IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Emotion Detection Using Learning Methods

Author Name: Revanashiddayya Malagitti[1]
Selection Grade Lecturer in CSE
Government Polytechnic Gajendragad, Karnataka

Author Name:Sanjay Lote[2]
Senior Scale Lecturer in CSE
Government Polytechnic Rabakavi-Banahatti, Karnataka

ABSTRACT

In human-computer interaction and emotional analysis, accurately understanding feelings in real-time is crucial but challenging. Existing systems often struggle with interpreting facial expressions, causing errors in user interactions. Our solution integrates efficient face detection using the Haar Cascade Classifier with a pre-trained Convolutional Neural Network (CNN) in TensorFlow for swift and accurate emotion deciphering. This fusion combines the Haar Cascade Classifier's face detection proficiency with the CNN model's speed and accuracy in emotion recognition, resulting in a notable 84.57% accuracy improvement. Our goal is to develop a real-time system that reliably identifies emotions on faces, enhancing empathetic and responsive interaction between humans and computers. This holistic approach advances emotional analysis in human-computer interactions, aiming for more intuitive systems that can facilitate personalized user experiences.

Keywords: Human-computer interaction, Emotional analysis, Real-time understanding, Facial expressions, Face detection, Haar Cascade Classifier, Convolutional Neural Network (CNN), TensorFlow, Emotion deciphering, Accuracy improvement, Real-time emotion identification, Empathetic interaction, Personalized user experiences.

I. INTRODUCTION

This project employs deep learning techniques for emotion recognition from images, addressing a critical aspect of human-computer interaction and sentiment analysis. The dataset, stored in the 'dataset' directory, contains a collection of images labelled with various emotions. These images undergo preprocessing, including shuffling and extraction of labels, facilitated by libraries such as NumPy, Pandas, and OpenCV. The resulting DataFrame provides a structured representation of the dataset, essential for subsequent model training and evaluation.To train a model for emotion recognition, the dataset is split into training and validation sets using TensorFlow's image data preprocessing utilities. The selected architecture for the model leverages the EfficientNetV2M, a state-of-the-art convolutional neural network pre-trained on ImageNet, serving as a feature extractor.This base model is augmented with additional layers, including a flattened layer for reshaping, a Dropout layer for regularization, and a Dense layer for classification into six emotion classes. The model is then compiled with the Adam optimizer and trained over seven epochs, with performance monitored on the validation set. The trained model's efficacy is evaluated through various metrics, including test loss and accuracy, providing insights into its generalization capability. Additionally, the training and validation metrics are visualized using Matplotlib, enabling a comparative analysis of the model's performance across epochs. This project demonstrates a comprehensive pipeline for emotion recognition from images, offering a foundation for further research and application in domains such as affective computing, user experience design, and mental health monitoring.Facial Emotion RecognitionVarious studies have explored facial emotion recognition using machine learning and deep learning approaches. Literature often discusses the use of convolutional neural networks (CNNs) for image-based emotion recognition, similar to the approach employed in the code.HarCascadeClassifiersare widely used for object detection, including faces.

The literature may discuss the HarCascade methodology, its advantages, and improvements in face detection algorithms.Datasets for Emotion RecognitionResearch often involve the utilization and analysis of specific datasets designed for emotion recognition tasks. Datasets like CK+, FER2013, or AffectNet might be mentioned in the literature.Real-time Video ProcessingTechniques for real-time video processing are crucial for applications like live emotion detection. the literature might cover optimization strategies and algorithms for efficient video stream processing.Graphical Analysis and Visualization,The use of Matplotlib for graphical analysis is common in various domains.

Literature may discuss the significance of visualizing emotional data and the different techniques for achieving this.Deep Learning Models for Emotion Recognition. The literature may delve into the architecture and training strategies of deep learning models, including CNNs, for emotion recognition.Techniques for transfer learning or fine-tuning pre-trained models might be explored.Applications of Emotion RecognitionBeyond the technical aspects, the literature may discuss the applications of facial emotion recognition in diverse fields, such as human-computer interaction, healthcare, and affective computing.Challenges and Future Directions research papers often highlight challenges in emotion recognition, such as handling diverse facial expressions, dealing with imbalanced datasets, and ensuring model interpretability. Future directions for improving the accuracy and robustness of emotion recognition systems may also be discussed.

The objective of this project is to develop a deep-learning model for emotion recognition from images. Emotion recognition is crucial in various domains, including human-computer interaction, marketing, and mental health monitoring. By accurately identifying emotions from images, this model aims to enhance user experience, personalize content delivery, and provide insights into emotional states.

This invention relates to, emotion detection using deep learning and analyzing human emotions in real-time during interactions with computers. It employs advanced technologies such as OpenCV for face detection, and Haar cascade for Object detection which identify the face within the image and video. EfficientNetV2 serves as a feature extractor. The convolutional Neural Network (CNN) model was implemented using TensorFlow for emotion recognition. Integrating these techniques improves the efficiency and accuracy of emotion analysis, leading to better user experiences and more empathetic interactions between humans and computers.

II. RELATED WORK

In related work on emotion recognition methods across various modalities, Alswaidan&Menai (2020) explored text emotion recognition using rule-based, classical learning, deep learning, and hybrid approaches.Rule-based approaches may lack flexibility and generalization, especially for capturing nuanced emotions.Deep learning models might require large amounts of labelled data, which can be challenging to obtain in text emotion recognition tasks.there is a gap in understanding how these methods can be effectively combined in hybrid approaches. Research is needed to develop hybrid models that leverage the strengths of each approach to improve overall performance in text emotion recognition tasks.

Abdullah et al. (2021) focused on multimodal emotion recognition methods utilizing neural network architectures and deep learning techniques. Multimodal emotion recognition can be complex and computationally intensive, particularly when integrating information from different modalities.Deep learning techniques may suffer from overfitting if not properly regularized, especially when dealing with multimodal data. There is a gap inDeep learning methodologies that heavily rely on large datasets, and the availability of such datasets in audiovisual emotion recognition might be limited. Performance may degrade in real-world scenarios where environmental factors can introduce noise and variability in audiovisual data.

Wu et al. (2014)developed audiovisual emotion recognition techniques, emphasizing deep learning methodologies and datasets like RML and VAM.Deep learning methodologies heavily rely on large datasets, and the availability of such datasets in audiovisual emotion recognition might be limited. Performance may degrade in real-world scenarios where environmental factors can introduce noise and variability inaudio-visual data. There is a gap in exploring and applying effective regularization techniques to prevent overfitting and improve model generalization.

Baltrusaitis et al. (2018) discussed technical challenges encountered by multimodal researchers, particularly highlighting different co-learning methods. Co-learning methods may require significant computational resources and may not always generalize well across different multimodal datasets.Technical challenges discussed may not have practical solutions yet, limiting the immediate applicability of the research. The gaps are Scalability and Generalization of Co-Learning Methods

Jam et al. (2021) investigated social signals and emotional expressions in real-world human-robot interactions, employing deep learning methods for analysis. Real-world human-robot interaction datasets may be limited in size and diversity, which could affect the generalization of the deep learning models.Deep learning methods might struggle to capture subtle social signals and nuances in emotional expressions, leading to misinterpretations. The gaps are Capturing Subtle Social Signals and Nuances**.**

Lim et al. (2020) established a taxonomy involving eye-tracking methods within machine learning algorithms. Eye-tracking methods may have limited applicability outside controlled laboratory settings, where factors like lighting conditions and user behaviour are more variable. the research gaps inMachine learning algorithms may struggle to generalize eye-tracking data across different individuals and cultural backgrounds.

Lastly, Malla et al. (2020) focused on speech emotion recognition methods, utilizing features like MFCC, STFT, and ECC, and employing a classification framework combining CNN and LSTM models. Speech emotion recognition methods may face challenges with noisy environments and speaker variability, impacting the robustness of the models.Combining CNN and LSTM models may increase computational complexity and training time, especially for large-scale datasets.These studies collectively contribute to a comprehensive understanding of diverse methodologies and challenges in the field of emotion recognition across text, multimodal, audiovisual, social interaction, eye-tracking, and speech-based modalities. The gaps areCombining CNN and LSTM models for speech emotion recognition may increase computational complexity and training time, particularly for large-scale datasets. This can pose challenges in terms of resource requirements and scalability.

III. MATERIAL AND METHOD

1. Material

OpenCV:(Open Source Computer Vision Library) is essential for face detection using the Haar Cascade Classifier.Installthe OpenCV library in Python using pip (pip install opencv-python).**Pre-trained CNN Model:**Choose a pre-trained CNN model for emotion recognition, such as VGG, ResNet, or EfficientNet, available through TensorFlow/Keras.Installthe TensorFlow library (pip install tensorflow) to access pre-trained CNN models and perform transfer learning.**Face Detection Haar Cascade Classifier:**Download the pre-trained Haar Cascade classifier XML file for face detection.**Real-Time Video Processing Library:**Use a real-time video processing library (e.g., OpenCV) for capturing and processing video frames.UseOpenCV (cv2.VideoCapture) to capture video streams from webcam or video files.**PythonLibraries:**Utilize essential Python libraries for array manipulation, image processing, and model inference.InstallNumPy (pip install numpy) for array operations and Matplotlib (pip install matplotlib) for visualizations.**ValidationDatasets:**Use validation datasets like CK+, FER2013, or custom datasets with labelled emotion annotations for model evaluation.Download and preprocess validation datasets to evaluate the system's performance on emotion recognition tasks.**Graphical User Interface (GUI) Framework (optional):**If developing a user-friendly interface, use GUI frameworks like Tkinter or PyQt for building interactive applications.InstallTkinter (pip install tk) for basic GUI development in Python.

2. Methods:

Input Image/Video Frame:The process starts with input images or video frames of human faces. **Face Detection with Haar Cascade Classifier**:OpenCV is used to apply the Haar Cascade Classifier to detect faces in the input images or video frames.Detected faces are identified as regions of interest (ROIs).**Preprocessing**: Preprocessing techniques may be applied to the detected face regions, such as resizing, normalization, and grayscale conversion. Preprocessed face images are prepared for emotion recognition.

IV. DESIGN AND IMPLEMENTATION

1. Architectural design

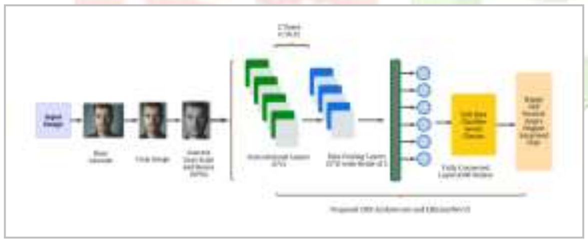

Fig.1 Architectural Diagram

Image Acquisition: Input images are captured using cameras or other imaging devices.Preprocessing steps may include face detection to locate and extract facial regions from the images.**Facial Landmark Detection:**Identify key facial landmarks, such as eyes, nose, and mouth, using techniques like the Viola-Jones algorithm.**Normalization and Alignment:**Normalize the facial region to a standard size and orientation.Align facial landmarks to a predefined template to ensure consistency across different images.**FeatureExtraction:**Extract relevant features from the normalized and aligned facial images.

Dimensionality Reduction: Reduce the dimensionality of the feature space to improve computational efficiency and reduce the risk of overfitting.Techniques like Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) may be employed for dimensionality reduction.

Classifier Training: Train a machine learning model using the preprocessed and reduced-dimensional feature vectors. Popular classifiers include Convolution Neural Network (CNN), or Ensemble Techniques using Deep Learning.**Cross-Validation:**Validate the trained model using cross-validation techniques to assess its generalization performance.

Emotion Classification: Apply the trained model to new facial images to predict the emotional state.Emotion labels may include basic emotions such as happiness, sadness, anger, surprise, fear, and disgust.

Post-Processing: Perform post-processing steps to refine the results and improve the overall accuracy. Techniques like temporal smoothing or ensemble methods may be applied.

2. Flow work

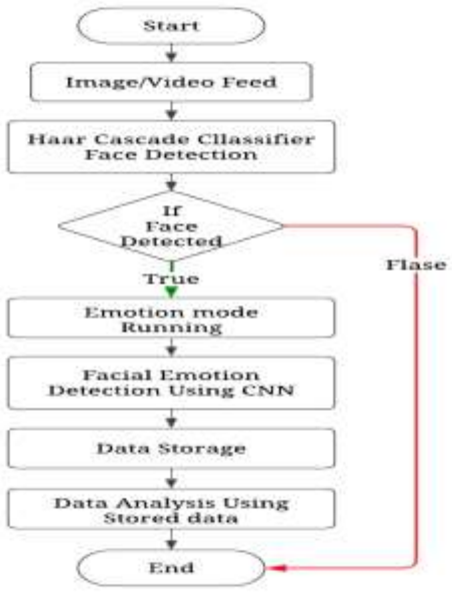
Start
Image/Video Feed
Haar Cascade Classifier Face Detection
If Face Detected
True
Flase
Emotion mode Running
Facial Emotion Detection Using CNN
Data Storage
Data Analysis Using Stored data
End

Fig.2 Flow Chart

CNN Training:TensorFlow is used to train a Convolutional Neural Network (CNN) model for facial emotion recognition.The CNN model is trained using the training dataset, with images as input and corresponding emotion labels as output.Training involves multiple epochs, and the model learns to extract features from face images and predict emotions.

Real-Time Emotion Detection:After training and evaluation, the trained CNN model is used for real-time emotion detection on input images or video frames.Detected faces are passed through the trained CNN model, which predicts the corresponding emotions such as happy, sad, anger, fear, disgust, and surprise.**DataStorage**:Detected and preprocessed face images along with corresponding labels (emotions) are stored in a dataset for training the CNN model.

The dataset is split into training and validation sets for model training and evaluation.

Model Evaluation:The trained CNN model is evaluated using the validation dataset to assess its performance in recognizing emotions from facial expressions.Evaluation metrics such as accuracy, precision, recall, and F1-score are calculated to measure the model's performance.

Output Visualization:The emotions predicted by the CNN model are visualized on the input images or video frames.Output may include text labels or graphical representations indicating the recognized emotions (e.g., happy, sad, angry).

End:The process concludes after real-time emotion detection and visualization, completing the facial emotion detection workflow.

3. Sequence Diagram

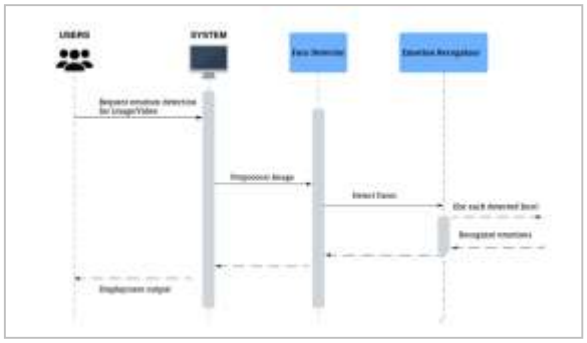

Fig.3 Sequences Diagram

The sequence starts with the user interacting with the system, providing input data. This input data could be images, or videos, containing expressions or emotions. For example, the user may give an image or input a live video for the system.Upon receiving the input data, the system preprocesses it to make it suitable for the deep learning model. Preprocessing steps may include resizing images, converting images to greyscale, normalization, or feature extraction. For example, in image data preprocessing, images may be resized to a standard size and normalized to ensure consistency in input data.If the input data is an image or video, the system performs face detection to identify regions of interest likely containing faces. Techniques like the Haar Cascade Classifier in OpenCV are commonly used for face detection. Once faces are detected, the system focuses on these regions for further processing.The preprocessed data, which now includes the detected face regions, is passed through the trained deep-learning model for inference.The deep learning model, which has been trained to recognize emotions from facial expressions, analyzes the input data and makes predictions.This step involves feeding the preprocessed image data into the model's input layers. The trained deep learning model predicts the emotion present in the input data based on the learned features. For example, the model may predict emotions such as happiness, sadness, anger, etc., based on the facial expressions detected in the image.The predicted emotions are recognized and displayed to the user. This could be in the form of textual output (e.g., displaying the predicted emotion label).The sequence ends after the emotion is detected and displayed to the user, completing the emotion detection process.

The code described below outlines how we detect emotions based on their facial expressions using CNN.

#Face Detection with Haar Cascade Classifier:
# Load pre-trained Haar Cascade Classifier for face detection
face_cascade=cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
#Detectfaces in the frame
faces=face_cascade.detectMultiScale(gray,

scaleFactor=1.3, minNeighbors=5, minSize=(30, 30))

# **Perform emotion recognition using the CNN model**
emotion_probs=emotion_model.predict(face_img)
emotion_label=emotions[np.argmax(emotion_probs)]

# **Display the emotion label on the frame and Bounding Box**
label_text = f"Emotion: {emotion_label}"
cv2.putText(frame, label_text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

V. RESULTS AND DISCUSSIONS

The designed model consists of a machine-learning architecture that is operated and launched using a Pycharm. ThedatasetAffectNet, CK+ from the Kaggle competition on FER2013 was the data source used for the application. The framework for detecting facial expressions is integrated using the database. The database comprises 15,887 images total, which are split into 12,298 pictures of trains and 3589 tests. For the final test, the dataset also includes 3589 more private test images.

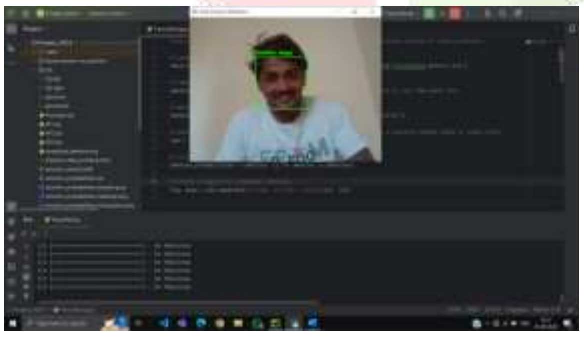

Image-1. The Emotion of the Picture is Happy

Once the main file runs, it will activate the webcam connected to your device.The webcam will start streaming live video.The program has functionality for face detection. It will continuously analyse the video stream to locate human faces.This is a crucial step as it identifies where the face is within the frame.After detecting a face, the program will display an interface, similar to image 1 emotion of the object present in the frameispredicted and the program displays the accuracy of the predicted emotion, specifically the probability of happiness.

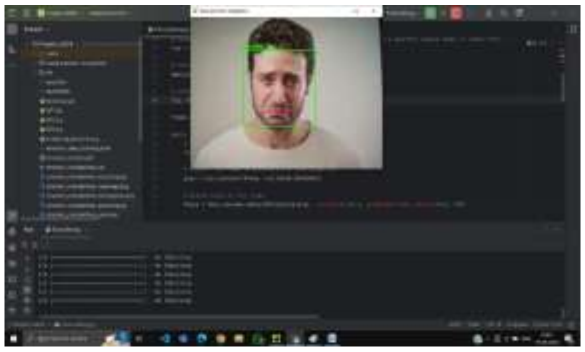

Image-2. The Emotion of the Picture is Sad

Once the main file runs, it will activate the webcam connected to your device.The program has functionality for face detection. It will continuously analyse the video stream to locate human faces.After detecting a face, the program will display an interface, similar to image 2 emotion of the object present in the frameis predicted and the program displays the accuracy of the predicted emotion, specifically the probability of sadness.

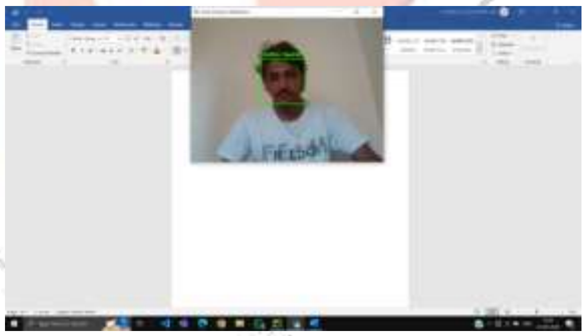

Image-3. The Emotion of the Picture is Neutral

Once the main file runs, it will activate the webcam connected to your device.The program has functionality for face detection. It will continuously analyse the video stream to locate human faces.After detecting a face, the program will display an interface, similar to image 3 emotion of the object present in the frameis predicted and the program displays the accuracy of the predicted emotion, specifically the probability of neutral.

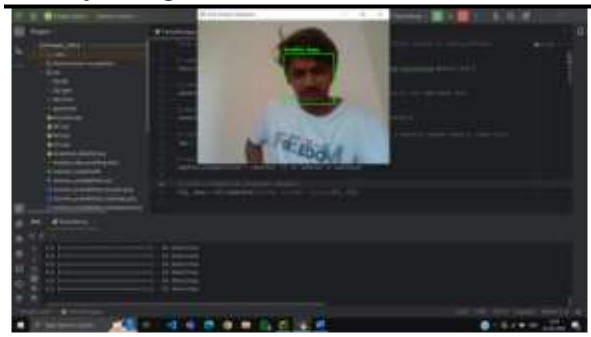

Image-4. The Emotion of the Picture is Angry

Once the main file runs, it will activate the webcam connected to your device.The program has functionality for face detection. It will continuously analyse the video stream to locate human faces.After detecting a face, the program will display an interface, similar to image 4 emotion of the object present in the frameis predicted and the program displays the accuracy of the predicted emotion, specifically the probability of angry.

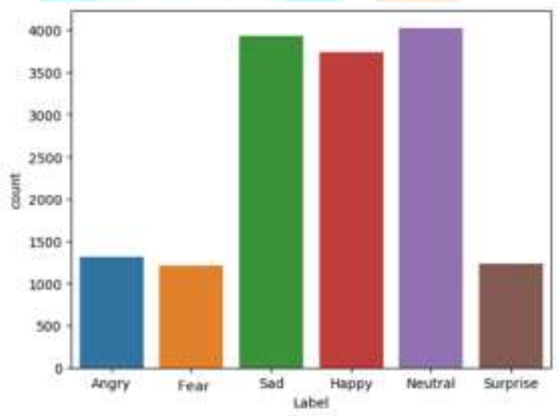

Fig.4 Classification Graph

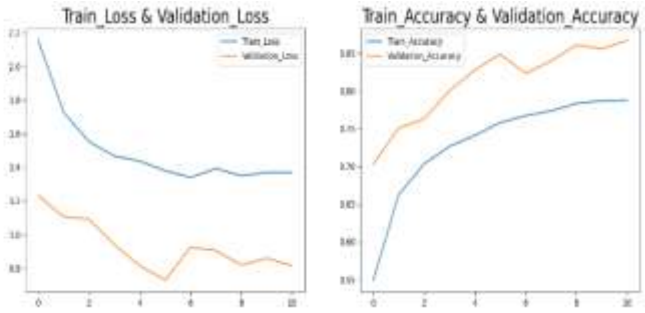

Fig2. Accuracy and Loss

Fig.5Train and Validation graphs

Table1. Accuracy results of Classified features

	Precision	Recall	F1-score	Support
0	0.88	0.94	0.91	240
1	0.69	0.73	0.71	252
2	0.83	0.95	0.89	741
3	0.68	0.72	0.70	815
4	0.76	0.62	0.68	799
5	0.74	0.66	0.70	244

Table2. Total Accuracy results of Classified features.

Accuracy			0.76	3091
Macro avg	0.76	0.77	0.76	3091
Weighted avg	0.76	0.76	0.76	3091

VI. CONCLUSION

In conclusion, the presented script embodies a convergence of cutting-edge technologies, seamlessly blending emotion recognition and iris scanning to create a versatile and insightful system. The utilization of OpenCV, TensorFlow, and specialized libraries underscores the script's commitment to harnessing the potential of computer vision and deep learning in tandem.The integration of a Haar Cascade Classifier for face detection and a CNN model for emotion recognition establishes a robust foundation for real-time analysis of facial expressions. The recorded emotion probabilities across frames offer a dynamic portrayal of human emotions during video interactions, providing valuable insights for applications in diverse fields.Augmenting this emotion analysis, the inclusion of an iris scanner library elevates the system's capabilities to biometric identification. The real-time scanning of irises not only enhances security measures but also introduces possibilities for personalized user experiences. The fusion of these technologies presents a comprehensive solution that extends beyond traditional emotion analysis, contributing to the broader landscape of intelligent systems.The graphical analysis, facilitated by Matplotlib, visually represents the temporal evolution of detected emotions, providing a clear and interpretable output. The script's modular structure and adaptability make it a versatile tool for various domains, including human-computer interaction

scenarios, secure access control, and healthcare applications.As the script seamlessly executes the integration of emotion recognition and iris scanning, it reflects the ongoing trend in AI and computer vision towards holistic, multifunctional systems. This work not only showcases the technical prowess of contemporary technologies but also underscores their practical applicability in real-world scenarios. Moving forward, the script lays the groundwork for further exploration and development in the realm of intelligent systems, where emotion understanding and biometric identification converge for enhanced human-machine interactions.

VI. ABBREVIATIONS

OpenCV: Open Source Computer Vision Library

HAAR: Haar Cascade

CNN: Convolutional Neural Network

ResNet: Residual Network

VIII. REFERENCE

1. Yu Zhenguo, "Emotion Recognition and Application of Learning Interface in Smart Learning Environment [D]", Shandong Normal University, 2019.

2. K Simonyan and A Zisserman, "Very deep learning for large-scale image recognition[J/OL]", Computer Science, 2014.

3. Zhao Yi Feng and Chen Deyun, "Expression Recognition Using Improved AlexNet Network in Robot Intelligent Interactive System[J]", Journal of Robotics, 2022.

4. M. Demirkus, D. Precup, J. J. Clark, and T. Arbel, ‘‘Multi-layer temporal graphical model for head pose estimation in real-world videos,’’ in Proc. IEEE Int. Conf. Image Process., Oct. 2015, pp. 3392–3396.

5. S. Jain, C. Hu, and J. K. Aggarwal, ‘‘Facial expression recognition with temporal modelling of shapes,’’ in Proc. IEEE Int. Conf. Comput. Vis. Workshops, Nov. 2011, pp. 1642–1649.

6. M. H. Siddiqi, R. Ali, A. Sattar, A. M. Khan, and S. Lee, ‘‘Depth camera-based facial expression recognition system using multilayer scheme, ’’IETE Tech. Rev., vol. 31, no. 4, pp. 277–286, 2014.

7. D. Mehta, M.F.H. Siddiqui, A.Y. Javaid. Recognition of emotion intensities using machine learning algorithms: A comparative study. Sensors (Switzerland)**2019**,

8. N. Mehendale. Facial emotion recognition using convolutional neural networks (FERC). SN Appl. Sci.**2020**, 2 (3), 1–8.

9. B. Hdioud, M. El, H. Tirari. Facial expression recognition of masked faces using deep learning. **2023**, 12 (2), 11591.

10. W. Mellouk, W. Handouzi. Facial emotion recognition using deep learning: Review and insights. Procedia Comput. Sci.**2020**, 175, 689–694.

11. Cui, Y.; Wang, S.; Zhao, R. Machine learning-based student emotion recognition for business English class. Int. J. Emerg. Technol. Learn. **2021**, 16, 94–107.

12. Jaiswal, A.; Raju, A.K.; Deb, S. Facial emotion detection using deep learning. In Proceedings of the 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 5–7 June 2020; pp. 1–5.

13. Sha, T.; Zhang, W.; Shen, T.; Li, Z.; Mei, T. Deep Person Generation: A Survey from the Perspective of Face, Pose, and Cloth Synthesis. ACM Comput. Surv. **2023**, 55, 1–37.