Recognition of Student Emotions in an Online Education System

Abstract:

Online education system was developed due to the Covid-19 pandemic. The core idea of this paper is to map the connection between teaching practices to student learning in an online environment. Face to face evaluation techniques are fairly quick and easy for formative assessments to check student understanding in existent environment. Prevailing studies illustrate that a person's facial expressions and emotions are closely related. In order to make the teaching-learning process more effective, teachers usually collect day to day feedback from the students. This feedback can be used to improve teaching skills and make the process more interactive. In a virtual learning mode, there is a need to identify and understand the emotions of people. Constructive information can be extracted from online platforms using facial recognition algorithms. An online course connected with students is used for examination; the results have shown that this technique performs strongly.

SECTION I.

Introduction

There is a necessity for teachers to understand the efficiency of students in an online environment. This issue doesn't occur in an offline scenario, where the teacher can clearly observe reactions and expressions of students to determine the extent of their understanding. The idea proposed in this paper provides support to teachers in adapting their teaching practices to match students' interests, progress and learning. Numerous researches have recommended that the intention of the expressive state of people influences (directly or indirectly) the learning process. Emotions are powerful feelings associated with every situation and hold a prominent share in any interpersonal communication. Emotion recognition can be performed using different features, such as face, speech and even text. There are two major categories in modes of communication – verbal and nonverbal. Online education system predominantly contains non-verbal communication (from students to the teacher). In an e-learning environment, in particular, students' facial expressions can be leveraged to understand their emotions. As a consequence, it becomes essential to interpret a student's frame of mind by means of some fundamental facial indicators. Facial expressions are crucial to estimate an individual's internal feelings, and are one of the most direct ways of expressing emotions. Hence, they have an important role in non-verbal communication. To analyze the emotions of students, specific deep learning algorithms can be integrated to virtual meeting platforms. This framework makes it possible for monitoring the emotions of students in a real time online education system. It ensures that the feedback expressed through facial expressions is made available to teachers in a timely manner.

SECTION II.

Literature Survey

The term 'facial emotion recognition' refers to categorization of facial features into one of the known emotions which are happiness, surprise, anger, sad, hatred, fear and contempt. Face detection and recognition have come from the 1960. [17], [18] The first proof of the concept that computers can actually detect faces was given by Woodrow Wilson Bledsoe. Face detection is quite

different from face recognition [1] Face recognition is recognizing individuality from face image. Algorithms to answer a problem are broken down in existence of composite variations. Deep learning algorithms [1], [15] are used for unconstrained face detection methodologies. One of the principal techniques in object detection is the universal bounding box regression technique. Through a face-detection system, the aim is to distinguish faces from other objects. The network so trained would then suggest candidate bounding boxes on an input image by classifying the convolution swatches as 'face' or 'not-face' [6]. Face recognition includes ascertaining the existence of face and formatting its position in the image. Face detection comes with its own set of challenges [7]. There are diverse statistical models that cater to each of the different problems that arise in face detection including (but not limited to) variety in expressions, different orientations, occlusion or partially hidden faces, variable illumination, complex or plain backgrounds, and different resolution of images, etc. Fig. 1 depicts various categories of emotions such as happiness, anger, sadness, neutral, and drowsiness etc. [2], [3].

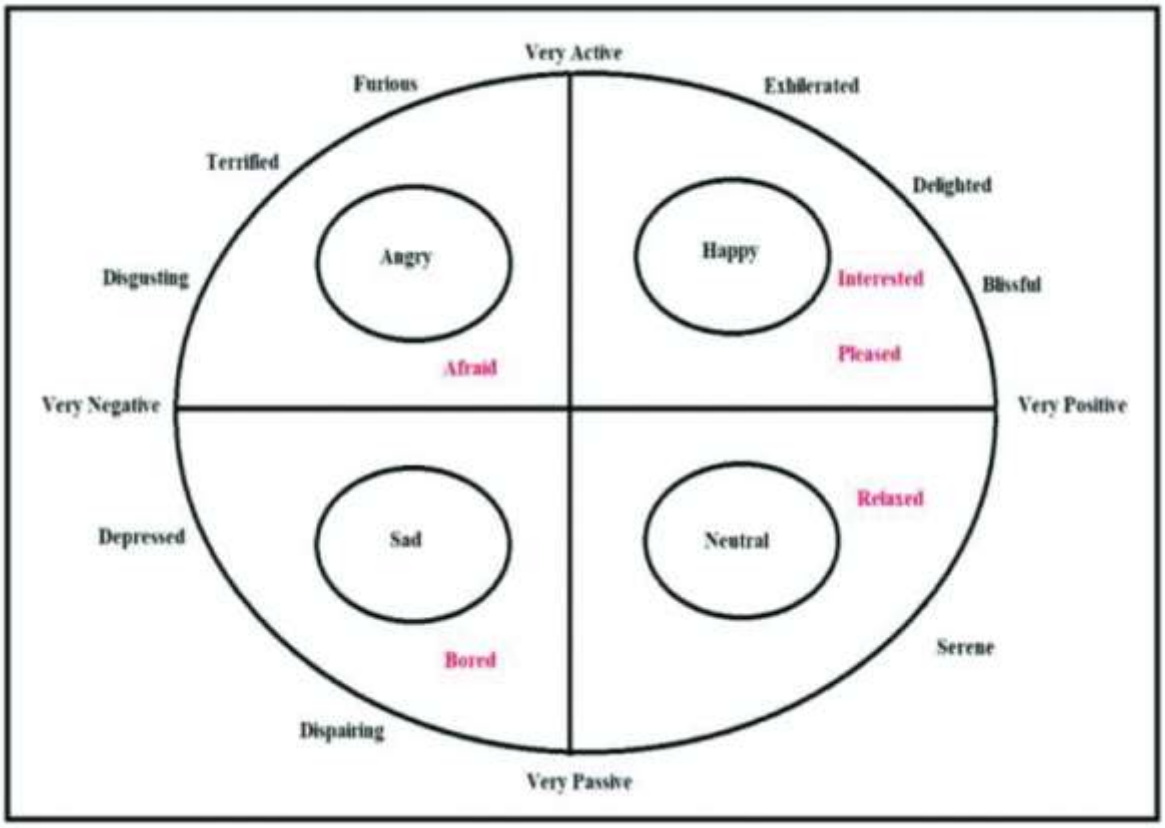

Fig. 1. Depiction of various human emotions

Machine learning is one of the application domains of 'Artificial Intelligence' [8], [12] that allows us to give a system the capacity to learn and improve by itself over time, without having to actively educate it. Deep learning is a subset of 'Artificial Intelligence' that is built on the branch of machine learning [15]. It is essentially a machine learning class that uses a large number of nonlinear processing units to do extraction of features and modification. Each succeeding layer takes the preceding layer's output as input. When there are a large number of inputs and outputs, deep learning algorithms are used.

SECTION III.

Proposed System

The objective of this paper is to use MTCNN or Multi-task Convolution Networks for face detection and emotion recognition. An image containing students' faces should be given as an input to the algorithm. The algorithm uses deep learning techniques to construct bounding boxes around each face and then indicates the probabilities for the face being in a particular emotional state. The algorithm also generates a distribution plot indicating the percentage of students in each emotional state (such as happiness, anger, sadness, neutral, and drowsiness etc.), as the algorithm perceives it. The 'manual' nature of the scheme requires the instructor to take a picture of the class and feed it as an input to the algorithm. This method is tedious and not so convenient in practice. To automate the task of capturing the image and executing the algorithms, the following framework is proposed — Integrate MTCNN in online meeting platforms, as a browser extension or an additional feature in the application itself, to enable deep learning algorithms to use students' images directly from the platform itself. This would reduce the instructor's efforts to do the same. Since the implementation of this idea requires working knowledge of APIs and software development, it is complicated to realize in practice. This paper describes the working and execution of MTCNN, which is the key concept involved in face detection and emotion recognition. MTCNN was used in the proposed model as it gave acceptable real time performance, and the expected variation of scale in this use-case was largely invariant. MTCNN, as the name suggests, leverages the information correlation between two sub-tasks of different categories in face recognition [19]. The two tasks are facial detection and face alignment. One task can be auxiliary to the other which is the primary task to improve its performance and accuracy. MTCNN combines these tasks in a cascaded CNN. The CNN consists of three stages, [4], [5] viz. first stage that produces candidate bounding boxes, second stage that rejects non face windows or boxes, and finally the last stage that refines results and outputs facial landmark positions. In this use-case facial landmark positions are identified and used to detect emotional states of students. The CNN architecture in MTCNN is lightweight to ensure realistic runtime performance. The three stages of refinement are principally non-maximum suppression (NMS), bounding box regression progressing towards a more refined output, with facial landmark recognition in the last step. The authors of MTCNN have named these stages as P-Net or proposal-network where multiple swatches or candidate bounding boxes are proposed, R-Net or refine-network [14], [15], [19], wherein large number of swatches that are not face bounding boxes are rejected; and the O-Net or output-network which outputs the final bounding boxes with facial landmarks. There is also a pre-processing stage that creates a pyramid of resized images to take into account scale aspects. Once the input image is run through MTCNN the list of bounding boxes of faces present in the image with their dimensions and position can be extracted. This image is further processed using the image-cropping pipeline.

Step 1: To recognize faces of varied sizes, an image pyramid is constructed. For each copy, a kernel with 12*12 dimensional [9] stages is accessible. The kernel can be used to scan each portion of the image to identify a person's face. The algorithm begins scanning the image from the top left corner, i.e. (0, 0). P-Net (Proposal Net) receives this portion of the image and gives the dimensions of a bounding box if any exists. However, there is still a large number of bounding boxes remaining, many of which might overlap. NMS (Non-Maximum Suppression) is a technique for lowering the

number of bounding boxes. NMS is carried out in this program by first sorting the bounding boxes according to their confidence, or score.

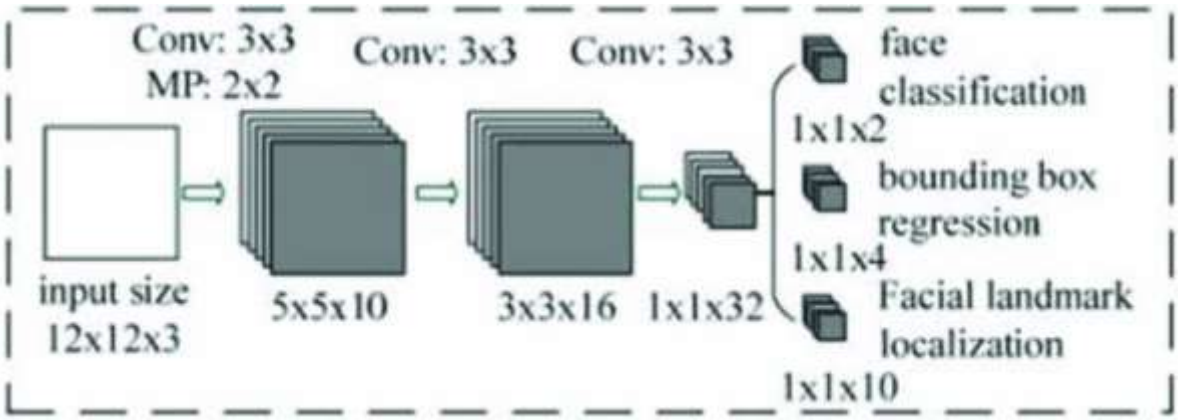

Fig. 2. Proposal Net (P-Net) Source: Google Images

Step 2: The information from the P-Net is compared with R-Net (Refine Network), the next layer of CNN [16], which is a fully connected, complicated CNN that rejects majority of the frames that do not include faces and if the entire face is not included in the bounding box, it copies the picture in the bounding box to a new array and fill the remaining blanks with zeros. This process of filling zeros is known as Padding. As a next step, boxes with low confidence values are deleted using NMS. The following image Fig. 3 shows the MTCNN Architecture for a Refine network.

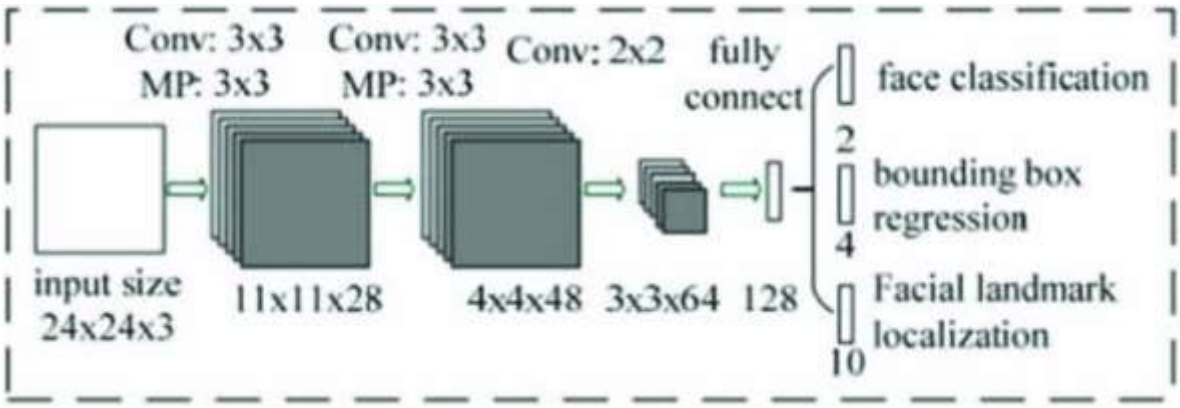

Fig. 3. Refine Net (R-Net) Source: Google Images

Step 3: The output network is the third and final step in this process. O-Net is a very complicated and powerful process. The outputs of the facial landmark positions are detected from the given image. As explained earlier, NMS (Non Maximum Suppression) is used to remove the boxes with low confidence values. As a result of this entire process, the final output is an image with one bounding box for every face in the image. The following image Fig. 4 is a visual representation of the O-Net.

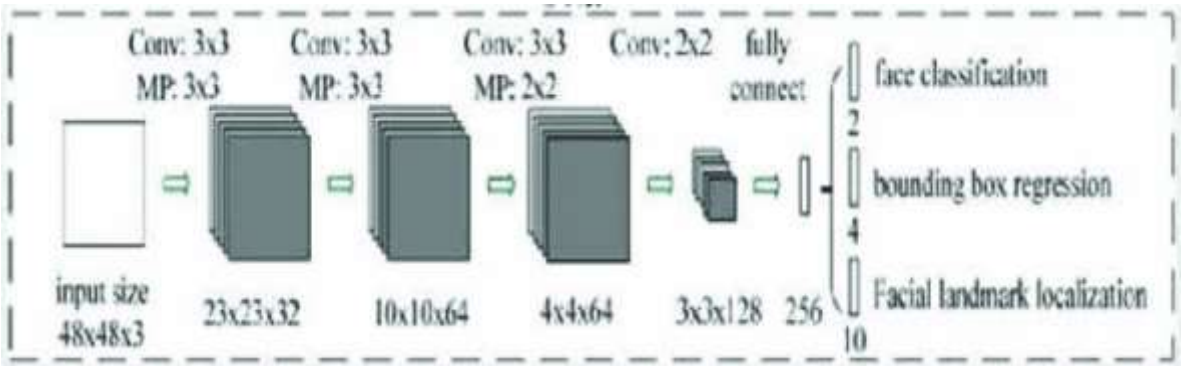

Fig. 4. Output Net (O-Net) Source: Google Images

SECTION IV.

Identifying Facial Landmarks

There are 68 landmark points in a human face. They can be demarcated using the dlib package. In general, regression trees are used to estimate these landmarks based on range of pixel intensities [10], [11].

The above picture Fig. 5 depicts the facial landmarks in a human face. Facial analysis points visualize the feature points. The process consists of three stages of convolution networks [13], [16] that can recognize faces and landmark location such as eyes, nose, and mouth.

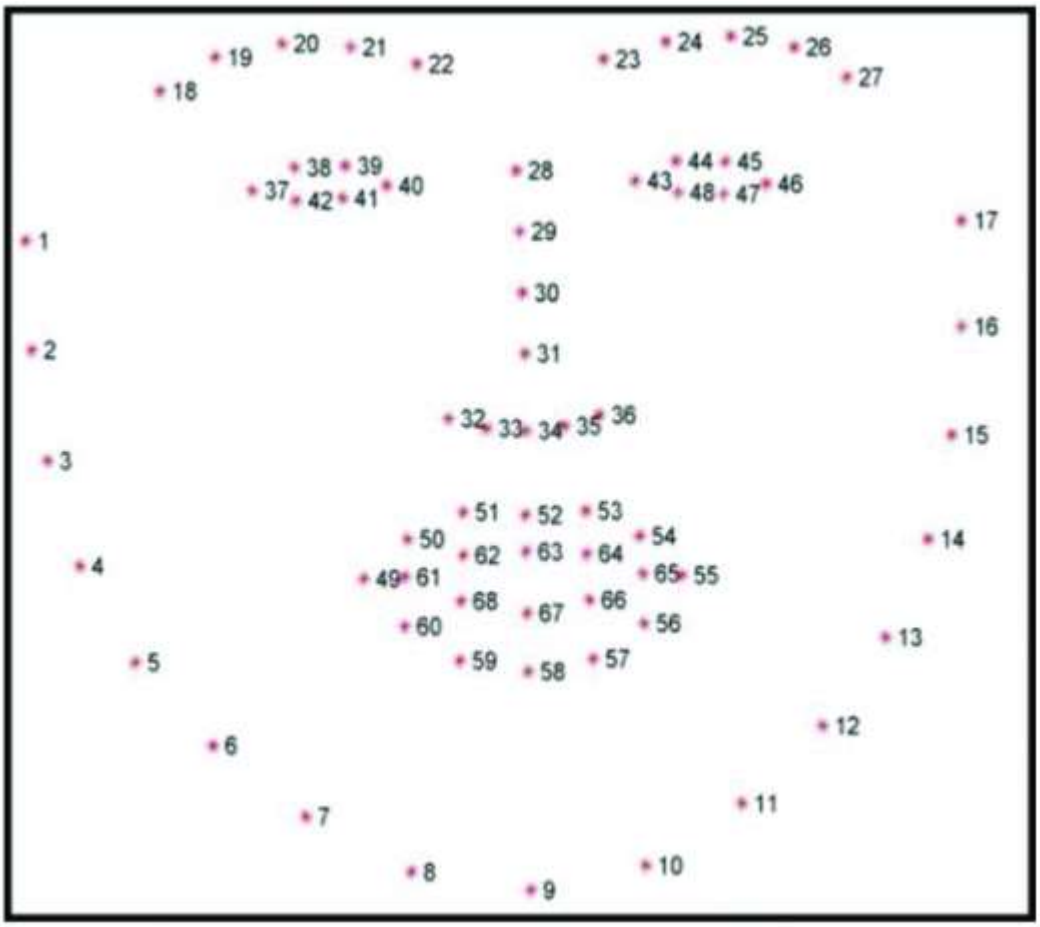

Fig. 5. Facial Landmark Points Source: Google Images

These points can be used to identify various facial features. For instance, according to Fig. 5, the right eye is located between the landmarks 37 and 42, while the left eye is located between landmarks 43 and 48, and the mouth is located between landmarks 49 and 68.

The facial land mark points are used to identify the yawn and Eye blink rate. Eye blink can be calculated by using the Eye aspect ratio between the vertical and horizontal eye land marks.

SECTION V.

Results

For accurate results, real time images were used as an input to the algorithm. Specifically, the input to this model is an image of students taken in a real time virtual classroom. As apparent from Fig. 6, most students seem either happy or neutral. These are exactly the emotions captured in the output probability distribution plot (Fig. 8) of the algorithm. Therefore, it can be concluded that the model works satisfactorily well in actual environments.

All the faces were recognized and marked with boxes as displayed in Fig. 7., with clear labeling of each emotion along with its estimated probability. These probabilities can be used to find out the dominant emotion in a person as detected by the algorithm. For instance, record 1 of the output is

{‘box’: (945, 74, 286, 286), ‘emotions’: {‘angry’: 0.02, ‘disgust’: 0.0, ‘fear’: 0.01, ‘sad’: 0.04, ‘surprise’:0.0, ‘neutral’: 0.93}. This means that the probabilities of the person, in the facial image detected in a box with pixel coordinates (945, 74, 286, 286), being angry, sad, and neutral are 0.02, 0.04 and 0.93 respectively. Similarly, other records indicate the probabilities of the face (enclosed within the bounding box of given coordinates) being in each emotional state.

Fig. 6. Image used as input to the algorithm

[{'box': (945, 74, 286, 286), 'emotions': {'angry': 0.02, 'disgust': 0.0, 'fear': 0.01, 'happy': 0.0, 'sad': 0.04, 'surprise': 0.0, 'neutral': 0.93}}, {'box': (50, -10, 307, 307), 'emotions': {'angry': 0.03, 'disgust': 0.0, 'fear': 0.03, 'happy': 0.02, 'sad': 0.03, 'surprise': 0.64, 'neutral': 0.26}}, {'box': (645, 48, 111, 111), 'emotions': {'angry': 0.02, 'disgust': 0.0, 'fear': 0.01, 'happy': 0.56, 'sad': 0.04, 'surprise': 0.01, 'neutral': 0.36}}, {'box': (601, 336, 234, 234), 'emotions': {'angry': 0.06, 'disgust': 0.0, 'fear': 0.05, 'happy': 0.49, 'sad': 0.27, 'surprise': 0.01, 'neutral': 0.12}}, {'box': (-16, 186, 304, 304), 'emotions': {'angry': 0.38, 'disgust': 0.02, 'fear': 0.04, 'happy': 0.11, 'sad': 0.33, 'surprise': 0.01, 'neutral': 0.12}}]

Fig. 7. Emotion labels associated to each bounding box

Fig. 8. is one of the outputs of the algorithm, which is a bar plot of the probability distribution of emotions of students present in the image. Each bar indicates the percentage of the total class strength corresponding to that emotion. Using this plot, the overview of emotional states of the class can be easily interpreted.

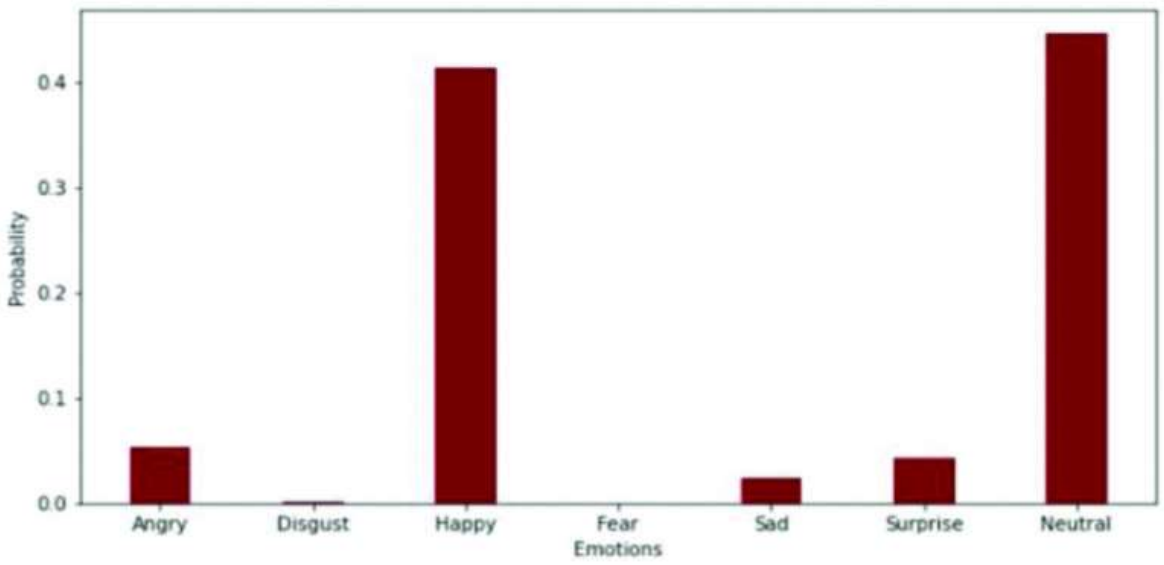

Fig. 8. Probability Distribution Plot

According to Fig. 8., majority of the students are recognized to be either happy or neutral. Though faces in the figure seem more happy than neutral, the difference in the output from the expected result can be explained by the fact that many emotions can be interpreted from a face at an instant of time. Emotions showcased on the face can be labeled according to the probable emotions determined by the features. The overall emotional state of the image takes the sum of probabilities of dissimilar emotions in each face into consideration. The results of this experiment showcase that this model is suitable for emotion detection in an online educational system.

SECTION VI.

Conclusion

To smoothen the progress in a smart virtual learning system, models which can understand emotions and provide a detailed perspective about the subtlety and complexity of facial expressions can be used. In this research a framework to evaluate students' emotions based on their facial expressions is developed. This was possible by analyzing the scenario of a virtual platform using a compressed deep learning model based on the MTCNN architecture. From the perspective of computer simulation, MTCNN performs well in terms of meeting quality requirements and runtime performance. Runtime performance is one of the concerns as the system had to perform in real time.