ML-Driven Real-Time Emotion and Stress Level Monitoring

*

Bharti Parate
Department of Electronics and Computer Science
Shri Ramdeobaba College of Engineering and Management
Nagpur, India
paratebn@rknec.edu

Diksha Mathankar
Department of Electronics and Computer Science
Shri Ramdeobaba College of Engineering and Management
Nagpur, India
mathankardn@rknec.edu

Richa Khandelwal
Department of Electronics Engineering
Ramdeobaba University
Nagpur, India
khandelwalrr@rknec.edu

Prashant Dwivedy
Department of Electronics Engineering
Ramdeobaba University
Nagpur, India
dwivedyp@rknec.edu

***Abstract*—The growing global prevalence of stress-related disorders has necessitated the development of advanced, non-invasive technologies for real-time stress detection. Traditional stress monitoring methods often require physiological sensors, which, while accurate, can be intrusive and impractical for everyday use. In this paper, we present an innovative approach that leverages deep learning and facial emotion recognition to detect stress levels dynamically and non-invasively. Our system is based on a Convolutional Neural Network (CNN) trained using the FER-2013 dataset, which contains a wide variety of facial expressions. The model is capable of classifying seven core emotions such as anger, disgust, fear, happiness, sadness, surprise, and neutral with high accuracy. Each of these emotions is then mapped to corresponding stress levels (low, moderate, or high), allowing for real-time stress evaluation through facial expressions alone. A key feature of our proposed system is its user-centric design, which includes a real-time visual feedback mechanism. This interface displays stress levels continuously as they are detected, making the system suitable for various practical applications such as mental health monitoring, stress management in workplace environments, and personal well-being tracking. The non-invasive nature of this approach eliminates the need for cumbersome sensors and facilitates its use in dynamic, everyday environments. To validate the effectiveness of the system, extensive experiments were conducted to assess both emotion recognition accuracy and the reliability of stress-level detection. The model achieved a significant level of performance, demonstrating the viability of using facial expressions as a surrogate for physiological stress markers. Additionally, we discuss potential real-world applications, where this system could be integrated into wearable devices or smartphone applications, providing continuous, unobtrusive stress monitoring. The use of deep learning for emotion-based stress detection represents a significant advancement in the field, offering an accessible, scalable solution for addressing the growing public health challenge of stress.**

***Index Terms*—Stress Detection, Deep Learning, Emotion Recognition, Convolutional Neural Network, FER-2013 Dataset, Real-time Monitoring, Non-invasive Technology.**

I. Introduction

Human emotions are fundamental drivers of behavior, playing a critical role in shaping decision-making processes, facilitating social interactions, and influencing overall mental health. In today's rapidly evolving and fast-paced environment, the ability to accurately recognize and interpret emotional states is increasingly vital [1]. This capability is particularly pertinent in applications related to mental health monitoring and stress management, where timely interventions can significantly improve outcomes [2]. Traditional methods of emotion assessment, such as self-reporting and observational techniques, present significant limitations. These methods are often inherently subjective, prone to individual bias, and can be influenced by various external factors [3]. Consequently, the reliance on these techniques may hinder the accurate evaluation of emotional states and stress levels, leading to ineffective management strategies [4]. To address these challenges, this project presents "ML-Driven Real-Time Emotion and Stress Level Monitoring," which aims to develop a robust and efficient system for real-time emotion recognition and stress level monitoring. By harnessing the power of machine learning and advanced computer vision techniques, the proposed system utilizes the FER-2013 dataset a well-established collection of labeled facial expression images, widely recognized within the research community [5]. This dataset serves as a foundational resource for training and validating machine learning models tasked with facial expression recognition. The primary objective of this system is to classify facial expressions into seven distinct emotional categories: anger, disgust, fear, happiness, sadness, surprise, and neutral [6]. Each of these emotional classifications will be systematically mapped to corresponding stress levels, thereby creating a comprehensive framework that enables a nuanced assessment of an individual's emotional well-being. This classification and mapping process is crucial for understanding the complex interplay between emotional states and stress responses [7]. To enhance the user experience and facilitate a clearer understanding of emotional states, the proposed solution incorporates dynamic visual feedback mechanisms. Stress levels will be displayed through an intuitive gauge interface, providing users with immediate insights into their emotional states [8]. This innovative approach not only empowers individuals to monitor their stress levels effectively but also serves as a valuable tool for mental health professionals who require precise, real-time data to inform their therapeutic interventions [9]. Table 1 provides an overview of the total number of data samples present in the FER-2013 dataset, which includes 35,887 labeled facial expression images categorized across seven emotion classes. In summary, the "ML-Driven Real-Time Emotion and Stress Level Monitoring" project aims to bridge the gap between traditional emotional assessment methods and modern technological capabilities. By leveraging the FER-2013 dataset and integrating advanced machine learning techniques, this project aspires to contribute significantly to the fields of affective computing and mental health monitoring, providing insights that can lead to improved emotional well-being and stress management [10].

TABLE I
TOTAL NUMBER OF DATA PRESENT IN FER-2013 DATASET

Micro-Expression	**Public Set**	**Private Set**	**Training Set**	**Dataset (Total)**
Happy	895	879	7215	8989
Neutral	607	626	4965	6198
Sad	653	594	4830	6077
Fear	496	528	4097	5121
Angry	467	491	3995	4953
Surprise	415	416	3171	4002
Disgust	56	55	436	547
Total	**3589**	**3589**	**28709**	**35887**

II. METHODOLOGY

A. *Convolutional Neural Network (CNN) Architecture*

A Convolutional Neural Network (CNN) is employed as the primary machine learning model for emotion recognition due to its high effectiveness in image-related tasks. The CNN model in this project is structured with multiple layers. Figure 1 illustrates the flowchart of the CNN-based stress detection process, detailing the sequential steps from input data preprocessing to emotion and stress level classification. The workflow begins by initializing the project, which involves defining the dataset and getting it ready for training. The FER-2013 dataset, consisting of facial expression images categorized into one of seven emotions, is then loaded. This dataset serves as the model's input. To maintain consistency, the images are resized to 48×48 pixels in grayscale [11]. Data augmentation techniques, such as rotation, flipping, and zooming, are applied to expand the size and variety of the dataset. This step helps prevent over fitting and improves the model's generalization ability. The first part of the model applies Convolutional layers (Conv2D) with Batch Normalization [12]. Batch normalization standardizes the output from the convolutional layers, making training faster and more stable. These layers extract features like edges and textures from the input images, helping the model to learn the facial patterns that correspond to different emotions [13]. After the initial convolutions, Max Pooling is applied to down sample the feature maps. Max pooling reduces the dimensionality of the data while retaining the most important features. This makes the model more efficient by focusing on the dominant features in each region of the image. The next layers are Separable Convolutional layers (Sep-Conv 2D), which are a more efficient form of convolution [14]. They break down standard convolutions into separate operations, reducing computational cost while maintaining accuracy. Batch normalization continues to be applied here to stabilize the learning process. Global Average Pooling is used to further reduce the spatial dimensions of the feature maps. It replaces fully connected layers and averages the features across each feature map, reducing over fitting while maintaining the essence of the features learned by the model [15]. The Softmax layer is applied at the end of the network. It converts the output from the previous layers into probability values corresponding to each emotion class (e.g., happy, sad, angry) [16]. The emotion class with the highest probability is chosen as the predicted emotion. The model is trained over several epochs, where the model's weights are adjusted iteratively to minimize the error between the predicted and actual emotion labels [17]. During training, data augmentation and back propagation help the model improve its performance. After completing each epoch, the model checks if the accuracy has improved. If the model hasn't yet converged, the training continues by increasing the number of epochs [18]. The training process continues until the maximum number of epochs is reached or until the model reaches a satisfactory accuracy level. Once training is complete and the model reaches its best performance, the model is saved. This saved model can be used later for real-time emotion recognition without retraining the model [19].

B. *Algorithm*

In a CNN, the convolution operation is used to extract spatial features from the input image.

$$Output(i,j) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} Input(i|m, j|n) * Kernel(m,n) \tag{1}$$

Input (i, j) represents the pixel value at location (i, j) of the input image. Kernel (m, n) is the filter (also called kernel) of size $k \times k$. The summation slides the kernel over the image, computing a weighted sum of pixel values, which is then stored in the output feature map. The Rectified Linear Unit

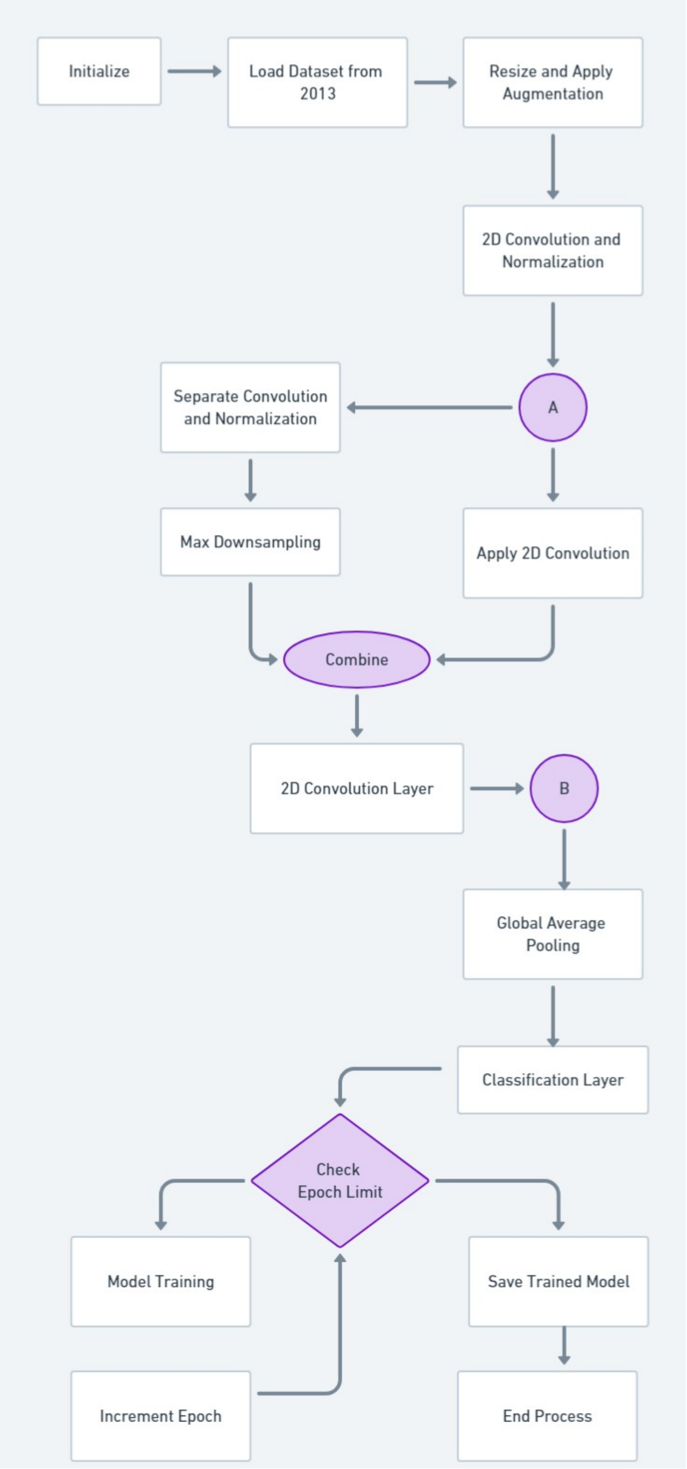

Fig. 1. CNN Flowchart for Stress Detection

(ReLU) activation function is applied after the convolution operation to introduce non-linearity [20].

$$ReLU(x) = max(0, x) \tag{2}$$

where, x is the output from the convolution operation. The ReLU function returns 0 for negative values and keeps positive values as they are. Pooling is used to down sample the feature maps, reducing the computational complexity and helping the network focus on the most important features [21].

$$MaxPool(i, j) = max(Input(i + m, j + n) : m\epsilon[0, p], n\epsilon[0, p]) \tag{3}$$

where, $p \times p$ is the size of the pooling window, and MaxPool(i, j) selects the maximum value from each region of the input. MaxPooling2D reduces the spatial dimensions of the feature maps by taking the maximum value from each 2×2 window. Once the convolution and pooling operations are completed, the resulting $2D$ feature maps are flattened into a $1D$ vector to be fed into the fully connected layers. If the feature maps have a shape of (h, w, d), flattening turns it into a $1D$ vector of size $h \times w \times d$. Fully connected layers apply a linear transformation followed by a non-linear activation function.

$$Output = Activation(W * x + b) \tag{4}$$

where, W is the weight matrix, x is the input vector (flattened output from previous layers) and b is the bias vector. The activation function can be ReLU or Softmax, depending on the layer. The final layer uses a softmax activation to predict the probabilities of each emotion class. The softmax function is used in the output layer to convert the raw class scores into probabilities.

$$Softmax(z_i) = e^{z_i} / \sum_{j-1^{e^{z_i}}}^{K} \tag{5}$$

where, z_i is the score for class i, K is the total number of classes (7 in the $FER - 2013$ dataset) and e^{z_i} represents the exponential of the score for class i. The softmax function ensures that all the output values are between 0 and 1 and sum to 1, effectively providing class probabilities. Since this is a classification problem, the loss function used is categorical cross-entropy, which measures the difference between the predicted probabilities and the true labels.

$$Loss = -\sum_{i=1}^{K} y_i * log(p_i) \tag{6}$$

where, y_i is the true label (1 if the sample belongs to class i, 0 otherwise) and p_i is the predicted probability for class i. The loss function is minimized during training to improve the model's accuracy. The model is trained using the Adam optimizer, which is an extension of the stochastic gradient descent (SGD) algorithm.

$$\theta_{t-1} = \theta_t * \eta * \frac{m_t}{\sqrt{v_t}|\epsilon} \tag{7}$$

where, θ_t represents the model parameters at step t, η is the learning rate, m_t and v_t are the first and second moments of the gradient (mean and variance, respectively) and ϵ is a small constant to prevent division by zero. It uses first-order momentum and adaptive learning rates to adjust the weights and minimize the loss function.

C. CNN Training Accuracy

In machine learning, particularly in the training of Convolutional Neural Networks (CNNs), an epoch refers to one complete cycle through the entire training dataset. Table 2 presents the training accuracy of the CNN model evaluated over multiple epochs, demonstrating the performance improvement during the training process. The model processes each training example once during each epoch and adjusts its internal weights to minimize the error or loss. This process is repeated over multiple epochs to progressively improve the model's performance. In this, where a CNN is being trained to recognize emotions from the FER-2013 dataset, each epoch represents an iteration of the model learning from the data. The performance of the model, in terms of accuracy, improves as the training progresses over these epochs. During the first epoch, the model has just started learning from the data. The accuracy is quite low (24.39) because the model is still initializing its weights and learning basic features from the input images. At this point, the CNN may have only recognized very simple patterns such as edges and contrasts, but it hasn't learned to effectively classify emotions yet. By the seventh epoch, the model has started to learn more meaningful features from the images. The accuracy has significantly improved to around 55.75 percentage, indicating that the model is getting better at identifying patterns in the data and correctly classifying emotions. At this stage, the CNN is likely learning more detailed facial features like the shape of the eyes, mouth, and overall face structure. By the fifteenth epoch, the model accuracy is around 63.07%. The CNN has learned to recognize more complex and abstract features in the images, and it is making more accurate predictions. The model is likely starting to converge, meaning that the rate of improvement in accuracy is slowing down compared to earlier epochs. After 20 epochs, the model has reached an accuracy of 69.9%. This indicates that the CNN has continued to improve and is becoming more proficient at classifying emotions. However, the rate of improvement has slowed compared to the earlier epochs, suggesting that the model is nearing the optimal weights for classifying the training data effectively. By the twenty-fifth epoch, the model's accuracy has increased to 73.21%. The model is performing well, and further training might yield only marginal improvements in accuracy. This is a good point to consider early stopping if the accuracy plateaus, as continuing training beyond this point may result in over fitting where the model starts to memorize the training data rather than generalizing to unseen data. Pre-processing Stage: Both segmentation and edge detection are applied during the pre-processing stage of the pipeline, where the face is detected and features like eyes, nose, and mouth are enhanced through edge detection techniques. Feature Extraction: After applying edge detection, the CNN extracts deeper features from the segmented images, capturing the most relevant patterns that correspond to different emotions. One of the most common edge detection techniques is Canny edge detection, which is used to identify sharp changes in pixel intensity that correspond to edges in the image. In this, Canny edge detection can highlight key facial features that are essential for emotion recognition. By emphasizing the boundaries around the eyes, mouth, and nose, the model can better understand the structure of the face and how it changes with different emotions. Segmentation is the process of dividing an image into meaningful regions, often focusing on specific areas like the face or key facial features (eyes, mouth, etc.). In your project, segmentation helps isolate the face from the rest of the image, ensuring that the model focuses only on relevant areas. Face Detection: The project typically applies face detection algorithms (like Haar Cascades or MTCNN) to segment and crop the face from the background. This serves as the primary segmentation technique to ensure only facial features are processed. For instance, using OpenCV's Haar Cascades or deep learning-based models like MTCNN (Multi-task Cascaded Convolutional Networks) can effectively segment out the face in real-time video feeds. Region of Interest (ROI): After detecting the face, further segmentation can focus on specific regions, like eyes, nose, and mouth, which are key for identifying micro-expressions and emotional cues. By focusing on these areas, the model improves its ability to differentiate between subtle expressions.

TABLE II
CNN Training Accuracy Over Epoch

CNN	***Epoch 1***	***Epoch 7***	***Epoch 15***	***Epoch 20***	***Epoch 25***
Accuracy	0.2439	0.5575	0.6307	0.699	0.7321

III. Test Results

The image depicts the output of an emotion and stress level detection system. The person in the image is smiling, which seems to reflect positive emotions such as happiness or contentment. Bounding Box, there is a red rectangular box around the person's face. This is likely a result of face detection, which isolates the face from the rest of the image, allowing the model to focus on analyzing the facial features for emotion recognition. 'Happy' the model has classified the emotion based on facial features (e.g., the smile) as "Happy". This classification is part of the real-time emotion detection system, which assigns one of several possible emotion labels (e.g., happy, sad, angry) based on trained data from a dataset like FER-2013. Figure 2 highlights the specific test cases used to evaluate the CNN model's effectiveness in stress detection, detailing input scenarios, expected outcomes, and actual results.

'Low Stress': Along with emotion detection, the system also estimates the stress level. Based on the happy emotion, the system predicts a "Low Stress" level, indicating that this individual is likely experiencing little to no stress. In this project, AI-generated images are utilized to detect stress levels by analyzing facial emotions. Different emotions such as "Disgust", "Happy", "Fear", "Surprise", "Neutral", and "Sad" are classified, and corresponding stress levels (ranging from

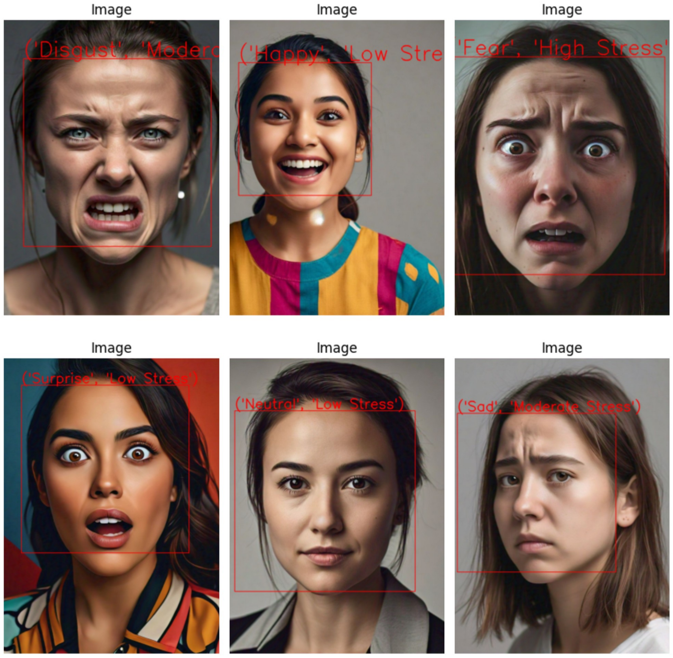

Fig. 2. CNN Test Cases for Stress Detection

low to high) are associated with each emotional state. This system leverages image-based emotion recognition to provide an automated, visual assessment of stress, which could be useful in mental health applications, real-time monitoring, and stress management interventions. Once the bounding box isolates the face from the rest of the image, the system can focus its analysis on the contents of the box, i.e., the facial features. These facial features are then passed to a Convolutional Neural Network (CNN) for emotion classification (such as "happy" in the image example). The bounding box helps ensure that the analysis is only performed on the facial area, reducing noise from the background or other irrelevant parts of the image. In the context of your project, after detecting the emotion within the bounding box, the system further classifies the stress level based on that emotion (e.g., "low stress" for a happy face).

Figure 3 presents a bar graph for visual understanding, providing a comparative representation of key metrics or results related to the study. The bounding box helps in focusing solely on the facial features for stress detection. The graph shown represents the emotion prediction confidence for an image analyzed by a machine learning model trained to classify emotions. The x-axis lists different emotion categories, while the y-axis shows the confidence score (ranging from 0 to 1) for each emotion, representing how likely the model thinks the detected face corresponds to a specific emotion.The model predicts happiness as the dominant emotion in the image with a confidence score of $0.6(60\%)$. The graph shown represents the emotion prediction confidence for an image analyzed by a machine learning model trained to classify emotions. The x-axis lists different emotion categories, while the y-axis shows the confidence score (ranging from 0 to 1) for each emotion, representing how likely the model thinks the detected face corresponds to a specific emotion.

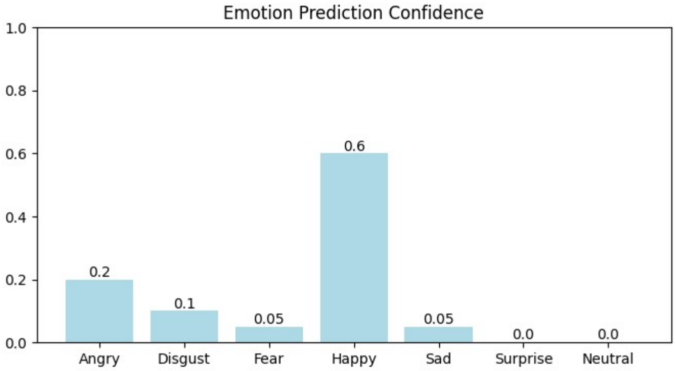

Fig. 3. Bar Graph for Visual Understanding

Figure 4 illustrates a dynamic visual representation of a box with a needle, used to depict the measurement or indicator for stress level in the study. A key feature of this system is the real-time dynamic visual feedback that enhances user interaction. The predicted stress levels are visually represented using The gauge displays stress levels ranging from Low to High. A needle moves their position between these levels, updating in real-time as new emotions are detected from the video feed. This visual feedback provides immediate, easy-to-understand insights into an individual's stress level, making the system useful for applications in mental health monitoring and stress management.

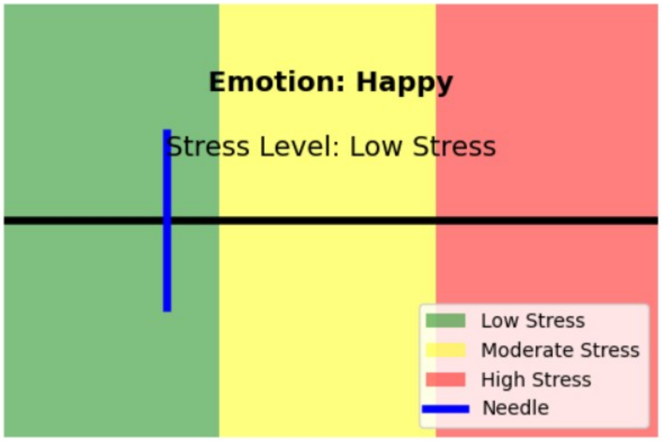

Fig. 4. Dynamic visual box with needle

IV. Conclusion

The stress detection system based on emotion recognition provides a powerful and non-invasive solution for monitoring stress levels in real time. By leveraging AI-generated images and advanced facial emotion analysis, the project demonstrates a novel approach to assessing stress without the need for physical or physiological sensors. The mapping of emotions such as fear, happiness, and sadness to different stress levels creates a comprehensive framework for identifying stress-related patterns. This technology holds great potential in fields like healthcare, workplace management, and smart environments, where stress can impact productivity and well-being. The use of AI-generated images not only enhances data diversity but also improves the accuracy of the detection models, making them more robust and adaptable. As this project evolves, it could be expanded to cover a wider range of emotions and contexts, offering even greater insights into emotional health and stress management.

References

[1] Uday C. Patkar, et.al., "An Hybrid and Syntactic Machine Translation Model for English to Ahirani Language," *International Journal of Grid and Distributed Computing*, vol. 14, no. 1, pp. 976-989, 2021.

[2] XuC., et. al., "A novel facial emotion recognition method for stress inference of facial nerve paralysis patients," *Expert Systems with Applications*, vol. 197, pp. 1-8, 2022.

[3] S.K. Meena, et.al., "Review and application of different contrast enhancement technique on various images," *2017 IEEE 1st International conference on electronics, materials engineering and nano-technology (IEMENTech)*, pp. 1-6, 2017.

[4] Hsin-Yen, Huei-Ling Chiu, "Virtual reality exergames for improving older adults' cognition and depression: a systematic review and meta-analysis of randomized control trials," *Journal of the American Medical Directors Association*, vol. 22, no. 5, pp. 995-1002, 2021.

[5] K. Sengupta, "Stress Detection: A Predictive Analysis," *2021 IEEE Asian Conference on Innovation in Technology (ASIANCON), Pune, India*, pp. 1-6, 2021.

[6] A. Bannore, T. Gore, A. Raut, K. Talele, "Mental stress detection using machine learning algorithm," *2021 IEEE International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pp. 1-4., 2021.

[7] S. K. Kanaparthi, et.al., "Detection of Stress in IT Employees using Machine Learning Technique," *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, pp. 486-493, 2022.

[8] Ankita Patil, Rucha Mangalekar, Nikita Kupawdekar, Viraj Chavan, Sanket Patil, Ajinkya Yadav, "Stress Detection in IT Professionals By Image Processing And Machine Learning," *International Journal of Research in Engineering Science and Management*, vol. 3, no. 1, 2020.

[9] M. Hariprasath, "Detection of Stress by Machine Learning in IT Industry," *2023 7th IEEE International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 53-56, 2023.

[10] S. Gedam, S. Paul, "Automatic Stress Detection Using Wearable Sensors and Machine Learning: A Review," *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India*, pp. 1-7, 2020.

[11] S. Pandey, et.al., "A survey on key frame extraction methods of a MPEG video," *2017 International Conference on Computing, Communication and Automation (ICCCA), IEEE*, pp. 1192-1196, 2017.

[12] P. Dwivedy, et.al., "Comparative study of MSVD, PCA, DCT, DTCWT, SWT and Laplacian Pyramid based image fusion," *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), IEEE*, pp. 269-273, 2017.

[13] Jorge Rodríguez-Arce, et.al., "Towards an anxiety and stress recognition system for academic environments based on physiological features," *Computer Methods and Programs in Biomedicine*, vol. 190, 2020.

[14] P. Mukherjee, A. H. Roy, "Detection of Stress in Human Brain," *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), Gangtok, India,* pp. 1-7, 2019.

[15] A. Agarwal, et.al., "A Novel Model for Stress Detection and Management using Machine Learning," *2023 International Conference on Disruptive Technologies (ICDT), Greater Noida, India*, pp. 562-570, 2023.

[16] U. Bhade, et.al., "Comparative study of DWT, DCT, BTC and SVD techniques for image compression," *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), IEEE,* pp. 279-283, 2017.

[17] M. K. Moser, B. Resch, M. Ehrhart, "An Individual-Oriented Algorithm for Stress Detection in Wearable Sensor Measurements," *IEEE Sensors Journal*, vol. 23, no. 19, pp. 22845-22856, 2023.

[18] P. Dwivedy, et.al., "Performance comparison of various filters for removing different image noises," *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), IEEE*, pp. 181-186, 2017.

[19] H. Ali, "Facial emotion recognition based on higher-order spectra using support vector machines," *Journal Of Medical Imaging And Health Informatics*, vol. 5, no. 6, pp.1272-1277, 2015.

[20] P. Dwivedy, et.al., "A survey on visible light communication for 6G: Architecture, application and challenges," *2023 IEEE International Conference on Computer, Electronics and Electrical Engineering and their Applications (IC2E3)*, pp. 1-6, 2023.

[21] Lutfiah Zahara, et al., "The facial emotion recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based Raspberry Pi," *2020 Fifth International Conference on Informatics and Computing (ICIC), IEEE*, pp. 1-9, 2020.