRAC University
January 2022

Declaration

It is hereby declared that,

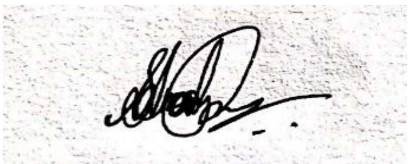
1. The thesis submitted is our original work while completing a degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and referenced accurately.
3. The thesis does not contain material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

Student's Full Name & Signature:

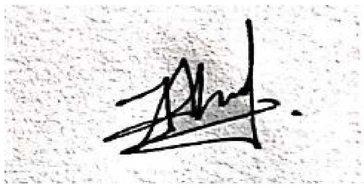
Abrar Islam
17101217

Orbin Ahmed Acanto
17201023

Mehzabin Islam Drishty
18101315



Samonty Zaman
19201102



Jaber Ahmed
19201131

Approval

The thesis titled "A Hybrid Approach to Determine Patients Critical Situation using Expression & Posture with Convolutional Neural Network & Blazepose Algorithm" submitted by

1. Abrar Islam (17101217)
2. Orbin Ahmed Acanto (17201023)
3. Mehzabin Islam Drishty (18101315)
4. Jaber Ahmed (19201131)
5. Samonty Zaman (19201102)

As of Spring, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in January, 2022.

Examining Committee:

Supervisor:
(Member)

Md. Khalilur Rhaman

Md. Khalilur Rhaman
Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

Patients are considered to be one of the most vulnerable persons. When it comes to critical patients their movements and behaviors need to be monitored constantly as simple negligences could result in severe consequences. It is almost impossible to monitor a patient 24/7 without making any slight error. Therefore, this paper will establish a simple but effective solution to this issue by creating a heuristic approach system that can detect a patient's facial expressions and postural movement to calculate the immediate conditions of patients with the assistance of deep learning algorithms. This is a hybrid approach as we have combined Convolutional Neural Network & BlazePose GHUM 3D to create a robust model which in our system can be used for image analysis in order to get precise monitoring results for critical situations by following specific sequences that would not have been possible without the hybrid model.

Keywords: CNN; BlazePose GHUM 3D; Facial expression; Posture Sequence detection; Critical situation detection.

Acknowledgment

First of all, our humble gratitude towards Dr. Md Khalilur Rhaman for being our thesis supervisor and guiding us throughout the whole journey. Holding meetings whenever we reached out even in these tough pandemic situations. Support us thoroughly by showing us how to improve our system to make it the best version which it is today.

Secondly, our beloved parents, whose support and encouragement played a vital role in our education, and the completion of this thesis.

Finally, special thanks to our fellow mates M Fayed Rafid, Bushra Ahsan, Ashraf Shahriar, Tasfia Chowdhury, and Nabila Sultana Riza for helping us to enrich our dataset by providing sample videos.

Table of Contents

List of Figures

Chapter- 1

Introduction

1.1 Background

We can not but accept that we all are a part of a technological revolution. An era where we as human beings are achieving milestones that were unimaginable to hit a couple of decades ago. Science made it all possible, reaching heights previously implausible. Advancement in all kinds of sectors is happening thanks to massive research and implausible leaps in computer science. We, humans, are accustomed to socializing, and when we do that one of the distinct ways which make us more communicative is our body posture and facial expressions. Whether we are cheerful, anxious, sick, or sad we are more likely to express them through expressions. Not to mention the general assumption that non-verbal information is more persuasive, truthful, and revealing [1].

In the above circumstances, we want to conduct research on this hybrid model, specifically with facial expression and body posture detection. So, the purpose of our research is to find a low-cost solution using computer vision and deep learning to help the patient during a critical situation.

Image Source: Bangla Tribune

1.2 Thesis Orientation

This section provides an outline of the topics covered in each chapter of this thesis paper. The remainder of the paper is organized logically after addressing the purpose of this study and what we desire to do and plan to accomplish.

Chapter 1 Includes the general concept of our work, the background, objectives, challenges, and detailed theoretical explanation.

Chapter 2 The existing knowledge of the current facial expression and posture detection state.

Chapter 3 Stating the issues that our system is trying to solve and works that have been done by others regarding it.

Chapter 4 The methods and data sets we are using in addition to our work plan.

Chapter 5 Detail discussion of pre-processing of data with the four critical stages (chest pain, headache, stomach ache, sleeping and falling from bed).

Chapter 6 Accuracy of DeepFace and CNN with the reasoning of why we chose CNN over DeepFace.

Chapter 7 Demonstration of facial expression and posture detection with OpenCV.

Chapter 8 Showcase of experiment result and analysis.

Chapter 9 Our paper comes to an end with a sum-up of our whole work including what could be done with it in the future.

1.3 Research Objectives

In this paper, we propose a heuristic approach to monitor facial expression and detect the posture of a patient. We aim to develop a machine that can detect the facial expressions and different postures of a critical patient. For example, we aim to detect whether a patient is sad, happy, anxious, or in fear. According to Kyaw Kyaw Htike and Othman O. Khalifa [11], posture detection is a very interesting and complicated sector in the vision of the computer because it has very effective applications in many areas among them personal health care is highly mentioned. We are using these two combinations to help patients. We intend to monitor both facial expression and posture detection of the patient. We would also consider a number of sequences when calculating the posture of the patient to get more accurate results like if the patient fell from the bed or is he suffering from chest pain or headache and many more. The objectives of this research are given below -

1. To create a developed model for automated monitoring.
2. To deeply understand the facial expression and different postures of patients.
3. To create a low-cost solution.
4. To solve the emergency issues of a patient immediately.
5. To evaluate the model and offer recommendations on improving the model.

Chapter- 2

Literature Review

Facial expression recognition has been a challenge for humans for a long time. Modern machine learning algorithms can solve those problems more accurately.

Facial expression: A human face can show different types of facial expressions to communicate. By studying facial expressions, one’s motive and affective state can be understood. This has an impact on many important fields such as security, medical, and many more. This can be done using an autonomous facial expression recognition structure. Autonomous facial expression recognition is a fascinating and demanding field in computer vision and machine learning. This field of study attracted many social scientists after the work in 1972 by Darwin [19]. Deriving facial expressions from actual faces is the main challenge. Many models are being used recently to provide more accurate data such as Self-Supervised Learning [10], Regional Attention Network (RAN), Pyramid with Super-Resolution (PSR) network, and many more. In the case of automatic facial expression, two major factors need to be worked on. The first one is a facial representation and the other one is classifier design. Patients’ facial expressions can be sad, fearful, anxious, and many more. Possible ways of taking input for expression recognition includes different types of video or image input, audio input, and biosignals. Those signals can later be compared with the base case of the patient to identify the patient's affective state in the present time situation. Using facial expression detection on a patient can be non-verbal communication between a doctor and a patient who is in trouble. Later on, doctors and nurses can serve patients more effectively in a real-time situation. To evaluate human face motions, Ekman and Friesen created the Facial Action Coding System [20]. However, this scheme generally requires skilled humans and takes a significant amount of time. Current progress in machine learning in computer vision that performs could aid in the simplification and automation of these processes. Our research focuses on automatic facial expression recognition.

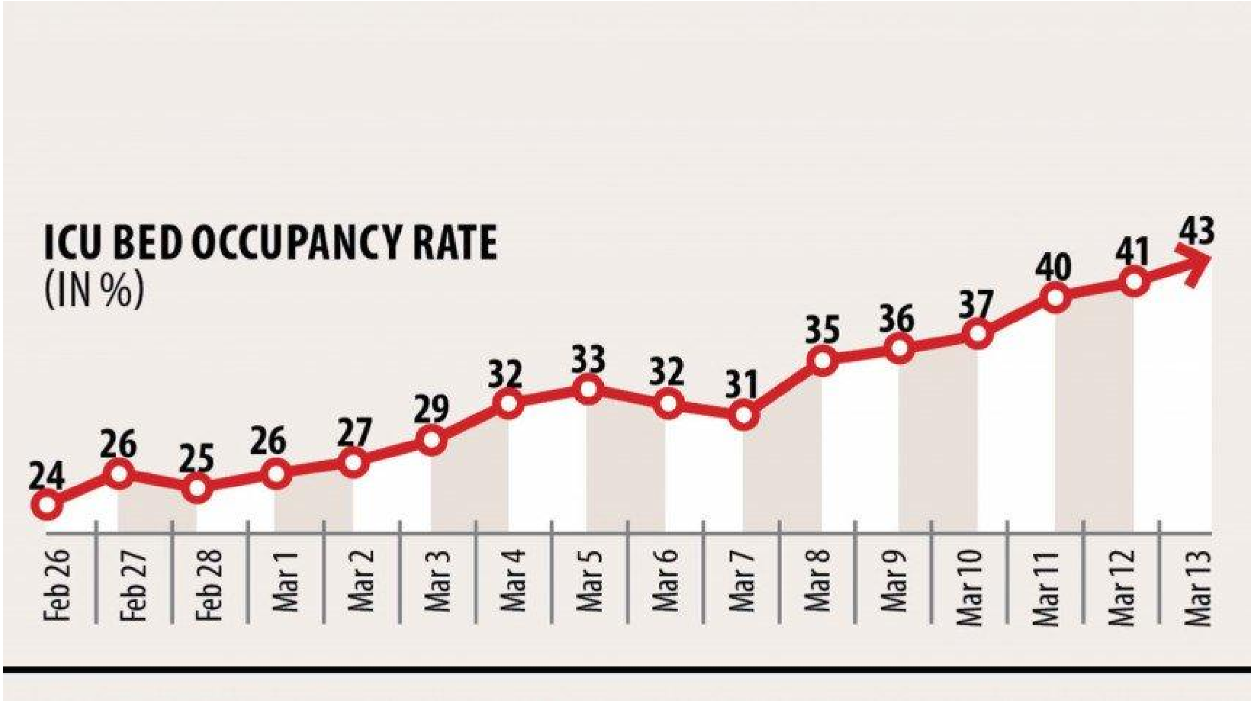
Posture detection: Body Posture can expressway more than it seems. Body Posture became a matter of concern back in the 1880s. The human body carries more information than we can ever imagine. It tends to react to its environment, at least it tries to react. As in, we are dealing with all cases on an extreme level, doctors and nurses also need to lend that extra care and observation. Along with facial expression, body posture movement also explains a lot. Even a blink can be a very important signal in many cases. Sometimes doctors also need to depend on only body movements. According to Mayo Clinic [21], Doctors must rely on physical clues of a patient as people in Coma cannot express themselves. Physical clues include not only any body part movement but as subtle as changes to the pupil size. We know that the light level of the environment controls its dilation. Even emotion can make a difference in pupil size. Observing breathing patterns is also important. Everything revolves in a pattern. One must observe it round the clock to understand it. Just like a coma patient for any other medical emergency, it is important to have a close observation of these kinds of signals given by body movement and posture. Hence, Posture detection is undoubtedly vital for patients. According to Foley GN [22], It has been led to the belief that between 60% to 65% of all communication is caused by body language. According to Kendra Cherry [23], Though understanding body language is important, it is also important to pay attention to other signs such as context. You should treat the signal as a whole rather than focus on a single operation. Gestures like a clenched fist, pointing at someone, saying something with fingers can be counted as a crucial signal. A thumbs up and thumbs down can be used for approving or disapproving something. Not only that, even arms and legs convey huge messages. Closed arms say something, open arms say something else. The position of hands, fingers, face, legs, even gaze is a great deal. It is a pretty perplexing situation to understand these signals and act accordingly. But that is why our system will do the work and detect all of these signals without any man-made error.

Chapter- 3

Research Problems & Related works

3.1 Research Problems

In Bangladesh, around one hundred hospitals are having services. [2]. Therefore, every day there is an immense number of critical patients getting admitted to these hospitals. In Bangladesh, the current doctor-patient ratio is only 5.26 per 10,000 people. [3]. By this ratio, we can get an idea that this number is way too low than it should have been. Most of the time it is even hard to find vacancies in the hospital wards for patients. The situation gets even worse when we consider public hospitals. For this handful of doctors to serve thousands of patients, to monitor their every movement is nearly impossible. Doctors are the last line of hope, a protector against diseases, and the savior of mankind. As they are humans, they can not be everywhere at all times. Especially in Bangladesh where the number of patients is constantly rising. A recent study by “The Daily Star” from 26/2/21 to 13/3/21 shows how rapidly the number of beds is being occupied [4]. While dealing with a patient or a patient who is in a critical stage the doctors and nurses have to be more tentative as these patients need more care than any other. These critical patients are struggling with their life and death. Therefore, they need constant monitoring so that whenever a slight spike is noticed or even an abnormal behavior occurs the patient gets attended to at once. It is becoming an impossible task for doctors and nurses to monitor this immense number of patients all the time. Though the continuous monitoring of a patient’s vital parameters is one of the most important aspects of an ICU.



[4]

A revolutionary solution to this kind of implementation could be monitoring patients by detecting their facial expressions combined with posture and acting according to that. After the invention of mortality was reduced from 90% to 40%. According to a study done by Bangladesh Crit Care in 2019, most (80%) had a nurse-to-patient ratio of 1:1 or 1:2. It was 1:1 in 27%, 1:2 in 53% and 1:3 or more in 20%. Multivariate analysis showed that the presence of 1:1 nursing staff was much lower in private hospitals [5]. Significant matters of a patient's treatment procedure are still not observed automatically whereas that could play a crucial role. For instance, natural facts which include sleeping disturbance and dementia-like loud noise, condition of room light, and exorbitant visits during rest time are not measured. Many different facts like the patient's expression due to pain and suffering, suffocation, and many other sentiment states are not traced steadily. Just think of the possibilities of how much easier it would be if we could achieve all these and how fruitful the results would be. Continuous monitoring of a patient's crucial signs or physiological functions helps ensure patient safety by alerting them to critical changes in their health status, as well as guiding daily treatment measures [6][7]. The ability to recognize patient deterioration early and intervene quickly is important to save patients' lives [6][7]. The main problem with building our hybrid system is building a low-cost solution along with maintaining an accurate result, as we are just using image data here for the result. Without using additional sensors, we are using only image and video data feed from the camera.

3.2 Related works

Methods to detect emotion can be divided into two subcategories, one of them is Conventional methods and the other one is Deep learning methods. Convolutional layers are one of several deep learning models that play important roles in Face recognition. By providing end-to-end learning from inputs, these models eliminate the need for preprocessing techniques entirely [12]. By combining two types of feature extraction modes [13], they start by utilizing features learned by CNN models with portraits and dense networks. Second, handcrafted attributes were removed by a bag of visual words. These created a local classification method after image segmentation mode to categorize each label for images. Their model consisted of three stages: where they used the K-nearest Neighbor model to select the closest training samples for data analysis images in the first part. [14][15]. There are many other methods proposed by researchers in recent years such as multiple kernel learning [16][17], Multiple feature fusion [16], and score level fusion was reported useful in facial expression recognition [18]. These things have been already experimented on several times. Experiments by “Wiley Periodicals, Inc”. on behalf of the American Association of Physicists in Medicine, developed a system that tracks the patient's facial expressions and forecasts their advanced movement during therapy [8]. But they were only able to complete the recognition based on the patient’s comfortable and uncomfortable expressions. Another experiment was done by Youngsu Cha, Kihyuk Nam, and Doik Kim from the Center for Robotics Research, Korea Institute of Science and Technology, Korea. They proposed a patient position monitoring system based on a patient cloth with unobtrusive sensors. The sensor generated electrical responses corresponding to the bending and extension of each joint [9]. But the issue was their idea wasn’t that stable and the testing was done with only 3 patients with a 88% accuracy. Where in our case we get the upper hand as we are using cameras that are more simple to operate and maintain, with the additional benefit of deep learning, which enables our system to improve by training itself over time. Combining all the data and analyzing them can give us the desired result we want. This could be the next big thing we have been waiting for.

Chapter- 4

Methodology

To detect face and posture detection we are using CNNs here. Convolutional neural networks (CNNs) are artificial neural networks that are specifically designed to process pixel data and are used in image recognition and processing. The model needs to recognize the expression to detect a certain situation. We have used Keras API, TensorFlow, OpenCV and CNN to make this facial and posture detection possible. Our main focus was to convert an image into a pixel and detect its expression.

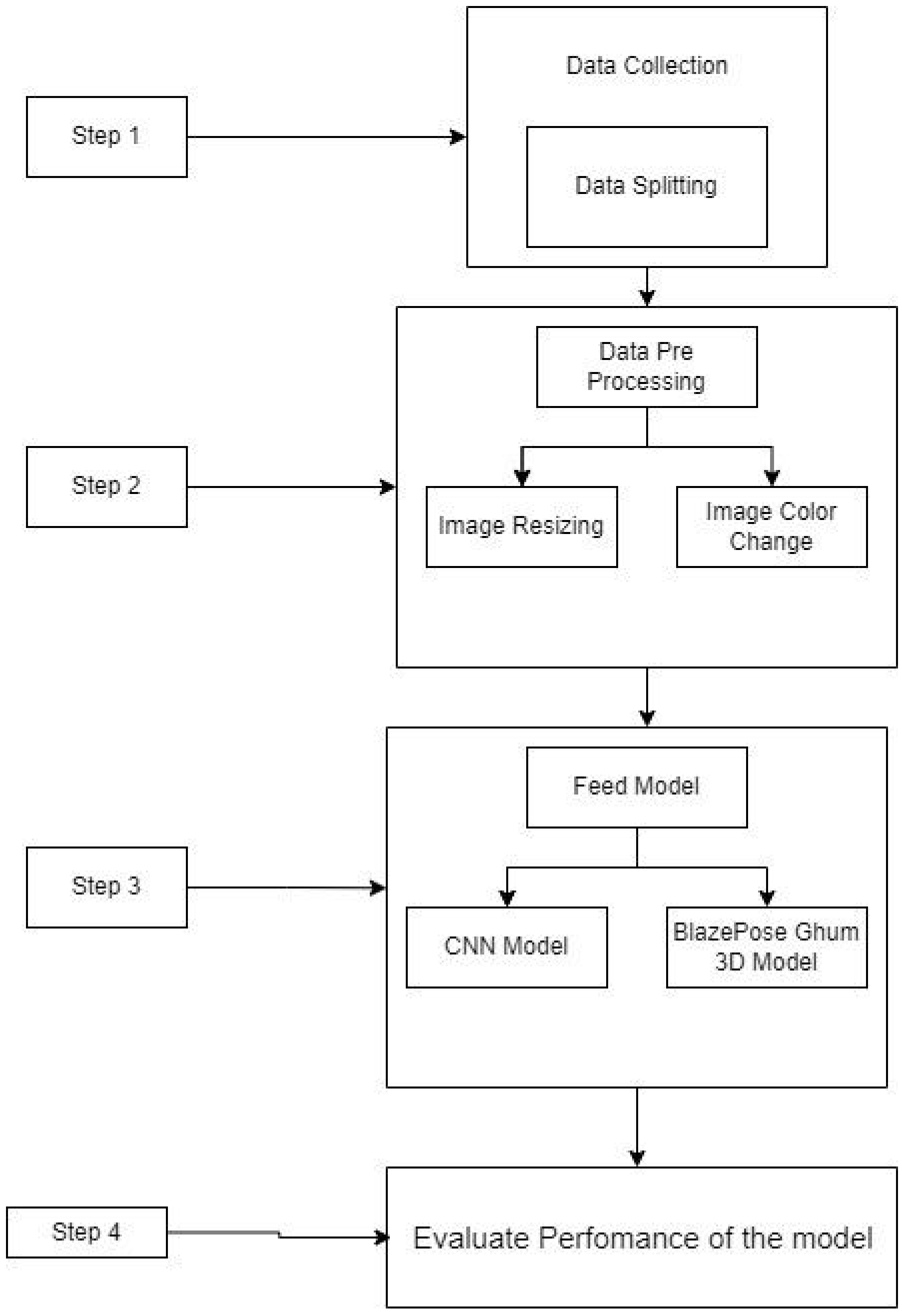


Fig 4: Workflow of methodology

Workflow: The purpose of this proposed hybrid model of facial expression and posture detection is to help medical patients in their emergency.

1. We have collected datasets from the FER2013 for facial detection and have created our own dataset to identify posture detection from given image or real time videos. Here, we have extracted facial features of images from FER2013.

2. On the pre-process step, We are resizing any given input into 48 X 48 pixels and converting it into a Gray image. Moreover, we are storing labels and features in a numpy array.
3. Then, for the training phase, we will feed the facial features of training datasets of FER 2013 into the CNN model and our own datasets of scaled videos and images to the Blazepose algorithm.
4. After that, we run the test photos from the FER2013 datasets to check the accuracy, as well as the same videos dataset to check the accuracy of our posture detection.

Chapter- 5

Details Discussion of Our Work

The entire process of how our system operates is detailed in our block diagram. which shows the major components or functions represented by blocks connected by lines. The relationships between the blocks are depicted by these lines. Since our system must detect facial and postural expressions, it is critical to establish the precise angles and distance using body landmarks that can identify the patient's attitude and the situation at that very moment.

For this, we have used a MediaPipe library and a CNN model to work on hybrid Algorithm, which is to recognize faces and body landmarks in an image or a real-time feed, which is a haar cascade [28]. Using the CNN model to identify the facial expression and MediaPipe to get the body posture.

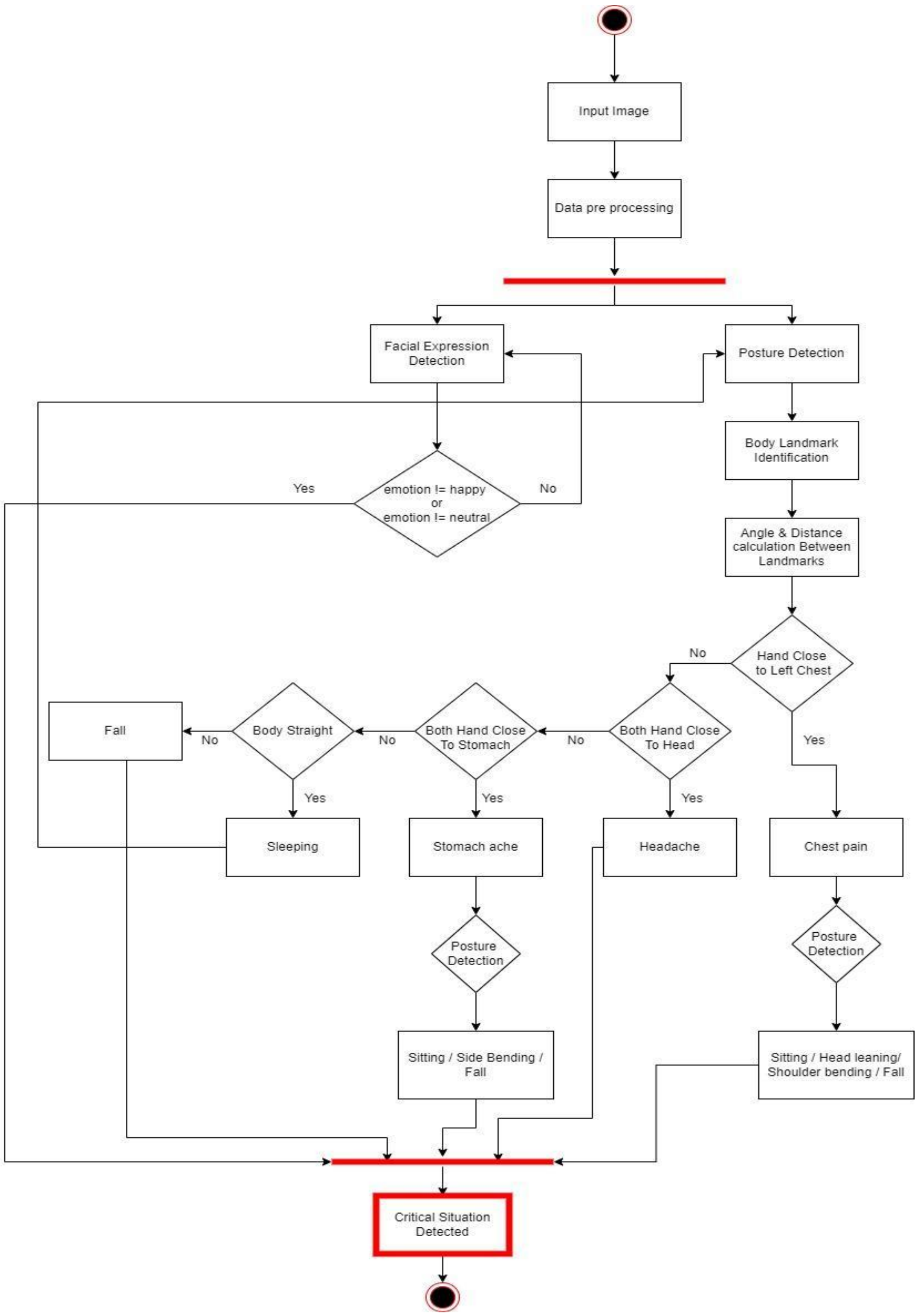


Fig 5: Block diagram of our work

5.1 Data Pre Processing

We have used the haar cascade for data pre-processing. It is a classifier algorithm that is used to detect and extract faces from an image or real-time data feed. After inserting a photo or real-time video, it resizes it to 48 x 48 pixels. Then we transform them to gray images to apply the Haar cascade classifier. In the posture module, the image is transformed into RGB images to feed the blazepose algorithm, which will later be used to find out body landmarks. Following that, it is divided into two categories: They are "Facial Expression Detection" and "Posture Detection".

```
Preprocessing Done
Number of Features: 48
Number of Labels: 7
Number of examples in dataset:35887
X,y stored in fdataX.npy and flabels.npy respectively
```

Fig 5.1: Output of processed data

5.2 Facial Expression Detection

Initially, we are converting the RGB image into a gray image. Then we convert the images to 48 by 48 pixels. Later, we applied the Haar cascade classifier to find faces in an image. To be more specific, we're looking for the expressions "angry," "disgust," "pain," "happy," "sad," "fear," and"neutral." Moreover, in our system, we are using a CNN model that has been trained and validated with the FER 2013 dataset [24]. Our model has 66% accuracy.

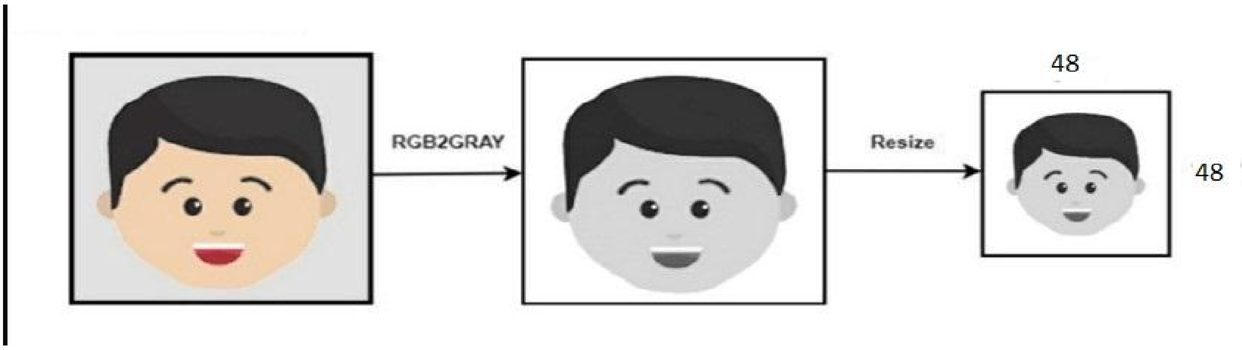


Fig 5.2: Image conversion

In the first instance, we are focusing on normalizing the image. We anticipate whether the image evokes happy feelings or neutral sentiments immediately after neutralizing it. If the given input contains happy or neutral expressions, we will rerun it without the flag turned on. The flag generally notifies authorities about critical situations, but we do not want that here, so we do not go for it. On the contrary, if the image contains any other emotion except happiness or neutral, we immediately send a notification to the main module, which will later authorize the critical situation.

5.3 Posture Detection

In our posture detection part, we are using the MediaPipe library. In the mediapipe library, the posture detection algorithm is a blazepose algorithm to identify landmarks at body keypoints [25]. This library has three modules. As models increase, the accuracy increases too, but, on the other hand, speed decreases. Here we are using model 2, which has the best result among others but competitively slower results. Here we are converting a BGR image to an RGB image.

Initially, we are detecting key points as landmarks. These landmarks are very crucial in terms of identifying different angles and distances between various postures. Each posture depends on different angles of degrees. Where a hip close to 90 degrees expresses one particular posture and more than 90 degrees indicates some other posture. Even a degree matters in terms of identifying these positions. For this, we pointed and found precise landmarks for exact positions. Then we apply our model to identify the posture and sequence. To detect those sequences we need accurate posture information from our posture module. Then using those landmark information we calculate the angles and distance then apply a heuristic approach to determine accurate posture of patients.

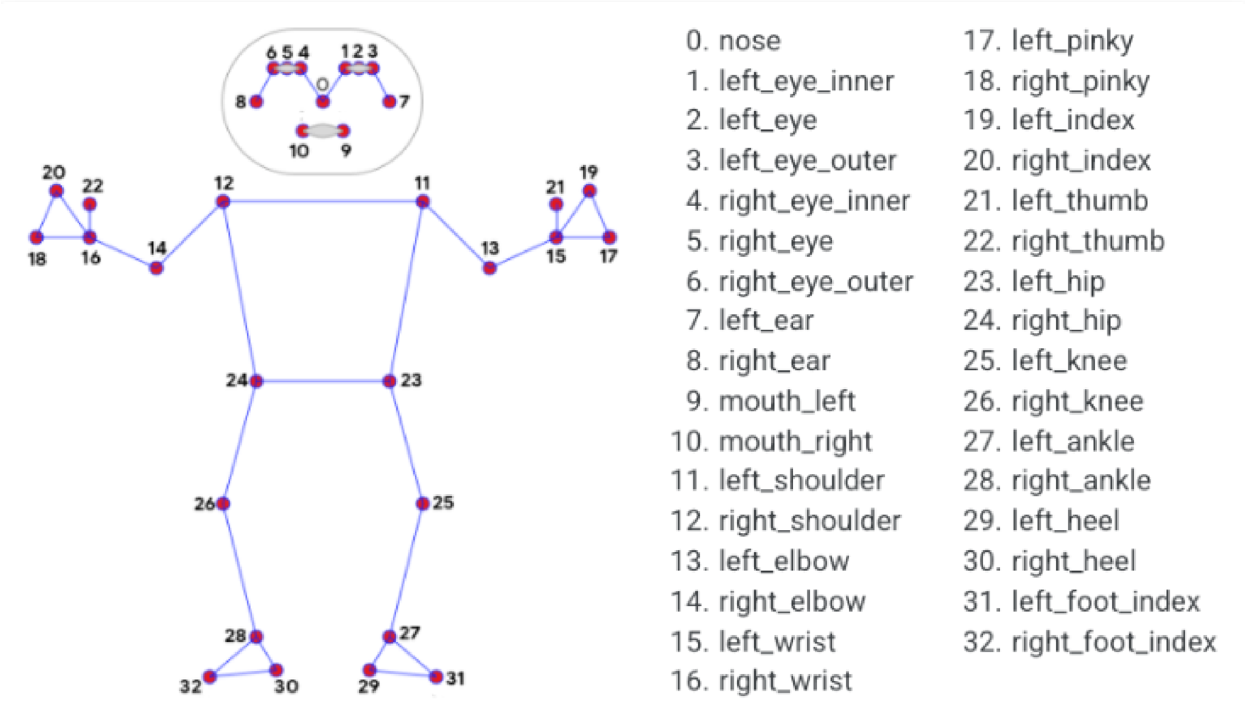


Fig 5.3: Pose landmarks

5.4 Chest Pain

To get "Chest Pain" as an accurate output, we have chosen some particular landmarks. Chest pain most of the time occurs on the left side of the chest, as there is a high probability of a person touching his/her left chest with the left hand. When the hand is close to the left chest or if it is touching the left chest, the "Chest Pain" output is shown. At the same time, if the hip angle changes, the different output will be shown, along with showing that the patient is having chest pain. This is the sequence detection part in our system. The system normally detects chest pain along with the body posture. Then it detects the sequence as sitting, head leaning, shoulder bending and lastly fall. If all those sequences are detected of a patient then the doctor or authority will be notified immediately.

Fig 5.4: Chest pain with expression

5.4.1 Chest Pain While Sitting

When the hip angle is near 90 degrees and the hand is in the left chest, the output will say "Sitting" with "Chest Pain." It indicates that the patient is experiencing chest pain and that his or her posture is "Sitting."

Fig 5.4.1: Chest pain while sitting without expression

5.4.2 Chest Pain While Head Leaning

In addition, if the hip angle decreases significantly and the head leans forward, "Head Leaning" is displayed as an output alongside "Chest Pain." as the faces were not detected by the system, that is why expression is shown as none.

Fig 5.4.2: Chest pain while head leaning without expression

5.4.3 Chest Pain While Shoulder Bending

If, on the other hand, the head angle shifts 180 degrees forward and the left leg bends slightly as the hip angle shifts, this is referred to as "shoulder bending." In this example, if the hip angle decreases in comparison to the "head leaning" circumstance, "shoulder bending" is considered.

Fig 5.4.3: Chest pain while shoulder bending without expression

5.4.4 Chest Pain While Falling

Finally, if the left leg bends entirely and the hip angle increases from the previous position and the head angle shifts from 130 degrees to 200 degrees, then the result shows "fall out" as his/her posture when experiencing chest pain.

Fig 5.4.4: Chest pain while falling without expression

5.5 Headache

To detect "Headache" as an output, we need to identify if the patient is touching his/her forehead or not. As it is a common reaction when someone has a headache, Human beings tend to touch their heads if there is any sort of pain they experience. So in our system, if both hands are touching the forehead, then we are showing "Headache" as an output. Even in this case, we have a particular angle of hand. Elbows should be at an angle of 20 to 60 degrees on both left and right hands.

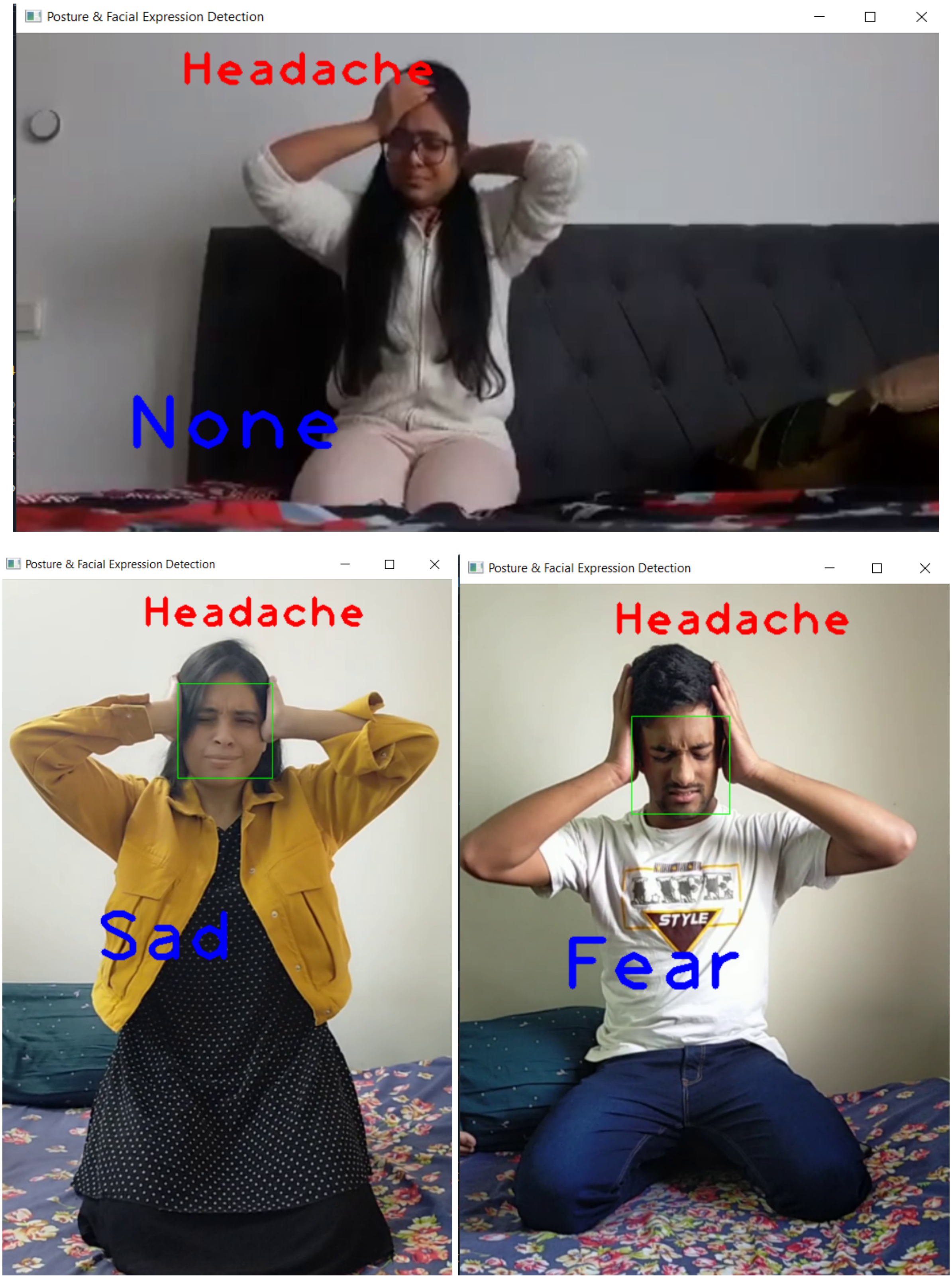


Fig 5.5 : Headache with and without expression

5.6 Stomach Ache

To detect stomach aches, we first need to find if our hands are anywhere near the abdomen portion. Here we are targeting both the upper and lower abdomen as pain could be in any part of the stomach. While touching our abdomen, our hands tend to shift from 60 degrees to 110 degrees. And our hip angle stays close to 110 to 310 degrees. That is the maximum shift that can happen to most patients' hips while experiencing stomach aches. With the help of these three conditions, we are addressing this situation as a stomach ache. After that we use those values and apply a heuristic approach to identify the accurate sequence of patient posture.

Fig 5.6 : Stomach ache without expression

5.6.1 Stomach Ache While Sitting

Three possible poses can occur while the stomach aches. To detect that posture precisely, we calculated exact angles. For sitting posture, the left-hand angle should be anywhere from 60 to 180 degrees. Even for the right hand, it should be the same, but in the MediaPipe library, it calculates the right hand's angle a bit differently. To determine the right hand's angle, it works in reverse. The angle is from 170 to 300 degrees for the right hand, along with the left-hand angle. The hip angle should be between 180 and 300 degrees. We've also introduced "value" as a variable, which is used to calculate the head-side nod. It estimates the distance between the right ear and the right shoulder and the distance between the left ear and the left shoulder to calculate head nodding. After calculating those distances, we subtract both values to get our value. If the absolute number of the value is less than 20, the patient's sitting position is indicated. So to indicate a "sitting" position, all these conditions should be filled.

Fig 5.6.1: Stomach ache while sitting with expression

5.6.2 Stomach Ache While Side Bending

For the next posture, the hands should be in the same order as in the previous situation, but the hip should shift from 140 degrees to 225 degrees. This scenario indicates "side bending" as an output.

Fig 5.6.2: Stomach ache while side bending with expression

5.6.3 Stomach Ache While Falling

If the left hand is on the abdomen and the right hand is on the ground, creating an angle of 130° to 260° for the hands and less than 190° for the hip angle, the previous posture changes and identifies a new position. Which should be the position of "Fall", but to detect "Fall" as a posture, it needs to be "emotion= null". It means the system will recognize the position as "fall" only if the face is not found or emotion is not shown in the face. So, to designate "Fall" as a position, all of the above conditions must be met.

Fig 5.6.3: Stomach ache while falling without expression

5.7 Falling From Bed While Sleeping

Detecting this position is very close to basic. If the body posture is straight, then it will identify "sleeping" as a position. However, a patient can often be found lying on the ground as a result of a fall. To identify that particular scenario, we need to check if the hips and legs are bending and if landmarks on the face can be detected. If the hips and legs are bent and the face is not visible, it may be a fall, which will be labeled as such.

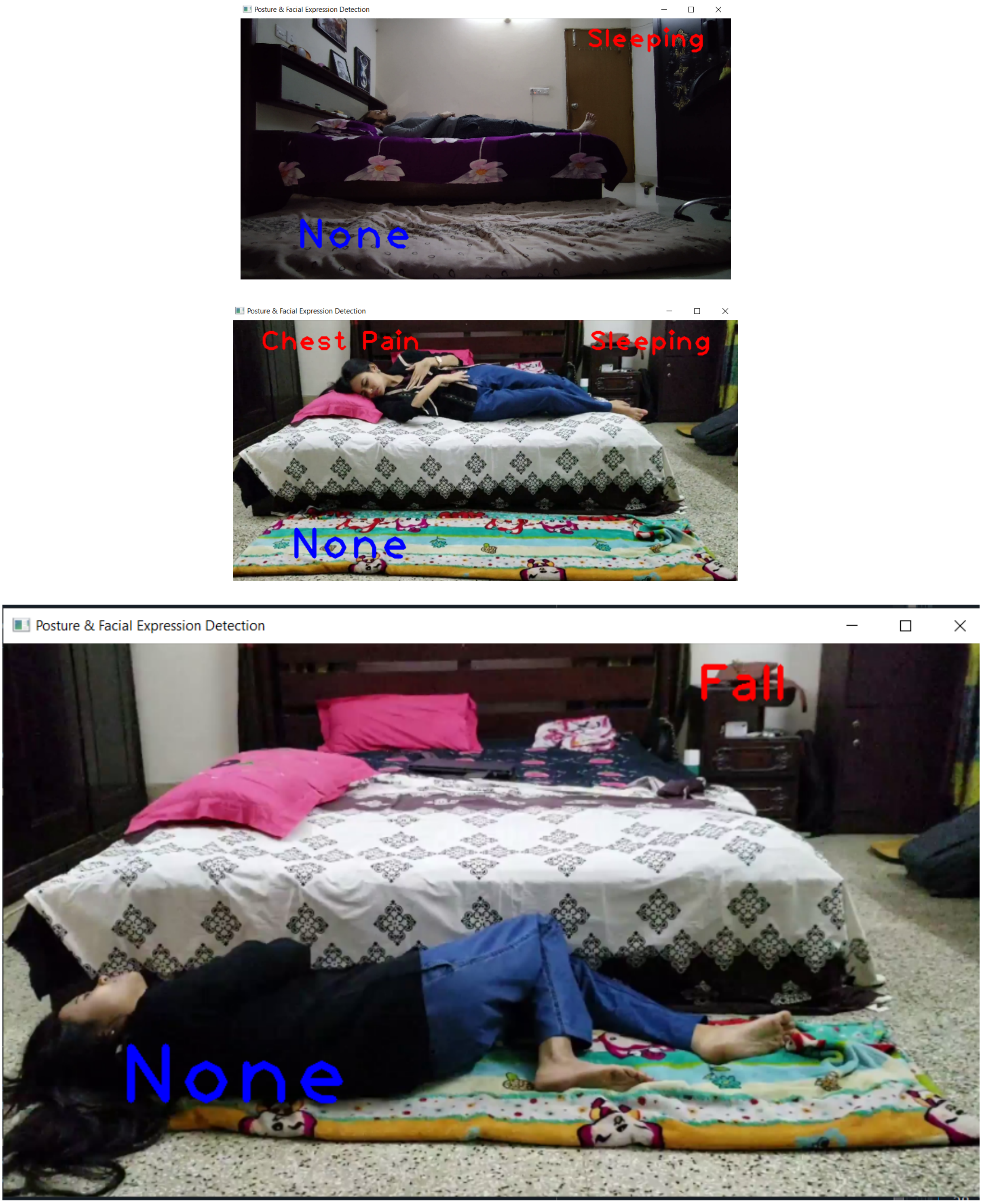


Fig 5.7 : Falling from bed while sleeping without expression

In the case of any of the above-mentioned situations, the flag will be 1, which means a critical situation has been detected. The flag normally notifies the main module, alerting it to the critical condition. As a result, nurse stations and doctors can be alerted to the urgency of the situation. Hence, necessary steps can be taken immediately, which can save many lives.

Chapter- 6

Alternative Model

We initially decided that we would use the DeepFace model for facial expression detection. DeepFace is a well-known facial expression recognition model that is used on popular websites like Facebook. It has an accuracy of over 97% in identifying facial expressions [26]. Next, we use the CNN model, which we build and train according to our needs from the very beginning, to identify facial expressions. We have designed this model in such a way that we can use it in our medical cases. However, if we were to use the deep face model, our accuracy would be even higher, which is now 66%. This accuracy can be further enhanced if we can train our model according to our medical purpose.

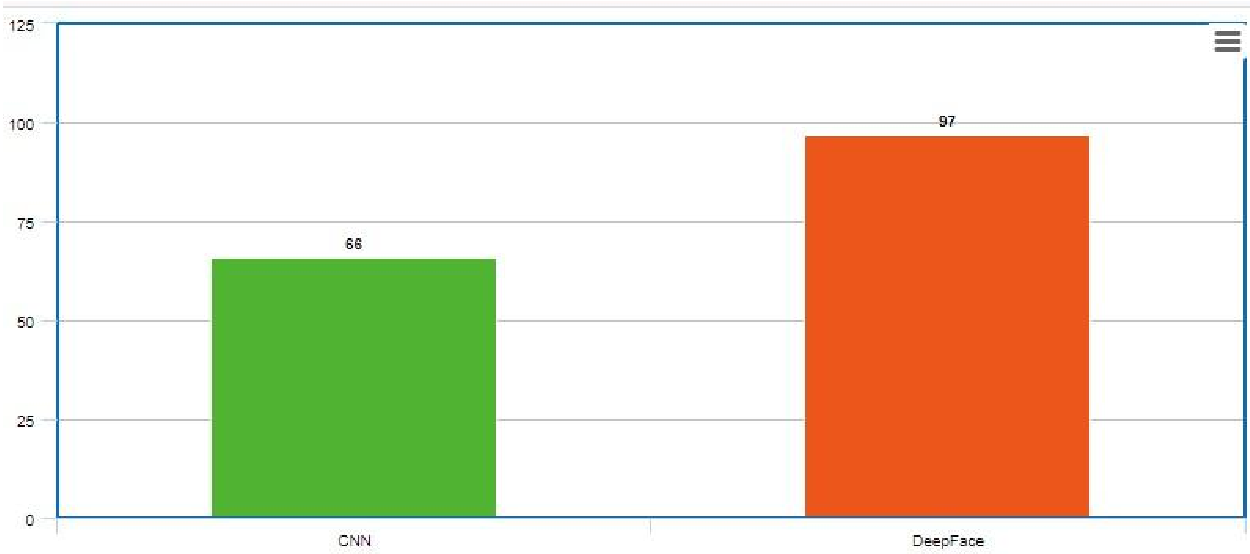


Fig 6: Accuracy of our CNN model and DeepFace model

Chapter- 7

Demonstration

7.1 Facial Expression Recognition Data Set

When it comes to facial expression and posture detection, we are adopting the FER-2013 dataset, which is essentially a collection of the results of a Google image search for each equivalent expression.

The data is a combination of grayscale images of faces with a pixel size of 48x48. To make the face almost centered, the faces are automatically registered. This also ends up consuming the same amount of space for every image which also helps in improving both accuracy and consistency.

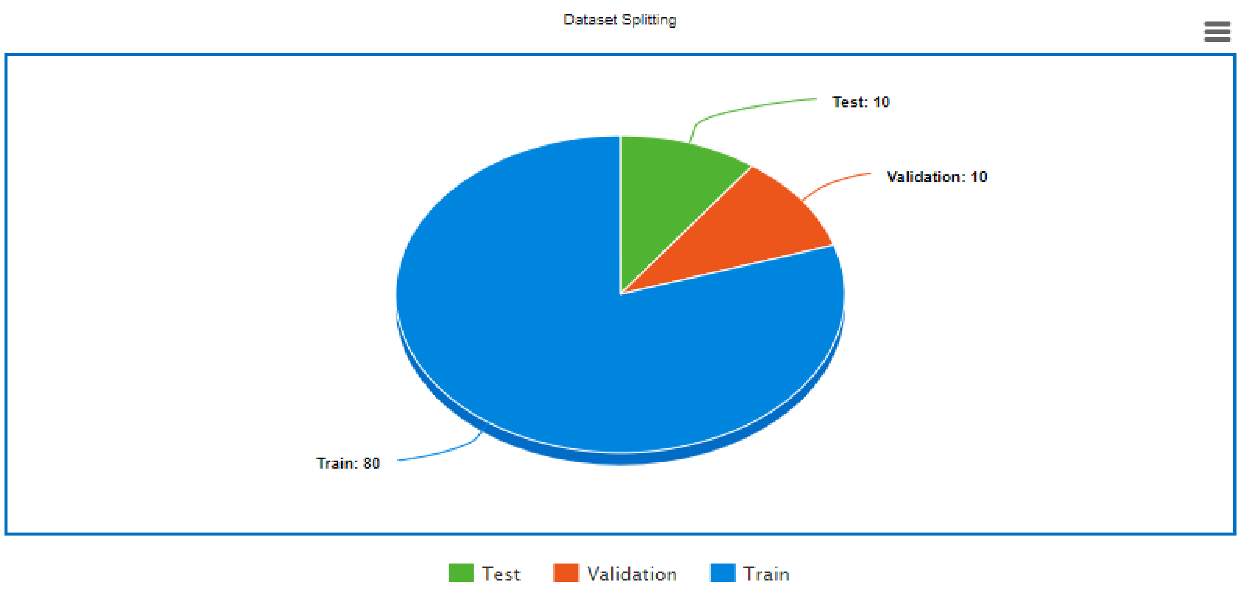


Fig 7.1: FER-2013 datasets test-train-validation

The original goal is to categorize every face based on the sentiment appearing in the facial expression into a total of seven categories, which are (0 for angry, 1 for disgust, 2 for fear, 3 for happy, 4 for sad, 5 for surprise, and 6 for neutral). It should also be mentioned that the public test and training set consists of 3,589 and 28,709 examples, respectively. [27]

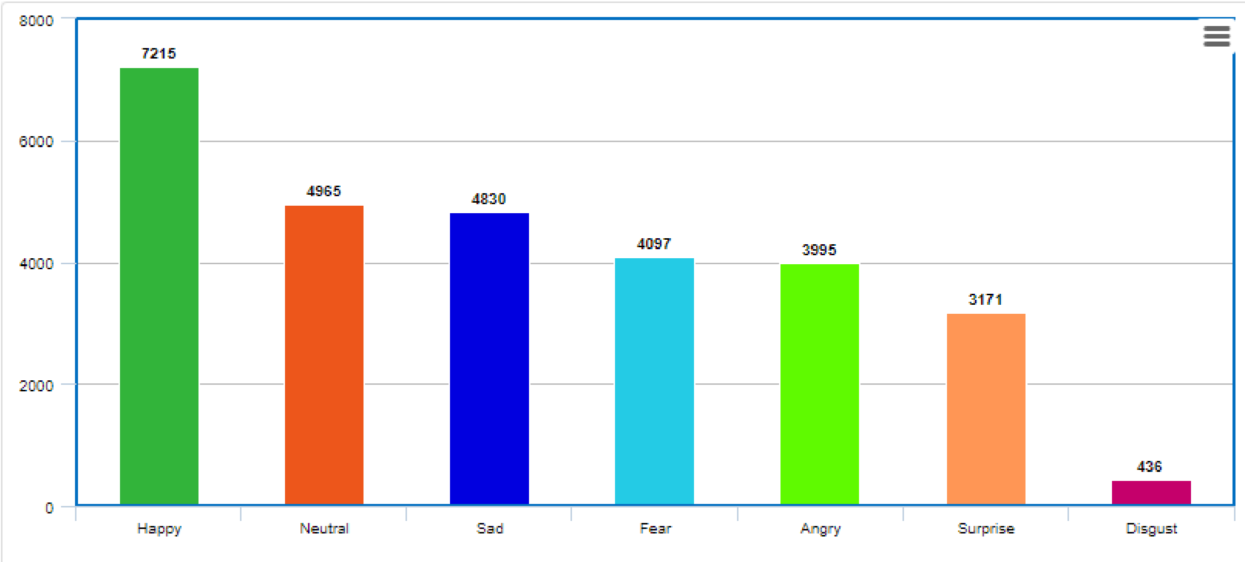


Fig 7.1.1: Total data of different labels

This dataset was prepared by Pierre-Luc Carrier and Aaron Courville as part of an ongoing research project and can be downloaded from Kaggle. [29]

When it comes to the advantages of using this data set, we can easily state that this dataset has been proven to be easier to understand and includes essential metadata with rich and machine-readable file formats. In addition, FER-2013 has a public kernel or task that provides us with assurances that the dataset is maintained. This dataset is well versatile with different labels of expressions.

7.2 Posture Detection Data Set

Since any hospital patient's data is pretty much confidential, it was quite hard for us to gather precise data. Therefore, we reached out to Dr. Iqra Tabassum Orna (MBBS, Sir Salimullah Medical College) for the general concept of the situations patients usually go through, including their postures. Then, with her supervision, we created sample videos to train our system. We manually recorded 113 videos and over 100 images. In our dataset, we used videos of nine individuals in different light conditions and scenarios. We are detecting 4 critical situations including 2 sequences of them. Our data set is well versatile among those critical situations.

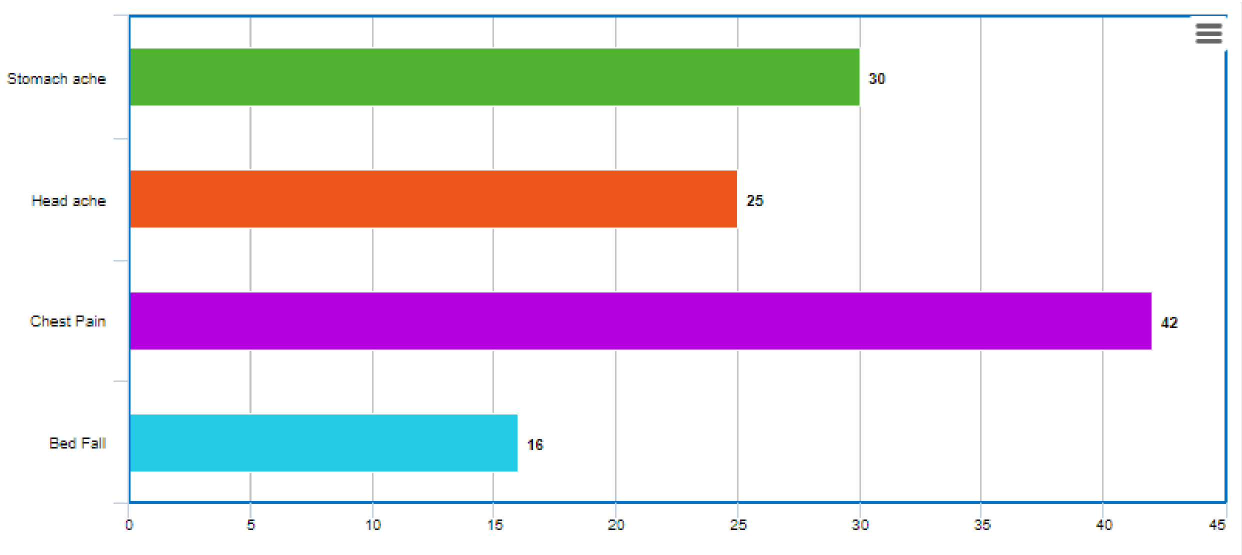


Fig 7.2: Posture detection dataset

7.3 Implementation

In our system, different kinds of algorithms mentioned in the previous section were used to detect the expressions and body posture of a patient. Results after using our system are mentioned in the following section as well.

7.4 Facial Expression Recognition

As mentioned earlier, we used FER2013 datasets in our facial expression recognition. After completing the data preprocessing, we started creating our convolutional neural network. In this case, we use a sequential model in the Keras API. In the convolutional network, first, it takes an image. Then it converts it into pixel data and stores it in a 2D array. There are different types of layers. We have added 11 layers in our CNN model to process the expression data set. There is a convolutional layer named "Conv2D". We keep the feature size at 64 and the window size at 3 by 3 for the first layer. The input shape is 48 by 48 as all the images in the dataset are this. In the activation, we use a rectified linear function to get a maximum value or 0 as an output. The batch normalizer layer normalizes its input data by dividing the value by 255, as the highest value in a pixel is 255. After those layers, we started to feed the pixel data into our model by using "MaxPooling2D". The input size is 2 by 2 of the 2D pixel array, and the window is shifted by the same amount as the input size. To control the feed and reduce the overfitting of our model, we use a dropout value of 0.5. We repeat this process three more times to create more layers in our convolutional neural network.

```
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', input_shape=(width, height, 1))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.5))
```

Fig 7.4: Adding convolutional layers to our CNN model

Then, within a certain pixel area, some convolutional operations are used. By doing those convolutional operations, a feature map is being created. We are unable to connect the neurons of the Conv2d to the dense layer of 1D datasets. So we convert those feature maps from a 2D array to a 1D array by using Flatten. After that, we join those 1D arrays by using a dense layer. Finally, we use activation softmax with another dense layer. Softmax converts those vector values to a probability distribution. The number of labels is set at 7 as there are 7 types of expressions in the dataset.

```
model.add(Flatten())

model.add(Dense(2*2*2*num_features, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(2*2*num_features, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(2*num_features, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(num_labels, activation='softmax'))
```

Fig 7.4.1: Adding dense and dropout layers to our CNN model

After defining the model, we compiled the model and set the learning rate as 0.001.Then, using the "fit" method, we started feeding the x and y coordinates from the training dataset into the model. We have set the epoch value to 100 to ensure high precision. The model's weighted graphs are then saved for later use.

Layer (type)	Output Shape	Param #
conv2d_104 (Conv2D)	(None, 46, 46, 64)	640
conv2d_105 (Conv2D)	(None, 46, 46, 64)	36928
batch_normalization_91 (Batc	(None, 46, 46, 64)	256
max_pooling2d_52 (MaxPooling	(None, 23, 23, 64)	0
dropout_91 (Dropout)	(None, 23, 23, 64)	0
conv2d_106 (Conv2D)	(None, 23, 23, 128)	73856
batch_normalization_92 (Batc	(None, 23, 23, 128)	512
conv2d_107 (Conv2D)	(None, 23, 23, 128)	147584
batch_normalization_93 (Batc	(None, 23, 23, 128)	512
max_pooling2d_53 (MaxPooling	(None, 11, 11, 128)	0
dropout_92 (Dropout)	(None, 11, 11, 128)	0
conv2d_108 (Conv2D)	(None, 11, 11, 256)	295168
batch_normalization_94 (Batc	(None, 11, 11, 256)	1024
conv2d_109 (Conv2D)	(None, 11, 11, 256)	590080
batch_normalization_95 (Batc	(None, 11, 11, 256)	1024
max_pooling2d_54 (MaxPooling	(None, 5, 5, 256)	0
dropout_93 (Dropout)	(None, 5, 5, 256)	0
conv2d_110 (Conv2D)	(None, 5, 5, 512)	1180160
batch_normalization_96 (Batc	(None, 5, 5, 512)	2048
conv2d_111 (Conv2D)	(None, 5, 5, 512)	2359808
batch_normalization_97 (Batc	(None, 5, 5, 512)	2048
max_pooling2d_55 (MaxPooling	(None, 2, 2, 512)	0
dropout_94 (Dropout)	(None, 2, 2, 512)	0
flatten_13 (Flatten)	(None, 2048)	0

```
_________________________________________________________________
dense_52 (Dense)             (None, 512)               1049088
_________________________________________________________________
dropout_95 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_53 (Dense)             (None, 256)               131328
_________________________________________________________________
dropout_96 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_54 (Dense)             (None, 128)               32896
_________________________________________________________________
dropout_97 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_55 (Dense)             (None, 7)                 903
=================================================================
Total params: 5,905,863
Trainable params: 5,902,151
Non-trainable params: 3,712
```

Fig 7.4.2: CNN model

7.5 Body Posture Detection

We utilized OpenCV again to get webcam imports and detect the body in the body posture section.In our posture detection method, the algorithm we are using is BlazePose GHUM 3D which is used by Mediapipe. We will make a module out of it to use in our work. Body key points are defined as landmarks in this library. We use it to get all the landmarks of the body. Then, using the findPose method of our pose module, we draw the line between those landmarks and connect them.

```python

    def findPose(self, img, draw=True):
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.pose.process(imgRGB)
        if self.results.pose_landmarks:
            if draw:
                self.mpDraw.draw_landmarks(img, self.results.pose_landmarks,
                                           self.mpPose.POSE_CONNECTIONS)
        return img
```

Fig 7.5: Landmark finding method design

Later on, we have used those landmarks to identify the angle and distance between those body key points which we later used to classify critical situations using different body postures of a patient.

```python
    def findAngle(self, img, p1, p2, p3, draw=True):

        # Get the landmarks
        x1, y1 = self.lmList[p1][1:]
        x2, y2 = self.lmList[p2][1:]
        x3, y3 = self.lmList[p3][1:]

        # Calculate the Angle
        angle = math.degrees(math.atan2(y1 - y2, x1 - x2)-math.atan2(y3 - y2, x3 - x2))
        if angle < 0:
            angle += 360

        if draw:
#            cv2.line(img, (x1, y1), (x2, y2), (255, 255, 255), 3)
#            cv2.line(img, (x3, y3), (x2, y2), (255, 255, 255), 3)
            cv2.circle(img, (x1, y1), 10, (0, 0, 255), cv2.FILLED)
            cv2.circle(img, (x1, y1), 15, (0, 0, 255), 2)
            cv2.circle(img, (x2, y2), 10, (0, 0, 255), cv2.FILLED)
            cv2.circle(img, (x2, y2), 15, (0, 0, 255), 2)
            cv2.circle(img, (x3, y3), 10, (0, 0, 255), cv2.FILLED)
            cv2.circle(img, (x3, y3), 15, (0, 0, 255), 2)
#            cv2.putText(img, str(int(angle)), (x2 - 50, y2 + 50),
#                        cv2.FONT_HERSHEY_PLAIN, 2, (0, 0, 255), 2)
        return angle
```

Fig 7.5.1: Angle finding method design

7.6 Experimental Setup

For the detection part, we have used the OpenCV python library to get the webcam feed. In the webcam image, we are printing a rectangular box around the face. This is done using the "Haarcascade frontal face" classifier. It detects faces in an image. The problem with this classifier is that it can detect faces only in grayscale images. That is why we convert the live feed image to a grayscale image and then apply the classifier. Then the expression is detected using the model we previously saved and printed along with the webcam feed using the "putText" method.

```
gray=cv2.cvtColor(full_size_image,cv2.COLOR_RGB2GRAY)
face = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
faces = face.detectMultiScale(gray, 1.3  , 10)

for (x, y, w, h) in faces:
        roi_gray = gray[y:y + h, x:x + w]
        cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray, (48, 48)), -1), 0)
        cv2.normalize(cropped_img, cropped_img, alpha=0, beta=1, norm_type=cv2.NORM_L2, dtype=cv2.CV_32F)
        cv2.rectangle(full_size_image, (x, y), (x + w, y + h), (0, 255, 0), 1)
        #predicting the emotion
        yhat= loaded_model.predict(cropped_img)
        cv2.putText(full_size_image, labels[int(np.argmax(yhat))], (x, y), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0), 2, cv2.LI
        print("Emotion: "+labels[int(np.argmax(yhat))])

cv2.imshow('Facial Expression', full_size_image)
cv2.waitKey()
```

Fig 7.6: Facial expression

For the posture part, we have used the same image from the live feed but converted it to RGB color, as the BlazePose algorithm detects body landmarks in RGB image only. After that, using the findAngle and findDistance methods, we calculated angles and distances between different body keypoints. Using our video dataset, we have calculated the accuracy and result of our proposed method.

In our facial expression dataset, there is 80% train data and 10% test data. On the other hand, in our posture detection method, we have used 113 videos and 100 images for both test and training purposes.

Chapter- 8

Experiment and Result Analysis

8.1 Stomach Ache

Stomach ache fall						
Sample video No.	Detected	Sitting	SideBending	Fall	Emotion	Comment
1	1	1	0.7	1	Fear	Late side bend detected
2	1	1	0.8	1	Fear+Sad	Sitting detected after fall
3	1	1	0	1	Fear	No side bend detected
4	1	1	0.6	1	Fear	Late side bend detected
5	1	0.8	0.8	1	Fear	late sittng detected
6	0.8	0.9	0.5	0.6	Fear	Sightly False detection
7	1	1	0.2	0.2	Angry	Slightly side bend and fall detected
8	1	0.8	0.5	1	Fear+Surprise	false sidebending and fall detected
9	1	0.7	0.8	1	Fear+Surprise	late side bend detected
10	1	0.7	0.7	1	Angry fear	wrong sitting detected
11	1	1	0.9	1	Sad+Fear	late side bend detected
12	1	1	0.5	1	Sad+Fear	Early Fall Detected
13	1	1	0	1	none	No side bend detected
14	1	1	0	0	Sad + angry	Wrong camera position
15	1	1	0.1	0	Sad	Wrong camera position
16	1	1	0	1	none	No side bend detected
17	1	1	1	1	Sad	Perfect
18	1	1	0	1	Sad	No side bend detected
19	1	1	1	1	Sad	Perfect
20	1	1	1	1	Angry	Perfect
21	1	1	0.5	0.5	Sad	Early Side bend and late fall detected
22	1	1	0.7	0	Pain	No fall detected and late side bend detected
23	1	1	1	0	Sad	No fall detected
24	0.5	1	1	1	Sad + angry	Slightly chest pain fall detected
25	1	1	1	0.5	Sad + pain	Slightly fall detected
26	1	1	1	1	sad	Perfect
27	1	1	1	1	sad	Perfect
28	1	1	1	1	sad	Perfect
29	0.7	1	0.5	0	pain	Slightly chest pain fall detected
30	1	1	1	1	sad	Perfect

Fig 8.1: Stomach ache sequence data result

We have used 30 sample videos of stomach aches. All those videos have been taken from different angles and lighting conditions. After feeding those videos into our system, in 28 cases, we got stomach aches as an output, and for the sequence analysis, we found that 80% of the time sitting is identified as the correct posture. We had some accuracy problems with side bending. In 30% of cases, it was predicted correctly. On the contrary,it predicted sitting or falling as body posture in alternative cases. For the fall sequence, we got a 70% correct result, and for the other 30% of the cases, it detected sitting and side bending as body posture.

In the vast majority of situations, the system recognizes sadness as an output of the expression portion, but our system is unable to detect any expression in only two example video experiments.

Fig 8.1.1: Confusion matrix of stomach ache

8.2 Chest Pain

Chest Pain			
Sample Video No	Detected	Emotion	Comment
1	1	Sad	Perfect
2	1	Sad + angry	Perfect
3	1	Pain	Perfect
4	1	Sad	Perfect
5	1	Sad + angry	Perfect
6	1	Pain	Perfect
7	1	Sad + angry + Fear	Perfect
8	1	Sad + angry	Perfect
9	1	Angry + Fear	Perfect
10	1	Sad + Fear	Perfect
11	1	Sad + Pain	Perfect
12	1	Sad	Perfect
13	1	Sad + angry	Perfect
14	1	Sad + angry	Perfect
15	1	Pain	Perfect
16	1	Pain	Perfect
17	1	None	No emotion detected
18	1	Pain + Fear	Perfect
19	1	Pain	Perfect
20	0	None	False Reading
21	1	Pain	Perfect
22	1	Angry	Perfect
23	0	Pain	No chest pain detected
24	1	Pain	Perfect

Fig 8.2: Chest pain data result

For our 24 sample chest pain videos, we got 22 correct predictions and 22 expressions. In the case of the two videos, faces were not detected successfully, which is why expressions could not be detected by our system.

Chest pain fall							
Sample video No	Detected	Sitting	Head Leaning	Shoulder Bending	Fall	Emotion	Comment
1	1	1	0	1	0	None	No head leaning detected
2	1	1	1	0	1	None	No shouder bending detected
3	1	1	1	0	1	None	No shouder bending detected
4	1	1	1	1	1	None	Perfect
5	1	1	1	1	1	None	Perfect
6	1	1	1	1	0	None	No fall detected
7	1	1	1	0.7	0.2	None	Wrong Camera Position
8	1	1	1	0.5	0.5	None	Late shoulder bending and fall detection
9	1	1	1	1	0	None	No fall detected
10	1	1	0	1	1	None	Early Shoulder bending
11	1	1	0	1	0	None	Early Shoulder bending
12	1	1	1	1	0.6	None	Late fall detected
13	1	1	0	0.7	0	None	Early Shoulder bend and no fall detected
14	1	1	1	1	0	None	No fall detected
15	1	1	0	1	1	None	No head leaning detected
16	1	1	1	1	1	None	Perfect
17	1	1	1	0	1	None	Subject did not bend shoulder
18	1	1	1	1	1	None	Perfect

Fig 8.2.1: Chest pain sequence data result

In our chest pain fall sequence detection, we were able to detect all the sitting positions perfectly. While testing those 18 videos, 3 of the head-leaning sequences were detected as sitting, and the remaining 2 were detected as standing and falling. The rest of the 13 were detected successfully as head leaning. In our shoulder bending sequence, 12 cases were successfully detected while the rest of the cases were found as head leaning and fall sequences. Moreover, for our fall sequence, our model performed lower than we expected as only 9 cases were perfectly detected, and the rest were found to be shoulder bending. Finally, for the expression part, no face was not detected during the test session. Therefore, expressions were not able to be detected from the sample videos.

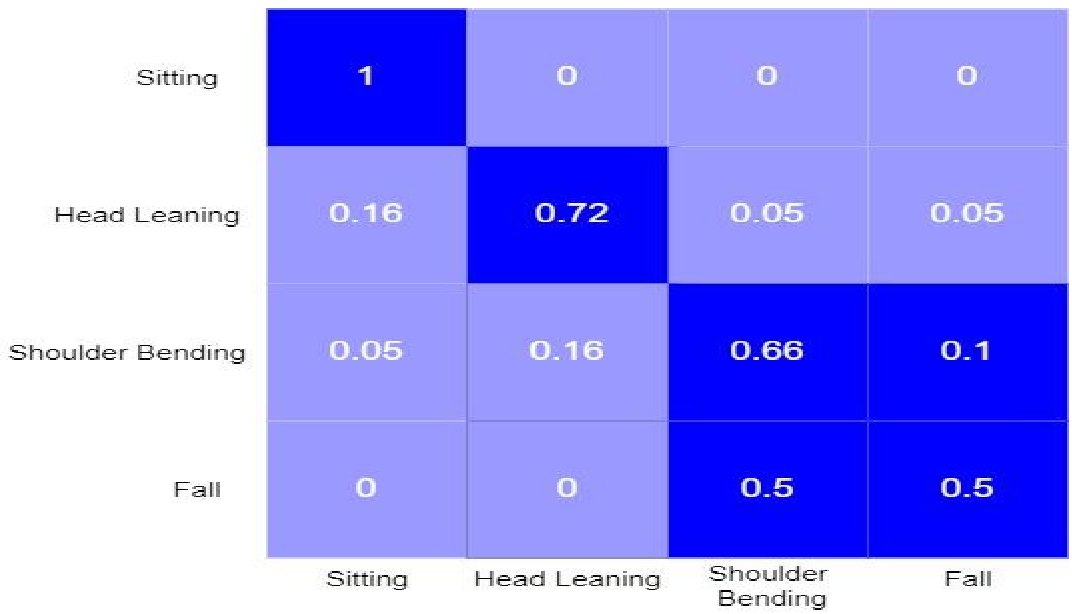


Fig 8.2.2: Confusion matrix of chest pain

8.3 Headache

Headache			
Sample Video No.	Detected	Comment	Emotion
1	0.8	Late detection	Pain
2	1	Perfect	Sad+fear
3	0.4	Late & False detected	None
4	1	Perfect	None
5	1	Perfect	Pain
6	1	Late emotion detection	Sad+pain
7	1	Perfect	Sad
8	1	No face detected	None
9	1	No face detected	None
10	1	Perfect	Pain
11	1	Perfect	Sad
12	1	Perfect	Pain
13	1	Perfect	Sad
14	0.5	Late detection	None
15	0	Not detected	None
16	0	Not detected	Pain
17	1	Perfect	Fear
18	1	Perfect	Fear + Angry
19	1	Perfect	Fear + Angry
20	1	Perfect	Sad
21	1	Perfect	Sad + Angry
22	1	Perfect	Pain + Angry
23	1	Perfect	Sad + Angry
24	0	Not detected	Sad + Pain
25	0.5	Late detection	Sad

Fig 8.3: Headache data result

In our 25 sample videos of headaches, in 19 cases, expression and headache were perfectly detected.

8.4 Falling from bed while sleeping

Bed Fall				
Sample Video no	Sleeping	Fall	Emotion	Comment
1	1	1	None	No Face Detected
2	1	1	None	No Face Detected
3	1	1	None	No Face Detected
4	1	1	None	No Face Detected
5	1	0.7	None	Late fall detected
6	1	0	None	No fall detected
7	1	0.2	None	Slightly fall detected
8	1	0	None	No fall detected
9	1	0	None	No fall Detected and wrong camera position
10	1	0.5	None	Early fall detected and wrong camera position
11	1	0	None	No Fall Detected and wrong camera position
12	1	1	None	No Face Detected and wrong camera position
13	0.5	0	None	No Fall Detected and wrong camera position
14	0.7	0.5	None	collision between sleeping and fall detection and wrong camera position
15	0.5	0.4	None	Early fall detected and wrong camera position
16	0.8	1	None	mostly perfect

Fig 8.4 : Falling from bed while sleeping sequence data result

During the test session of the falling from bed while sleeping sequences, videos were taken from different angles which result in no face detection. Therefore, the expressions are unable to be extracted. On the other hand, for the sleeping sequence, rather than 4 cases, all cases are successfully detected whereas only 6 cases were detected as fall for the fall sequence.

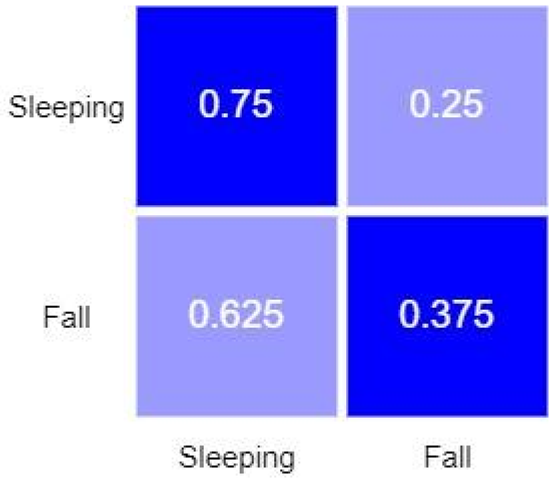


Fig 8.4.1: Confusion matrix of Falling from bed while sleeping

8.5 Accuracy Analysis

Our facial expression recognition CNN model achieved 66% accuracy. On the other hand, the posture detection method achieved 86% accuracy. Those accuracy levels were tested against various camera positions and light conditions from which we found that our model performs better in daylight and when the background is of a light color. Again, it performs less than our expectations in low light and dark background situations as they are a bit more challenging. We trained the CNN model for 100 epochs. In terms of increasing the accuracy of CNN, we can train it more using medical patients' facial expression datasets. After testing our system in live feed we came to a conclusion that if we set the camera on an angle of 45 degrees only then we get a precise outcome for the scenario. In our cases of Chest Pain an average 72% fall sequences detected successfully and in case of Stomach Ache average 60% fall sequences were detected perfectly. Lastly, falling from bed while sleeping was detected only 56% in our system. All those accuracy can be increased by training more data in our model.

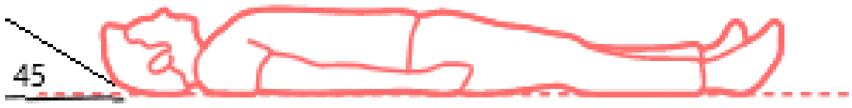


Fig 8.5 : Input live data from 45° angle

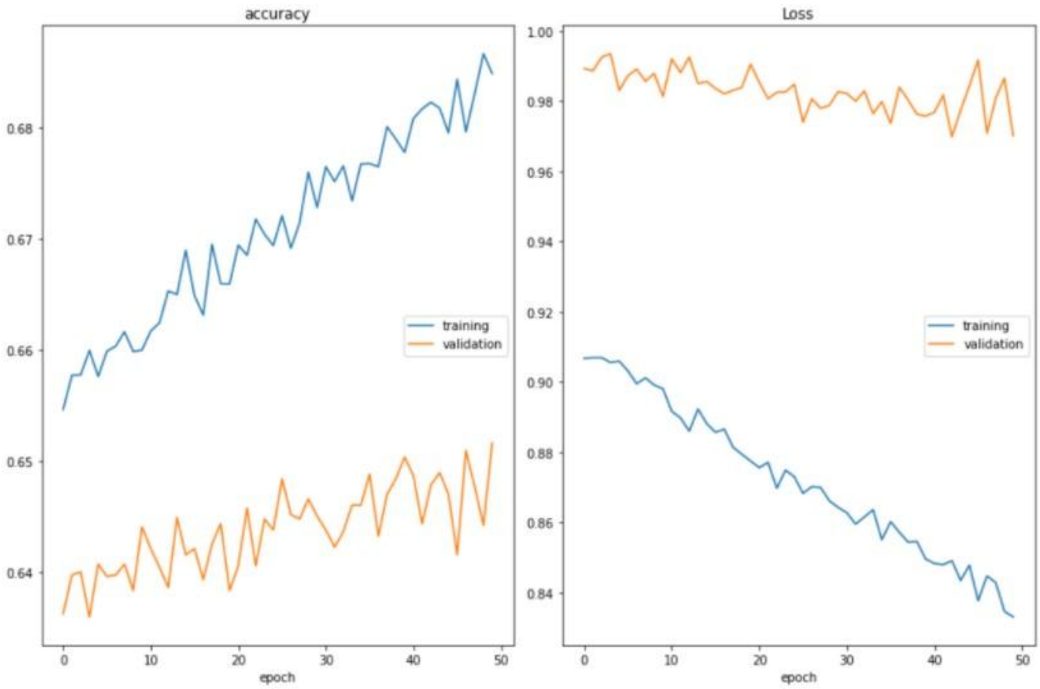


Fig 8.6: Training vs Validation after 50 epochs

Chapter 9

Future Work & Conclusion

9.1 Future scope of work

If it is complete, it is not perfect. No level of improvement is ever enough. From time to time, maintenance and development are necessary parts of a good system. The same can be said for our A Hybrid Approach to Determine Patients' Critical Situations using Expression and Posture with Convolutional Neural Networks and Blazepose Algorithms. In our case, we could only classify four critical situations (stomach ache, headache, chest pain, falling from bed whilst sleeping).

In the future, many more critical situations could be classified to make our system more efficient. With the help of Blazepose and CNN, we created sample videos of ourselves considering the postural stages the patients express. So if we use actual medical data sets and apply deep learning, the accuracy level would be much higher, not to mention more precise.

9.2 Conclusion

Posture detection is gaining more and more popularity every day. Starting from virtual reality to making live emojis. So many major companies including Apple, HTC, Facebook, and many more are investing a lot of assets in this sector. It seems like what we imagined our future would be is happening right in front of our eyes. Science fiction is not just stuck in movies and books anymore, rather it is turning into reality. Things that used to be unimaginable and full of awe are becoming habitual.

Now, regardless of the few dismissive facts, the patient's facial expression and posture detection system is more supreme than we might think it is. The novelty of combining these two into one is giving us results that are stunning. Now we are able to monitor patients easily without a sweat. It would not just save human resources but also will be significantly accurate as human error will not be a factor. Not to mention we can use that human resource in other sectors which would end up being way more beneficial to us plus improve efficiency at the same time. Moreover, we can study the data we have and train our systems to further enhance our current state of posture detection as with the increment of data our accuracy will also improve. Thus, becoming more and more reliable with the passing of time.

Reference

[1] Ekman, P. & Friesen, W. (1974). Detecting deception from the body and face, Journal of Personality and social psychology, 29, pp. 288-298.

[2] Message from the president; Criticon Bangladesh 2018. Bangladesh Society of Critical Care Medicine; 2018.

[3] Mohiuddin AK. An extensive review of patient health-care service satisfaction in Bangladesh. Adesh Univ J Med Sci Res 2020;2(1):5-16.

[4] Number of critical patients surging, The daily star https://www.thedailystar.net/frontpage/news/number-critical-patients-surging-2059989

[5] An analysis of structure, https://www.banglajol.info/index.php/BCCJ/article/view/40761/30610

[6] Poncette AS, Spies C, Mosch L, Schieler M, Weber-Carstens S, Krampe H, Balzer F. Clinical requirements of future patient monitoring in the intensive care unit: qualitative study. JMIR medical informatics. 2019;7(2):e13064. DOI:10.2196/13064

[7] Khanna AK, Hoppe P, Saugel B. Automated continuous noninvasive ward monitoring: future directions and challenges. Critical Care. 2019 Dec;23(1):1-5. DOI:10.1186/s13054-019-2485-7

[8] Facial expression monitoring system for predicting patient's sudden movement during radiotherapy using deep learning,
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7484824/

[9] Patient Posture Monitoring System Based on Flexible Sensors,

https://www.researchgate.net/publication/314976843_Patient_Posture_Monitoring_System_Based_on_Flexible_Sensors

[10] Papers with Code - Facial Expression Recognition. (2021). Retrieved 1 June 2021, from https://paperswithcode.com/task/facial-expression-recognition

[11] Htike, K., & Omran Khalifa, O. (2015). Human posture recognition: Methodology and implementation. Retrieved 1 June 2021, from https://www.researchgate.net/publication/283103054_Human_posture_recognition_Methodology_and_implementation

[12] R. Walecki, O. Rudovic, V. Pavlovic, B. Schuller, and M. Pantic, "Deep structured learning for facial action unit intensity estimation," 2017, doi: 10.1109/CVPR.2017.605.

[13] M. I. Georgescu, R. T. Ionescu, and M. Popescu, "Local learning with deep and handcrafted features for facial expression recognition," IEEE Access, 2019, doi: 10.1109/ACCESS.2019.2917266.

[14] M. Pantic, L. Rothkranz, Expert system for automatic analysis of facial expression, Image and Vision Computing 18 (11) (2000) 881–905.

[15] Y. Tian, T. Kanade, J. Cohn, Recognizing action units for facial expression analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2) (2001) 97–115.

[16] J. Chen, Z. Chen, Z. Chi, and H. Fu. Emotion recognition in the wild with feature fusion and multiple kernel learning. In Proceedings of the 16th International Conference on Multimodal Interaction, pages 508–513. ACM, 2014.

[17] K. Sikka, K. Dykstra, S. Sathyanarayana, G. Littlewort, and M. Bartlett. Multiple kernel learning for emotion recognition in the wild. In Proceedings of the 15th ACM on International conference on multimodal interaction, pages 517–524. ACM, 2013.

[18] B. Sun, L. Li, T. Zuo, Y. Chen, G. Zhou, and X. Wu. Combining multimodal features with hierarchical classifier fusion for emotion recognition in the wild. In Proceedings of the 16th International Conference on Multimodal Interaction, pages 481–486. ACM, 2014.

[19] C. Darwin, The Expression of the Emotions in Man and Animals, John Murray, London, 1872.

[20] P. Ekman and W. Friesen, Facial Action Coding System, Mountain View, CA, USA: Consulting Psychologists Press, vol. 1, 1978.

[21] Coma - Diagnosis and treatment - Mayo Clinic. (2021). Retrieved 2 June 2021, from https://www.mayoclinic.org/diseases-conditions/coma/diagnosis-treatment/drc-20371103

[22] Foley GN, Gentile JP. Nonverbal communication in psychotherapy. Psychiatry (Edgmont). 2010;7(6):38-44

[23] Cherry, K., & Gans, S. (2021). How to Read Body Language and Facial Expressions. Retrieved 2 June 2021, from https://www.verywellmind.com/understand-body-language-and-facial-expressions-41472

[24] Carrier, P. and Courville, A., 2022. *fer2013*. [online] Kaggle.com. Available at: <https://www.kaggle.com/deadskull7/fer2013> [Accessed 15 January 2022].

[25] Pose. (2022). Retrieved 16 January 2022, from https://google.github.io/mediapipe/solutions/pose

[26] DeepFace - The Most Popular Open Source Facial Recognition Library - viso.ai. (2022). Retrieved 16 January 2022, from https://viso.ai/computer-vision/deepface/

[27]FER-2013. (2021). Retrieved 1 October 2021, from https://www.kaggle.com/msambare/fer2013

[28] opencv/haarcascade_frontalface_default.xml at master · opencv/opencv. (2022). Retrieved 16 January 2022, from https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml

[29] Kaggle.com. 2022. *Challenges in Representation Learning: Facial Expression Recognition Challenge | Kaggle*. [online] Available at: <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data> [Accessed 19 January 2022].

Appendix A.

Result:

1. Thesis data