iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.43205

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

Emotion and Gesture detection

Swati Raman[1], Sanchita Patel[2], Surbhi Yadav[3], Dr. Vanchna Singh[4]
[1, 2, 3]*Student, Computer Science Engineering, Inderprastha Engineering College, Ghaziabad, Uttar Pradesh,India*
[4]*Associate Professor, Applied Sciences and Humanities, Inderprastha Engineering College, Ghaziabad, Uttar Pradesh, India*

Abstract: ***Machine learning algorithms have removed the constraints of computer vision. Researchers and developers have developed new approaches for detecting emotions making it possible to predict the behaviour and consecutive actions of human beings. As machine learning methods make use of GPUs' massive computation capability, these models' image processing skills are well suited to real-world issues. Computer vision has moved from a niche field to a variety of other fields, including behavioural sciences. These algorithms or models are utilised in a wide range of real-world applications, including security, driver safety, autonomous cars, human-computer interaction, and healthcare. Due to the emergence of graphics processing units, which are hardware devices capable of doing millions of computations in seconds or minutes, these models are constantly changing. Technologies like augmented reality and virtual reality are also on the rise. Robotic vision and interactive robotic communication are two of their most intriguing uses. Both verbal and visual modalities can be used to identify human emotions. Facial expressions are an excellent way to determine a person's emotional state. This work describes a real-time strategy for emotion and gesture detection. The fundamental idea is to use the MediaPipe framework which is based on real-time deep learning, to generate critical points. Furthermore, A series of precisely constructed mesh generators and angular encoding modules are used to encode the generated key points. Finally, by assessing failure instances of existing models, we are evaluating the applicability of emotion and gesture detection from our model.We are using models such as Random Forest(RF),logistic regression(LR),Gradient Classifier(GR) and Ridge classifier(RC). Real-time inference and good prediction quality are demonstrated by the suggested system and architecture.***

Keywords: Body Landmarks, MediaPipe, Prediction, Accuracy, Real-time on-device Tracking, Recognition.

I. INTRODUCTION

Machine Learning is primarily concerned with the research of algorithms that improve with the use of data and experience. Machine learning is divided into two phases: training and testing. Machine Learning provides a useful platform in the medical area for quickly resolving a variety of difficulties. Machine Learning can be classified into two types which are supervised and unsupervised. We tend in supervised learning to frame a model with the help of well-labelled data. Unsupervised learning models, on the other hand, learn from unlabeled data. Face expression recognition based on hand motions is analysed using a variety of methodologies, including machine learning.

Artificial Neural Networks, Artificial Intelligence, Computer Vision and so on. Any recognition system can be designed by following steps :

1) A face emotion and hand gesture database should be developed
2) Facial expression and hand gesture tracking as well as its position inside an image sequence
3) Collection of face and hand information for features identification
4) Human emotion recognition and classification

II. LITERATURE REVIEW

1) Z.Ren, J.Meng, Yuan J. Depth Camera Based Hand Gesture Recognition and its Application in human-computer-Interaction. In Processing of the 2011 8th International Conference on Information, Communication and Signal Processing (ICICS). Singapore. 2011. "Of various Human-Computer-Interactions (HCI), hand gesture-based HCI might be the most natural and intuitive way to communicate between people and machines. Its intuitiveness and naturalness have spawned many applications in exploring large and complex data, computer games, virtual reality, etc."
2) S.Rautaray S, Agrawal A. Vision-Based Hand Gesture Recognition for Human-Computer Interaction: A Survey. Springer Artificial Intelligence Review. 2012. DOI: https://doi.org/10.1007/s10462-012-9356-9."Recognition of hand gestures based on vision is used in the design and development of interfaces such as that of automobile users. The video-based gesture recognition problem is not trivial and poses problems due to intra and inter-person variations in the movement of human hand gestures."

3) Lugaresi C, Tang J, Nash H, McClanahan C, et al. MediaPipe: A Framework for Building Perception Pipelines. Google Research. 2019. https://arxiv.org/abs/2006.10214. "We describe a real-time on-device hand tracking pipeline for AR/VR applications that predicts hand skeleton from a single RGB camera. There are two models in the pipeline: 1) palm detector, and 2) hand landmark model It's done with MediaPipe, a framework for creating cross-platform machine learning solutions. On mobile GPUs, the proposed model and pipeline architecture show real-time inference speed and outstanding prediction quality."
4) Z.Xu, et.al, Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors, In Processing og IUI'09, 2009, pp 401-406. "This paper describes a novel hand gesture recognition system that utilizes both EMG sensors and a 3D accelerometer. Signal segments of meaningful gestures are determined from the continuous EMG signal inputs. For a set of 18 kinds of gestures, each trained with 10 repetitions, the average recognition accuracy was about 91.7%."
5) C.Chua, H. Guan, Y.Ho, Model-Based 3D Hand Posture Estimation From a Single 2D Image. Image and Vision Computing vol.20, 2002, pp. 191-202. “The 3D geometric posture of the human hand has been studied extensively over the past decade. A novel algorithm to estimate the 3D hand posture from eight 2D projected feature points is proposed. Experimental results using real images confirm that our algorithm gives good estimates of the hand pose.”

III. METHODOLOGY

The user guide application used in this study shows the steps done by the system by recognising hand gestures as commands. The MediaPipe framework and Python programming language are being used to create an application. There have been numerous exciting research accomplishments for articulating human body tracking, position, and even recognition systems. For capturing a real-time image to process using the MediaPipe framework.

1) MediaPipe Face Mesh is a mobile-friendly technology that estimates 468 3D face landmarks in real time. It uses machine learning (ML) to infer the 3D facial surface, which requires only a single camera input and no separate depth sensor. The method achieves real-time performance for live experiences by combining lightweight model designs with GPU acceleration across the pipeline. The Face Transform module is also included in the solution, which bridges the gap between face landmark estimation and usable real-time augmented reality (AR) applications. It creates a metric 3D space and estimates a face transform within that space using the facial landmark screen positions. The face transform data is made up of standard 3D primitives, such as a face pose transformation matrix and a triangular face mesh. Our machine learning pipeline is made up of two real-time deep neural network models that function together: a detector that operates on the entire image and computes face positions, and a 3D face landmark model that acts on those locations and uses regression to predict the approximate 3D surface. The necessity for typical data augmentations such as affine transformations consisting of rotations, translations, and scale modifications is greatly reduced when the face is precisely cropped. Instead, it allows the network to focus the majority of its resources on accuracy in coordinate prediction. Furthermore, in our pipeline, crops can be created based on the face landmarks recognised in the previous frame, with the face detector being activated only when the landmark model can no longer detect face presence. Face Transform Module The Facial Landmark Model detects face landmarks in the screen coordinate space using a single camera: the X and Y coordinates are normalised screen coordinates, while the Z coordinate is relative and scaled as the X coordinate under the weak perspective projection camera model. Although this format is suitable for some applications, it does not directly support the full range of augmented reality (AR) features, such as aligning a virtual 3D object with a detected face. The Face Transform module transforms a detected face into a standard 3D object by moving away from the screen coordinate space and into a metric 3D space. By design, you'll be able to utilise a perspective camera to project the completed 3D scene back into screen coordinate space while maintaining the location of the face landmarks.

a) *Key Concepts*

- *3D Metric Space:* The Face Transform module's Metric 3D coordinate space is a right-handed orthonormal metric 3D coordinate space. A virtual perspective camera is present in the space, positioned at the space origin and pointing in the negative Z-axis direction. The input camera frames are presumed to be watched by this virtual camera in the present pipeline, and its parameters are later utilised to transform the screen landmark coordinates back into Metric 3D space. The virtual camera settings can be set however, for best results, they should be set as closely as possible to the real physical camera parameters.

- *Canonical Face Model:* The Canonical Face Model is a 3D static model of a human face that adheres to the Face Landmark Model's 468 3D face landmark topology. The model serves two critical purposes:
  - The metric units of the Metric 3D space are defined by the scale of the canonical face model. A centimetre is a metric unit used by the default canonical face model.
  - The face position transformation matrix is a linear map from the canonical face model to the runtime face landmark set estimated on each frame, bridging static and runtime spaces. By applying the face position transformation matrix to virtual 3D assets based on the canonical face model, they can be aligned with a tracked face.

b) Components

- *Transform Pipeline:* The Transform Pipeline is a critical component that is in charge of estimating face transform objects in Metric 3D space. The following steps are performed in the following order on each frame: Face landmark screen coordinates are transformed into Metric 3D space coordinates; The face position transformation matrix is estimated as a rigid linear mapping from the canonical face metric landmark set into the runtime face metric landmark set in a way that minimises a difference between the two; The vertex positions (XYZ) of a face mesh are determined by the runtime face metric landmarks, but the vertex texture coordinates (UV) and triangle topology are inherited from the canonical face model. A MediaPipe calculator is being implemented by the transform pipeline... This calculator has been packed with relevant metadata into a unified MediaPipe subgraph for our convenience. A Protocol Buffer message is used to define the face transform format.
- *Effect Renderer:* The Effect Renderer is a component that acts as a working example of a renderer for face effects. It supports the following rendering modes and targets the OpenGL ES 2.0 API to allow real-time performance on mobile devices: 3D object rendering mode: a virtual object is aligned with a detected face to simulate an object on the face (example: glasses); Face mesh rendering mode: to simulate a face painting approach, a texture is stretched on top of the face mesh surface. The face mesh is drawn as an occluder straight into the depth buffer in both rendering modes. By hiding invisible elements under the face surface, this phase helps to produce a more convincing effect.

c) Configure Options

- *STATIC_IMAGE_MODEL:* The solution handles the input photos as a video stream if false is specified. It will attempt to recognise faces in the first input images and, if successful, it will localise the face landmarks further. Once all max num faces have been identified and the accompanying face landmarks have been located, it simply monitors those landmarks in consecutive photos without performing another detection until it loses track of any of the faces. This minimises latency and is perfect for video frame processing. Face detection is enabled on every input image if true, making it perfect for processing a batch of static, possibly unrelated images. False by default.
- *MAX_NUM FACES:* The maximum number of faces that can be detected. The default is 1.
- *REFINE_LANDMARKS:* Whether to use the Attention Mesh Model to refine the landmark coordinates around the eyes and lips and output additional landmarks around the irises. False by default.
- *MIN_DETECTION_CONFIDENCE:* The face detection model must have a minimum confidence value of [0.0, 1.0] for the detection to be regarded as successful. The default is 0.5.
- *MIN_TRACKING_CONFIDENCE:* For the face landmarks to be regarded and tracked correctly, the landmark-tracking model must have a minimum confidence value ([0.0, 1.0]), or else face detection will be triggered automatically on the next input image. Setting it to a higher number can improve the solution's robustness at the cost of increased latency. If static image mode is set to true, face detection is performed on every image. The default is 0.5.

d) Multi Face Landmarks

Faces that have been recognised or tracked are represented as a list of 468 face landmarks, with each landmark consisting of x, y, and z. The image width and height are used to normalise x and y to [0.0, 1.0]. The landmark depth is represented by z, with the origin being the depth at the centre of the head, and the smaller the value, the closer the landmark is to the camera. The magnitude of z is measured on a scale similar to that of x.

2) The capacity to recognise the shape and motion of hands can help improve the user experience across a wide range of technological domains and platforms. It may be used to comprehend sign language and control hand gestures, and it can also be used to overlay digital content and information on top of the physical world in augmented reality. Because hands frequently occlude themselves or one another (e.g. finger/palm occlusions and handshaking) and lack high contrast patterns, robust real-time hand perception is a difficult computer vision problem. MediaPipe hand is a high-resolution hand and finger tracking solution. Machine learning (ML) is used to deduce 21 3D landmarks of a hand from a single frame. Our solution delivers real-time performance on a mobile phone, and even scales to several hands, whereas existing state-of-the-art systems rely mostly on powerful desktop environments for inference. We anticipate that making these hand perception capabilities available to the broader research and development community will spur the creation of new applications and research directions.

a) ML Pipeline

MediaPipe Hands makes use of a machine learning pipeline that combines many models: An orientated hand bounding box is returned by a palm detection model that operates on the entire image. A hand landmark model that returns high-fidelity 3D hand key points from the cropped image region determined by the palm detector. This approach is similar to the one used in our MediaPipe Face Mesh solution, which combines a face detector with a face landmark model.

Providing the hand landmark model with a correctly cropped hand image dramatically minimises the requirement for data augmentation (e.g. rotations, translations, and scaling) and instead allows the network to focus on coordinate prediction accuracy... Furthermore, in our pipeline, crops can be created based on the hand landmarks recognised in the previous frame, with palm detection being used only when the landmark model can no longer detect hand presence. The pipeline is implemented as a MediaPipe graph that renders using a specific hand renderer subgraph and leverages a hand landmark tracking subgraph from the hand landmark module. A hand landmark subgraph from the same module and a palm detection subgraph from the palm detection module is used internally by the hand landmark tracking subgraph.

b) Configuration Options

- *STATIC_IMAGE_MODE:* The solution handles the input photos as a video stream if false is specified. It will attempt to recognise hands in the first input images and, if successful, will localise the hand landmarks further. After all, max num hands have been identified and the matching hand landmarks have been located in future photos, it simply monitors those landmarks without triggering another detection until it loses track of any of the hands. This minimises latency and is perfect for video frame processing. Hand detection runs on every input image if true, making it perfect for analysing a batch of static, possibly unrelated photos. False by default.
- *MAX_NUM_HANDS:* The maximum number of hands that can be detected. The default value is 2.
- *MODEL_COMPLEXITY:* The hand landmark model's complexity is either 0 or 1. With increasing model complexity, both benchmark accuracy and inference latency generally increase. The default is 1.
- *MIN_DETECTION_CONFIDENCE:* The hand detection model's minimum confidence value ([0.0, 1.0]) is required for the detection to be regarded as successful. The default is 0.5.
- *MIN_TRACKING_CONFIDENCE:* The landmark-tracking model's minimum confidence value ([0.0, 1.0]) for the hand landmarks to be regarded as successfully tracked, or else hand detection will be performed automatically on the next input image. Setting it to a higher number can improve the solution's robustness at the cost of increased latency. If static image mode is true, hand detection is simply applied to every image. The default is 0.5.

c) Multi Hand Landmarks

A collection of detected/tracked hands, each of which is represented by a list of 21 hand landmarks, each of which is made up of x, y, and z. The image width and height are used to normalise x and y to [0.0, 1.0]. The origin of z is the depth of the landmark at the wrist, and the smaller the value, the closer the landmark is to the camera. The magnitude of z is measured on a scale similar to that of x.

d) Multi Hand World Landmarks

Each hand is represented as a list of 21 hand landmarks in world coordinates in this collection of detected/tracked hands. Each landmark is made up of three numbers: x, y, and z, which are real-world 3D coordinates in metres, with the origin at the approximate geometric centre of the hand.

e) Multi Handedness

Collection of the detected/tracked hands' handedness (i.e. is it a left or right hand). Labels and scores make up each hand. the label is a string with the value "Left" or "Right" in it. the score is the projected handedness's estimated probability, which is always greater than or equal to 0.5. (and the opposite handedness has an estimated probability of 1 - score). Note that the input image is assumed to be mirrored, i.e. captured with a front-facing/selfie camera with photos reversed horizontally. If this is not the case, change the application's handedness output.

IV. MODEL DESCRIPTION

In this section, we are going to discuss models in this project.

A. Face Landmark Model

For 3D face landmarks, we used transfer learning and trained a network with several goals: the network predicts 3D landmark coordinates on synthetic rendered data while also predicting 2D semantic contours on annotated real-world data. Not just on synthetic data, but also on real-world data, the resulting network provided us with reasonable 3D landmark predictions. A cropped video frame is fed into the 3D landmark network without any additional depth information. The model returns the 3D point coordinates as well as the likelihood that a face is present and adequately aligned in the input. Predicting a 2D heatmap for each landmark is a typical alternative, but it is not suitable for depth prediction and has large processing costs for so many landmarks. We iteratively bootstrap and refine predictions to increase the accuracy and resilience of our model. In that manner, we may expand our dataset to include more difficult scenarios like grimaces, oblique angles, and occlusions.

B. Attention Mesh Model

We also present a model that focuses attention on semantically relevant facial regions, predicting landmarks more reliably around lips, eyes, and irises at the cost of more computation, in addition to the Face Landmark Model. It enables applications like augmented reality cosmetics and augmented reality puppeteering.

C. Palm Detection Model

We created a single-shot detector model tailored for mobile real-time purposes, comparable to the face detection model in MediaPipe Face Mesh, to detect initial hand placements. Hand detection is a difficult task: both our lite and complete models must detect occluded and self-occluded hands and work across a wide range of hand sizes with a huge scale span (>20x) relative to the image frame. Faces have high contrast patterns, such as around the eyes and mouth, while hands lack similar features, making it more difficult to consistently distinguish them from their visual features alone. Providing additional contexts, such as arm, body, or human traits, instead helps with precise hand localisation.

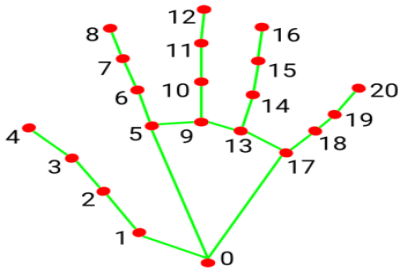
Our strategy employs a variety of strategies to address the aforementioned issues. First, instead of training a hand detector, we train a palm detector, because estimating bounding boxes of rigid objects like palms and fists is much easier than recognising hands with articulated fingers. Furthermore, because palms are smaller objects, the non-maximum suppression method performs effectively even in two-hand self-occlusion situations such as handshakes. Furthermore, palms can be simulated using square bounding boxes (anchors in ML language) that ignore other aspect ratios, resulting in a reduction of 3-5 anchors. Second, even for little objects, an encoder-decoder feature extractor is used for larger picture context awareness (similar to the RetinaNet approach). Finally, to support a large number of anchors, we reduce focal loss during training which is a product of high scale variance.

We attain an average precision of 95.7 per cent in palm detection using the strategies described above. With no decoder and a regular cross-entropy loss, the baseline is just 86.22 per cent.

D. Landmark Model

Following palm detection over the whole image, our subsequent hand landmark model uses regression to conduct exact keypoint localization of 21 3D hand-knuckle coordinates within the detected hand regions. Even with partially visible hands and self-occlusions, the model develops a consistent internal hand posture representation.

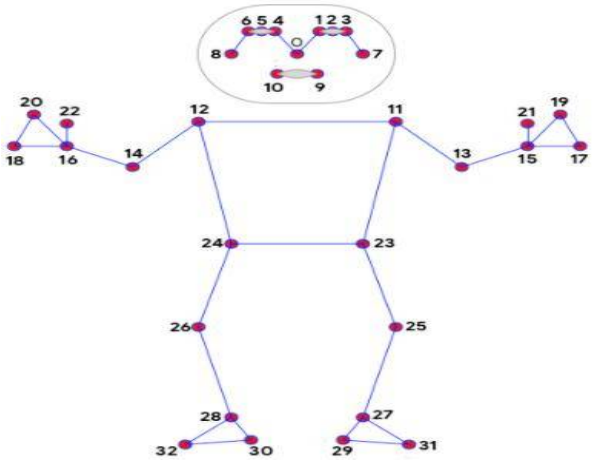
We manually tagged 30K real-world photos with 21 3D coordinates to obtain ground truth data, as seen below (we take Z-value from the image depth map if it exists per corresponding coordinate). We also generate a high-quality synthetic hand model over various backgrounds and map it to the associated 3D coordinates to better cover the available hand poses and provide additional supervision on the nature of hand geometry.



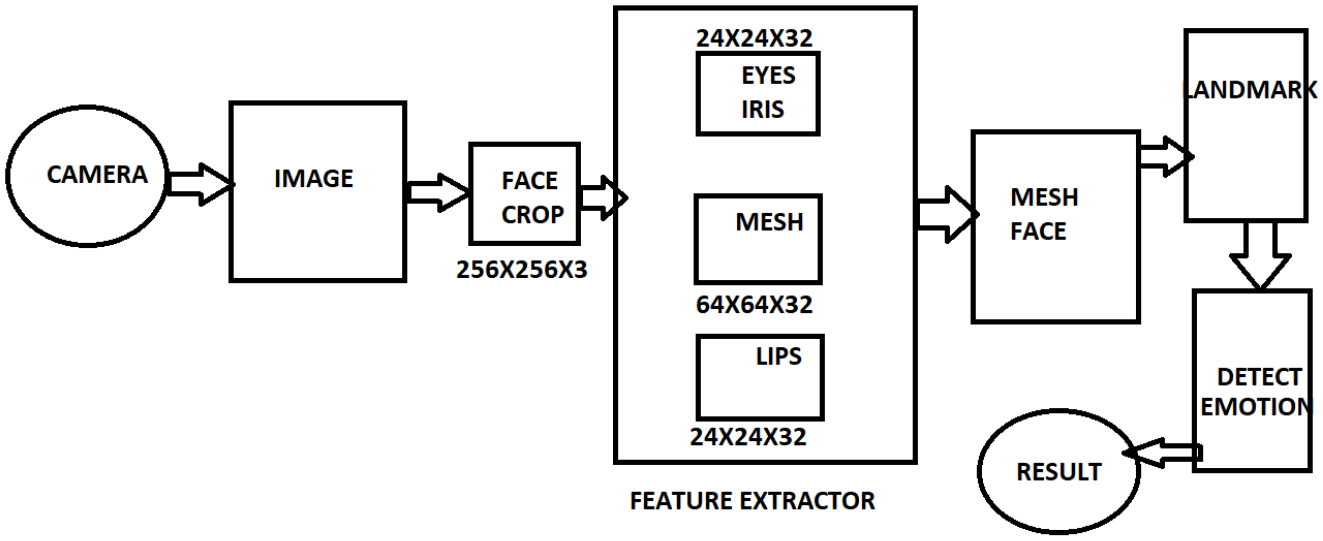
0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

E. Pose Landmark Model (BlazePose GHUM 3D)

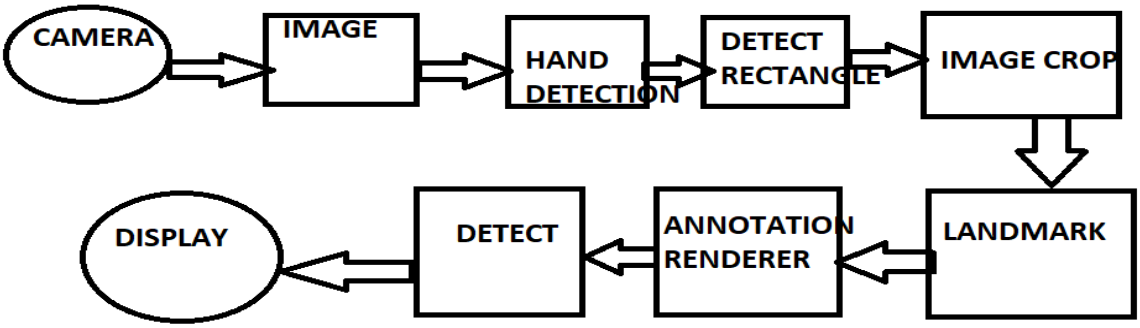
The landmark model in MediaPipe Pose predicts the location of 33 pose landmarks (see figure below).

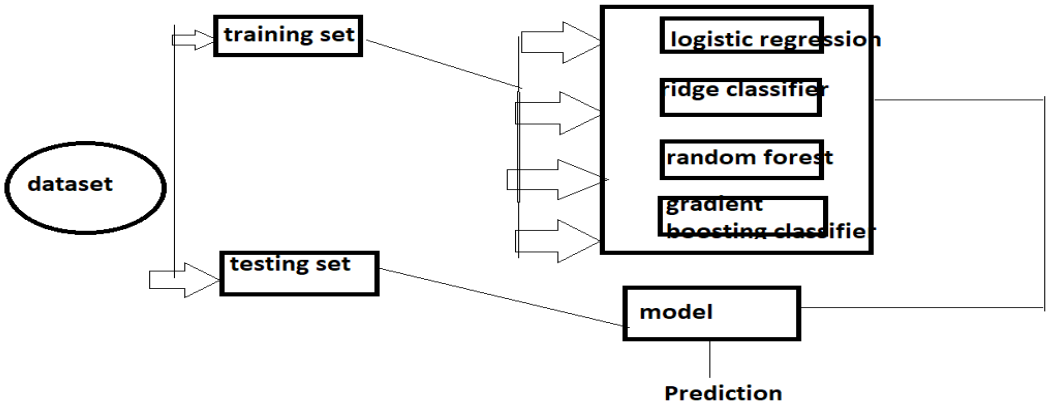


V. SYSTEM ARCHITECTURE



Face emotion detection with attention mesh model and gesture.





VI. RESULTS

Following all of the preceding, the suggested system's outcomes are generated by using several Machine Learning methods. Python is used to build Machine Learning classification algorithms such as logistic regression, Ridge classifier, Random forest, and Gradient boosting classifier. Four performance evaluation measures are utilised to evaluate the proposed system. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) make up the confusion matrix (FN). The confusion matrix can be represented graphically as follows:

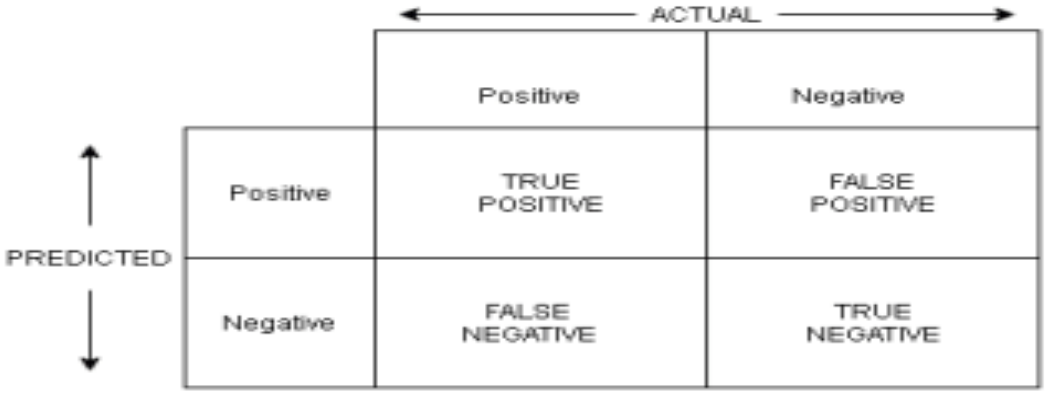


Figure 1: Confusion Matrix

The report includes the values for accuracy, precision, recall, support, and F1-score and also includes a confusion matrix for each Machine learning algorithm. These are explained as:

1) *Accuracy:* The classification accuracy is described as the ratio of correct predicted values to the total predicted values and is mathematically depicted as follows:

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

2) *Precision:* The precision or positive predictive value (PPV) is the number of correctly identified positive results divided by the number of all positive results and is mathematically depicted as follows:

$$\text{PRECISION} = \frac{\text{TRUE POSITIVES (TP)}}{\text{TRUE POSITIVES (TP)} + \text{FALSE POSITIVES (FP)}}$$

3) *Recall:* The recall or sensitivity or true positive rate (TPR) is the number of correctly identified positive results divided by the number of all samples that should have been identified as positive and is mathematically depicted as follows:

$$\text{RECALL} = \frac{\text{TRUE POSITIVES (TP)}}{\text{TRUE POSITIVES (TP)} + \text{FALSE NEGATIVES (FN)}}$$

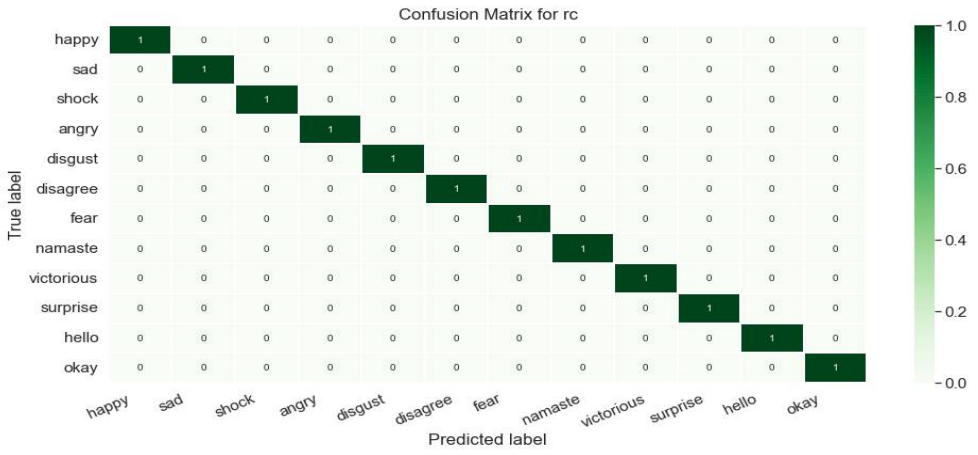
4) *F1-Score:* The F1 score (also F-score or F-measure) is a measure of a test's accuracy. It is calculated from the precision and recall of the test. mathematically it can be as follows:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precison} + \text{Recall}}$$

5) *Support is the Number of Actual Occurrences of the class in a Specified Dataset.*

The prediction outcomes for different machine learning models are given below:

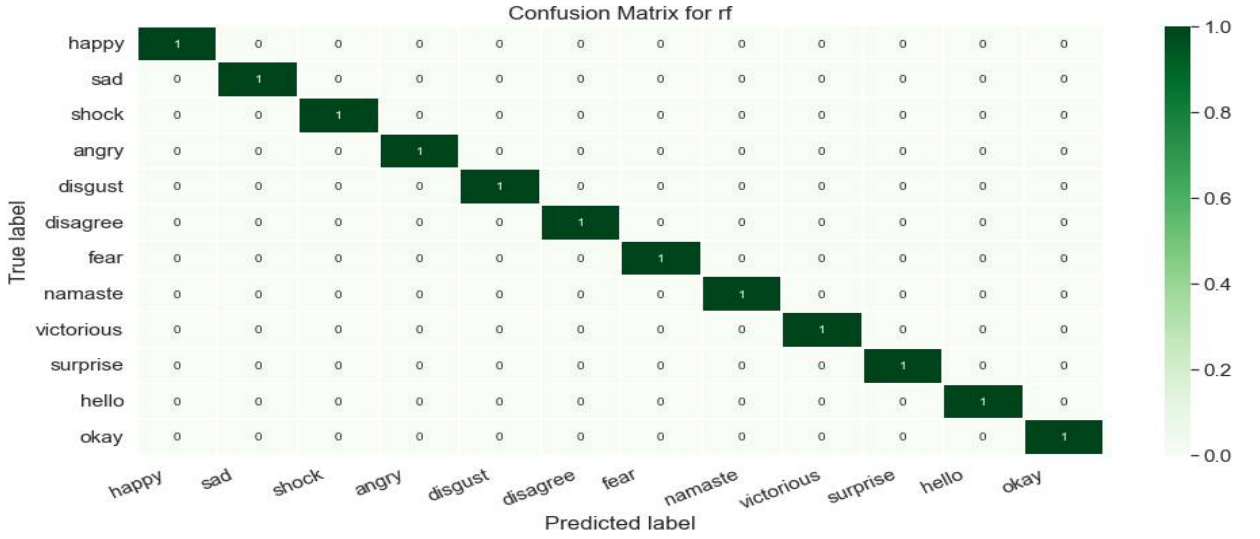
a) *Ridge Classifier:* Based on the Ridge regression method, the Ridge Classifier turns the label data into [-1, 1] and solves the problem using the regression method. The target class with the greatest prediction value is chosen, and multi-output regression is used for multiclass data.



	precision	recall	f1-score	support
Happy	1.00	1.00	1.00	22
angry	1.00	1.00	1.00	32
disagree	1.00	1.00	1.00	18
disgust	1.00	1.00	1.00	13
fear	1.00	1.00	1.00	13
hello	1.00	1.00	1.00	24
namaste	1.00	1.00	1.00	28
okay	1.00	1.00	1.00	20
sad	1.00	1.00	1.00	22
shock	1.00	1.00	1.00	25
surprise	1.00	1.00	1.00	6
victorious	1.00	1.00	1.00	11
accuracy			1.00	234
macro avg	1.00	1.00	1.00	234
weighted avg	1.00	1.00	1.00	234

```
rc 1.0
[[22  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 32  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 18  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 13  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 13  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 24  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 28  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 20  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 22  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 25  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  6  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 11]]
```

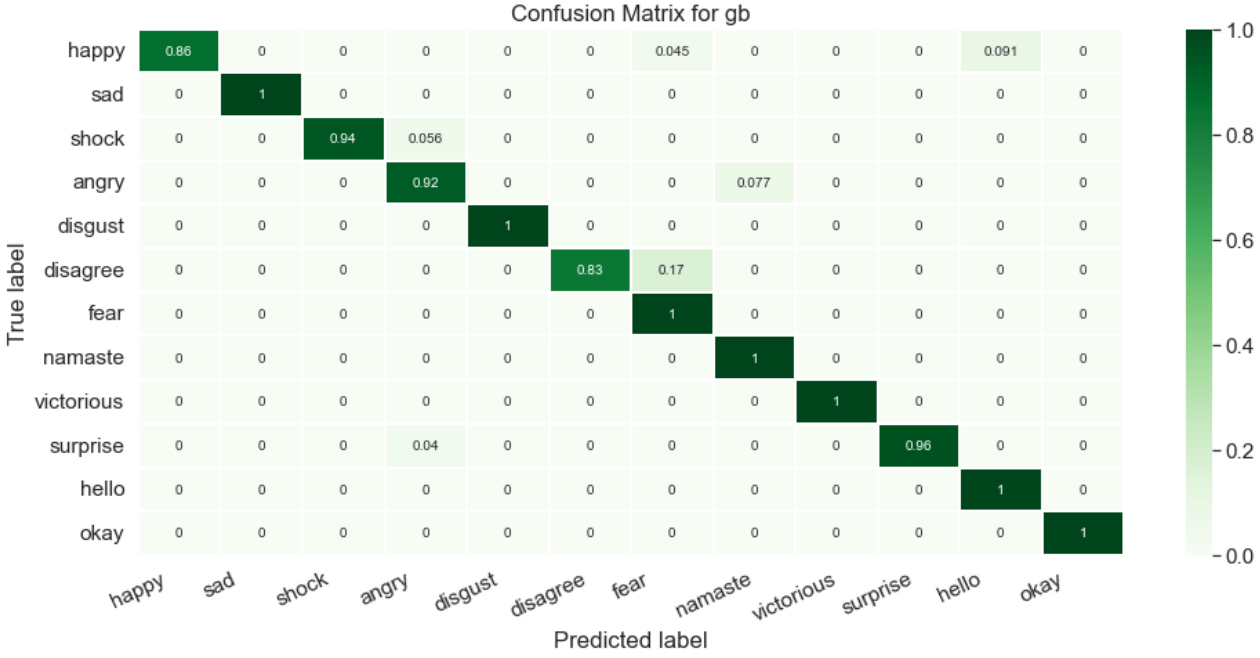
b) Random Forest: Random forest is a supervised machine learning algorithm that is commonly used to solve classification and regression problems. It creates decision trees from various samples, using the majority vote for classification and the average for regression.



	precision	recall	f1-score	support
Happy	1.00	1.00	1.00	22
angry	1.00	1.00	1.00	32
disagree	1.00	1.00	1.00	18
disgust	1.00	1.00	1.00	13
fear	1.00	1.00	1.00	13
hello	1.00	1.00	1.00	24
namaste	1.00	1.00	1.00	28
okay	1.00	1.00	1.00	20
sad	1.00	1.00	1.00	22
shock	1.00	1.00	1.00	25
surprise	1.00	1.00	1.00	6
victorious	1.00	1.00	1.00	11
accuracy			1.00	234
macro avg	1.00	1.00	1.00	234
weighted avg	1.00	1.00	1.00	234

```
rf 1.0
[[22  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 32  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 18  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 13  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 13  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 24  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 28  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 20  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 22  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 25  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  6  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 11]]
```

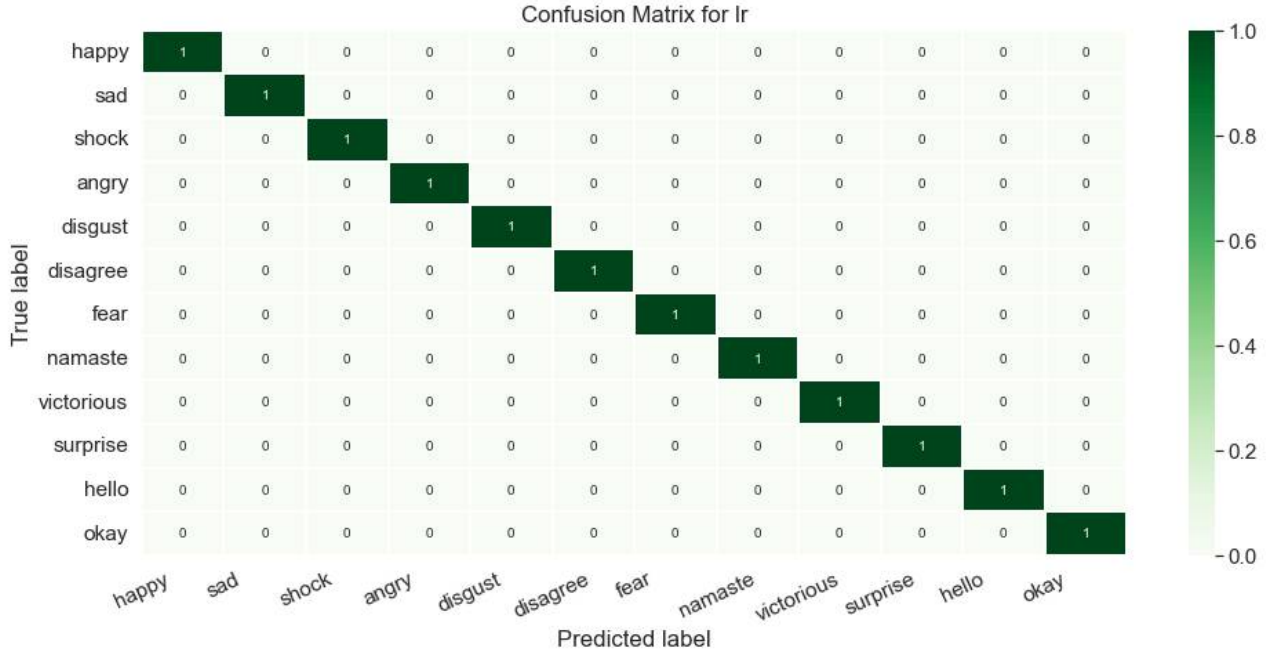
c) Gradient Boosting Classifier: Machine learning boosting is a kind of gradient boosting. It is based on the assumption that when the best potential next model is coupled with prior models, the overall prediction error is minimised. To decrease error, the fundamental notion is to specify the target outcomes for the following model.



	precision	recall	f1-score	support
Happy	1.00	0.86	0.93	22
angry	1.00	1.00	1.00	32
disagree	1.00	0.94	0.97	18
disgust	0.86	0.92	0.89	13
fear	1.00	1.00	1.00	13
hello	1.00	0.83	0.91	24
namaste	0.85	1.00	0.92	28
okay	0.95	1.00	0.98	20
sad	1.00	1.00	1.00	22
shock	1.00	0.96	0.98	25
surprise	0.75	1.00	0.86	6
victorious	1.00	1.00	1.00	11
accuracy			0.96	234
macro avg	0.95	0.96	0.95	234
weighted avg	0.96	0.96	0.96	234

```
gb 0.9572649572649573
[[19  0  0  0  0  0  1  0  0  0  2  0]
 [ 0 32  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 17  1  0  0  0  0  0  0  0  0]
 [ 0  0  0 12  0  0  0  1  0  0  0  0]
 [ 0  0  0  0 13  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 20  4  0  0  0  0  0]
 [ 0  0  0  0  0  0 28  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 20  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 22  0  0  0]
 [ 0  0  0  1  0  0  0  0  0 24  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  6  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 11]]
```

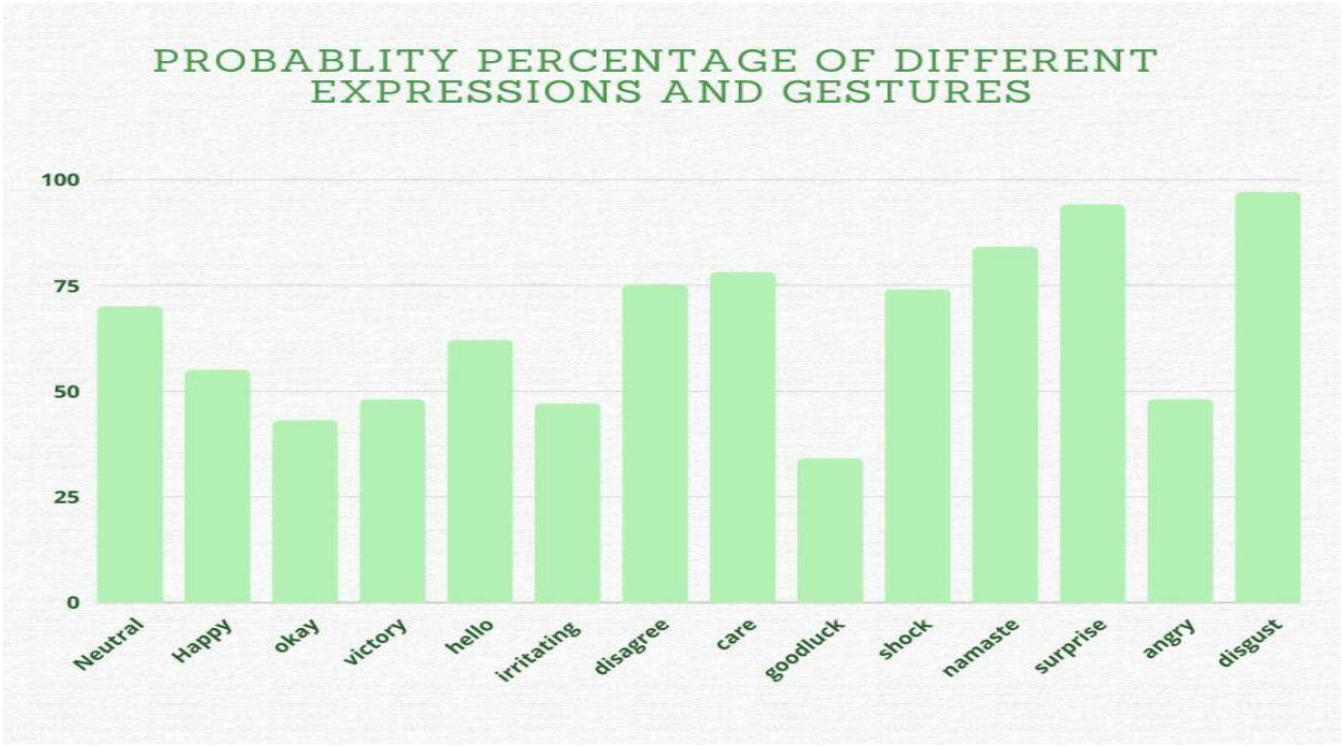
d) Logistic Regression: Based on prior observations of a data set, logistic regression is a statistical analytic approach for predicting a binary outcome, such as yes or no. A logistic regression model analyses the relationship between one or more independent factors to predict a dependent data variable.



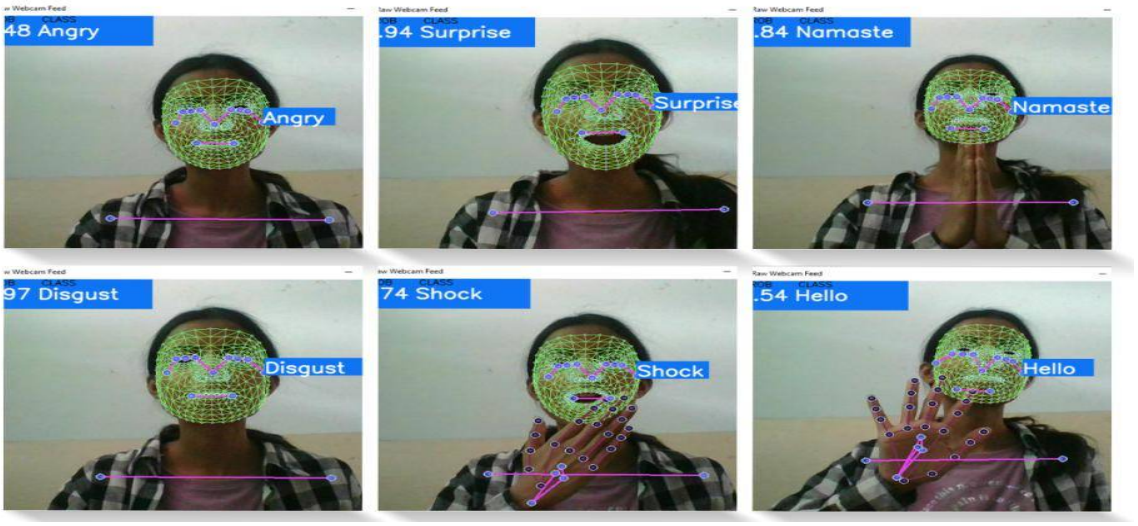
	precision	recall	f1-score	support
Happy	1.00	1.00	1.00	22
angry	1.00	1.00	1.00	32
disagree	1.00	1.00	1.00	18
disgust	1.00	1.00	1.00	13
fear	1.00	1.00	1.00	13
hello	1.00	1.00	1.00	24
namaste	1.00	1.00	1.00	28
okay	1.00	1.00	1.00	20
sad	1.00	1.00	1.00	22
shock	1.00	1.00	1.00	25
surprise	1.00	1.00	1.00	6
victorious	1.00	1.00	1.00	11
accuracy			1.00	234
macro avg	1.00	1.00	1.00	234
weighted avg	1.00	1.00	1.00	234

```
lr 1.0
[[22  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 32  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 18  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 13  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 13  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 24  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 28  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 20  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 22  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 25  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  6  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 11]]
```

The prediction for respective emotions and gestures are as follows:



International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

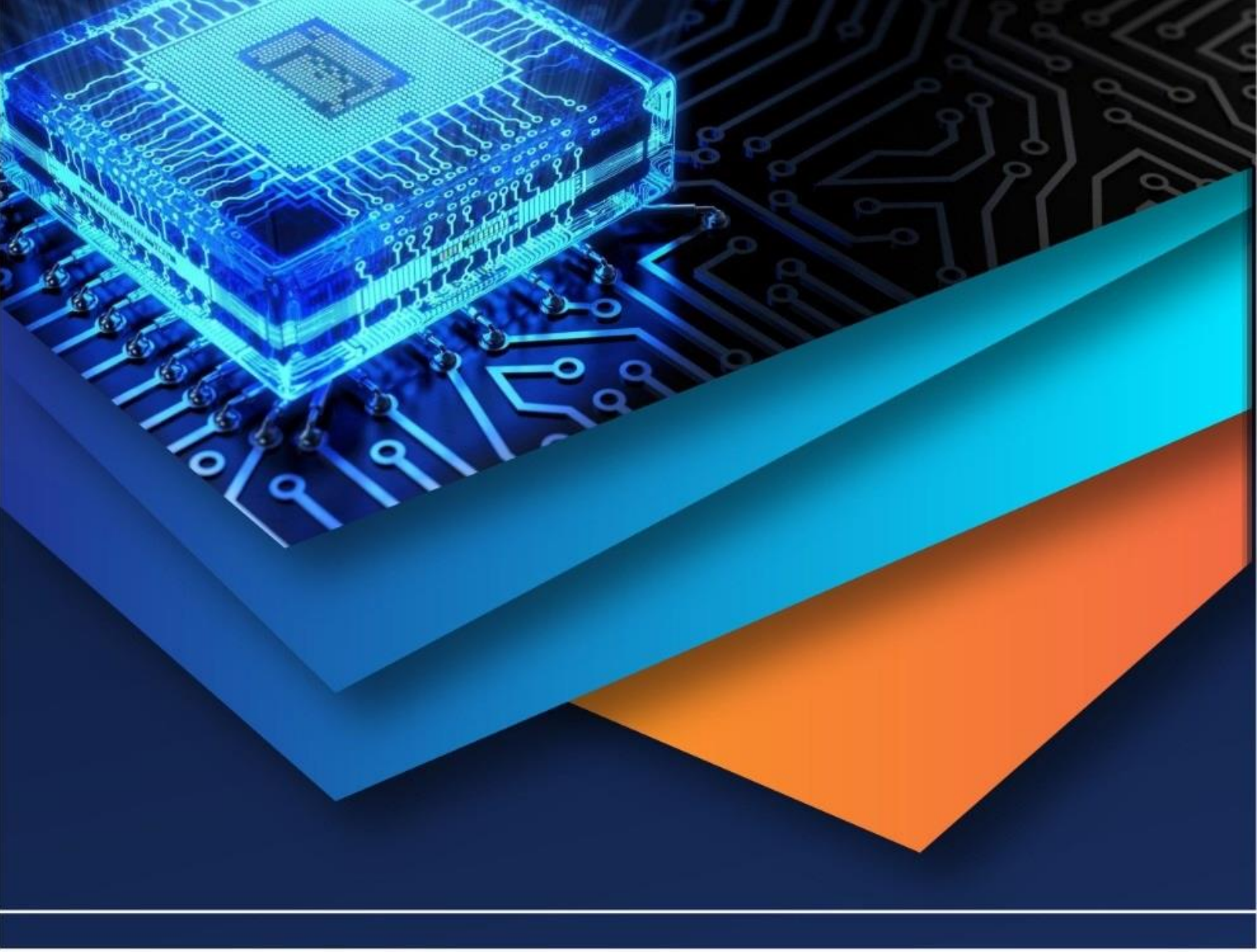


VII. CONCLUSIONS AND FUTURE SCOPE

Body language detection and analysis have recently received a lot of attention. This paper covers the emotion of the face and gestures identification. In this paper, we discussed the work done on emotion recognition and the methods used to achieve better and unique ways and approaches. We have sketched out a possible solution and method for emotional recognition. We are using different machine learning models such as random forest, gradient classifier, linear regression and ridge classifier to obtain results of emotion and gesture detection. We have used the MediaPipe framework which is based on real-time deep learning, to generate critical points. The ability to recognise and analyse client/customer facial expressions aids organisations and marketing teams in obtaining honest assessments and feedback. However, the facial expression is only one aspect of body language. Other parts of body language include hand gestures and body positions. In addition, body language is also crucial in communicating. In interviews, for example, interviewers analyse the candidate's body language. We're working on creating an emotional machine. A machine or system that can think like humans can experience warmth in the heart, judge events, prioritise choices, and express many other emotions. To make the fantasy a reality, we need a machine or system that can understand apes and master human emotions. We've only recently begun to do so. Even so, there are several true examples these days. Some features and services, such as Microsoft Cognitive Services, are gaining popularity, but much work remains in terms of efficiency, accuracy, and usability. Consequently, in the future interviewers will have access to a tool that will help them better understand how candidates answer questions from various areas or circumstances during HR rounds. Because this project allows for real-time hand landmark identification, hand sign language can be used. Not only that, but by utilising this project, existing projects such as driver sleepiness detection, action detection, and others may be made much easier to execute with much better results.

REFERENCES

[1] Z.Ren, J.Meng, Yuan J. Depth Camera Based Hand Gesture Recognition and its Application in human-computer-Interaction. In Processing of the 2011 8th International Conference on Information, Communication and Signal Processing (ICICS). Singapore. 2011.
[2] S.Rautaray S, Agrawal A. Vision-Based Hand Gesture Recognition for Human-Computer Interaction: A Survey. Springer Artificial Intelligence Review. 2012. DOI: https://doi.org/10.1007/s10462-012-9356-9.
[3] Lugaresi C, Tang J, Nash H, McClanahan C, et al. MediaPipe: A Framework for Building Perception Pipelines. Google Research. 2019. https://arxiv.org/abs/2006.10214.
[4] Z.Xu, et.al, Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors, In Processing og IUI'09, 2009, pp 401-406.
[5] C.Chua, H. Guan, Y.Ho, Model-Based 3D Hand Posture Estimation From a Single 2D Image. Image and Vision Computing vol.20, 2002, pp. 191-202.
[6] https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html
[7] https://heartbeat.fritz.ai/simultaneously-detecting-face-hand-motion-and-pose-in-real-time-on-mobile-devices-27849560fc4e
[8] https://medium.com/jstack-eu/using-machine-learning-to-analyse-body-language-and-facial-expressions-a779172cc98
[9] http://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html
[10] https://google.github.io/mediapipe/solutions/pose.html
[11] https://en.wikipedia.org/wiki/F-score
[12] https://www.reddit.com/user/tutort-academy/comments/s7nh0j/effectiveness_of_random_forest/
[13] https://techrobcrew.blogspot.com/2021/06/hand-tracking-in-real-time.html/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY