ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 6, June 2025

Employee Stress Identification in the IT Industry Using Deep Learning Models

[1] Vemuri Gayathri, [2] Dr. Appala Srinuvasu Muttipati

[1] Department of Computer Science and Engineering (Data Science), Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India
[2] Associate Professor, Department of Computer Science and Engineering (Artificial Intelligence & Machine Learning), Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India
Corresponding Author Email: [1] vemurigayathri.23mtech.csd@anits.edu.in [2] srinuvasum.csm@anits.edu.in

Abstract*— Stress is uncertain, physiological, and psychological constituents that impact human attainment and decrease an individual's lifespan. Accurate detection of stress through facial expressions is unavoidable, otherwise, it leads to severe health problems including cardiovascular problems, shortening immune system, and premature mortality. Stress detection was implemented using facial expressions through image processing, data augmentation, and deep-learning classification techniques. In this research, a Real-world Affective Database (RAF-DB) is an image dataset from Kaggle that was utilized. Three deep learning models: Convolutional Neural Network, Densenet, and a combination of Efficient Net and Squeeze-and-Excitation were used for transfer learning from the pre-trained initial weights and were trained and tested for classifying facial emotions. The accuracy, precision, recall, f1-score, and support for best-performing models. Convolutional Neural Networks in the detection of stress outperformed apart from the other deep learning models; the accuracy for CNN was 85.43%. This paper also used visualizations to analyze and understand the image data, resulting in improved accuracy.

Keywords: Convolutional Neural Network, Data Visualizations, Deep Learning, Efficient Net, Squeeze-and-Excitation.

I. INTRODUCTION

Stress is an apprehension of impact or pressure. In day-to-day activities, stress has become part and parcel of every IT employee. As new technologies emerge rapidly, there is a great scope for new product launches every time. To establish the new projects, it needs to be submitted by the specified deadlines which is a cause for the improvement of stress levels. About 284 million people worldwide experience stress and mental disorders of different age groups as indicated by the World Health Organization [1]. It not only affects the mind but also is a reason for health problems such as heart attack, high blood pressure, obesity, headache, anxiety, and depression. This provokes the earnestness of stress detection.

In the traditional approach, counseling is widely preferred while dealing with stress which manifests as time-consuming. In the real world, to sustain as a software developer, engineer, or IT employee, the hired person must be up to date with the newest innovations. It clearly shows competition between the employees and modernizes themselves at one's convenience. In some scenarios, there was a chance of a lack of communication, trust issues, and so on which gives rise to mental problems such as anxiety, depression, and frustration. If it is not treated correctly, it has a huge impact on society and the economy [2].

Stress can be rendered through facial emotions which are ubiquitous of reference. In any workplace environment, stress was mocked up with the help of filling out surveys [3], the Perceived Stress Scale [4], and some set of questionnaires. Enhancing, the facial gestures or looks, the emotion can be triggered which is referred to as Facial Emotion Analysis. To attain the consequential stress characteristics, the features of that face need to be extracted by Deep learning a subset of Artificial Intelligence. This emphasizes a practical solution for a stress detection system.

To give a talk on the need for a stress detection system, this research proposes a deep learning-based approach to facial expression analysis. For a better understanding of facial emotions, the RAF-DB dataset is used. Three deep learning models: Convolutional Neural Network, Densenet, and a combination of Efficient Net and Squeeze-and-Excitation – predict facial emotions such as anger, happiness, sadness, disgust, surprise, fear, and neutrality. The main goal of this research is to build an AI-based tool that helps IT companies by preparing their employees to deal with stressful events before they occur.

II. LITERATURE SURVEY

Suresh Kumar Kanaparthi, et. al [1] developed the detection of stress in IT employees using a machine learning technique. The author mainly focuses on stress management and lays the foundation for a healthy, impulsive work environment. They applied various machine learning and visual processing to pinpoint overworked IT employees. K-Nearest Neighbor classifier is utilized for detecting facial emotions.

A Stress Reduction session was held against 55 refugees, to convey the stress classification by Dorota Kaminska [3]. The author focuses on taxonomy with the assistance of

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Volume 12, Issue 6, June 2025

EEG/GSR signals which is a treasured tool for health care professionals. It utilizes the machine learning models, prompting 86.7% for stress level classifications.

To foreground the stress issues among experts and understudies, Disha Sharma and Sumit Chaudhary [4] introduced a stress detection system. The author mainly intends to identify the stress in undergraduates and postgraduate students. To contribute to this, the researchers asked the basic questions among 220 undergrad and postgrad students that are related to the situations and how they reacted from the last month. The stress levels are categorized into three types: Low, Medium, and High. Support Vector Machine, K Nearest Neighbor, Bayes Net, Random Forest, Naive Bayes, Logistic Regression, and Decision Tree were employed. Baye's Net outclassed when compared to the other machine learning algorithms.

To design a stress detection system, Abdul Gaffar Khan and Dr. D. Krishna Madhuri [5] foremostly used the features to recognize stress, and then they predicted the stress levels in postgraduates and undergraduate students. They preferred basic machine learning algorithms such as Support Vector Machine, K – Nearest Neighbor, Random Forest, Naïve Bayes, Logistic Regression, and Decision Tree.

Sleeping habits play a major part in identifying stress, Lakshmi Shree M S, et.al [6], by making use of machine learning algorithms. The stress levels can be attained or calculated based on the Snoring time, Oxygen percentage in blood, body temperature, movements of an eye, and so on. The stress levels that were classified are as follows: 0 – low/normal, 1 – Medium-low, 2 – Medium, 3 – Medium-high, and 4 – High. To obtain the relationship between sleeping patterns and stress levels, the random forest and decision tree were used.

III. PROPOSED SYSTEM

The main key for identifying stress in humans can be achieved by facial emotion analysis. An emotion tells about the mood swings of an employee at work. The proposed system was well-designed with the help of facial emotions by taking the RAF-DB dataset into consideration. First, the dataset was scrutinized through image processing. Owing to this, the deep learning models handle numerical data efficiently and quickly. An image contains small units called pixels. These pixel values enhance the image quality, color, and brightness which are applicable for analysis of the models and refining by computers. The proposed system contains the following steps: Image preprocessing, Data Visualization, Data Balancing, Splitting, Deep Learning model, Model Training, Evaluation, and Predictions. The visual representation of the proposed system is shown here:

IV. SYSTEM ARCHITECTURE

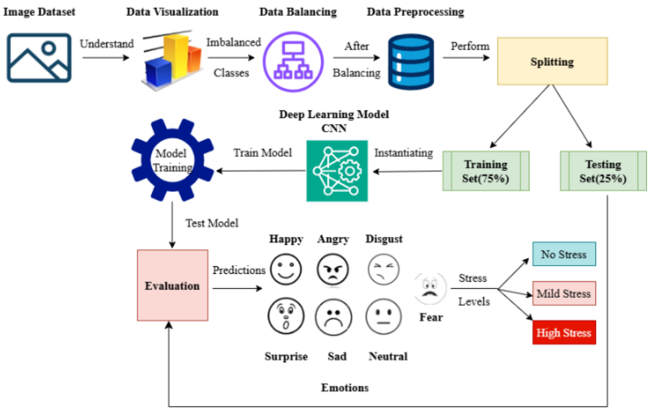

Figure 1: System Architecture for Stress Detection System

Figure 1 shows the system architecture for predicting stress based on facial emotions.

Data Collection:

The Real-world Affective Face Database (RAF-DB) contains 15,000 facial images, each of which is annotated with emotion labels derived from crowdsourced public opinion. Basic emotions and compound emotions are the subsets of the dataset. The most frequently used RAF-DB focuses on the basic emotion subset, which includes seven classes: anger, disgust, fear, happiness, sadness, surprise, and neutral. The images were stripped of personal or identifiable information. The data is not altered which contains facial and emotional labels.

Data Visualization:

As the data accommodates seven emotions, each emotion needs to be processed clearly for the efficient building of AI-driven systems. To analyze, interpret, and understand the RAF-db dataset data visualization plays an important role in deep learning, which remoulds complex datasets into visual formats, that the human brain can easily understand. It not only breaks a great deal of data into uncomplicated formats but also shows data inconsistencies, class imbalances, outliers, and patterns with the datasets. Investigating the data, Bar charts and Pie charts are the widely used visualization techniques.

Bar Chart:

To check whether each emotion consists of the same number of images that hold a unique category, a bar chart was employed. It is represented in the rectangular format where the x-axis signifies the categories of nothing but the emotions and the y-axis constitutes the values that are related to each category of emotion. This practically evokes the emotions of classes and the number of images that were present in the respective class.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 6, June 2025

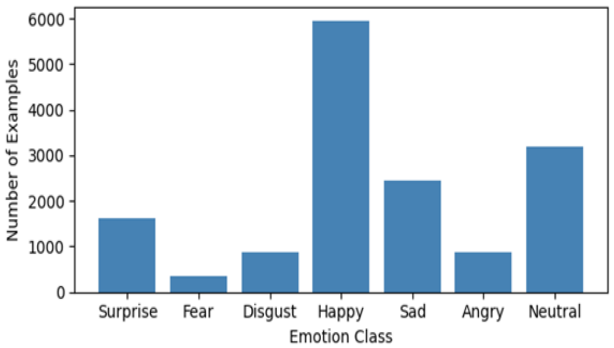

Figure 2: Count of each Emotion class of the RAF-DB dataset

Figure 2 shows that the happy emotion contains more examples when it was compared with the other classes. It proves that the data was completely imbalanced.

Pie Chart:

The pie chart exhibits the data in a circular format. It segregates the different categories. It represents the simple data and offers a clear comparison of proportions.

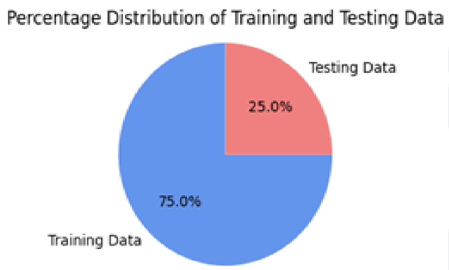

Figure 3: Splitting of the data after data is balanced

Figure 3 shows the split after the data is stabilized, training data (75%) and testing data (25%).

Data Balancing:

As the bar chart mentions the data unbalancing between the emotion classes, is mandatory to maintain the level of each emotion. If the dataset is imbalanced, then the specified model learns more about the majority class which means the happy class in this scenario. Each emotion must be treated equally. Increasing the size of all the classes, augmentation is preferred. In this research, for each emotion, 3500 image samples were considered.

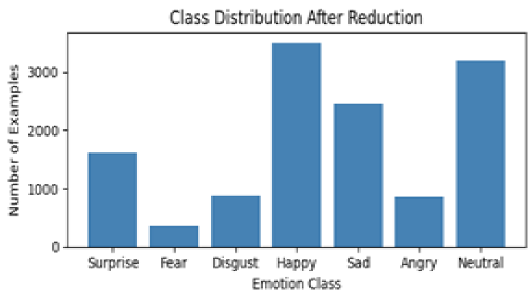

Figure 4: Reduction of image samples for Happy class emotion.

Figure 4 shows the reduction of image samples for happy class emotion because it contains more image samples i.e. 4772. So, reducing it is a must in this case.

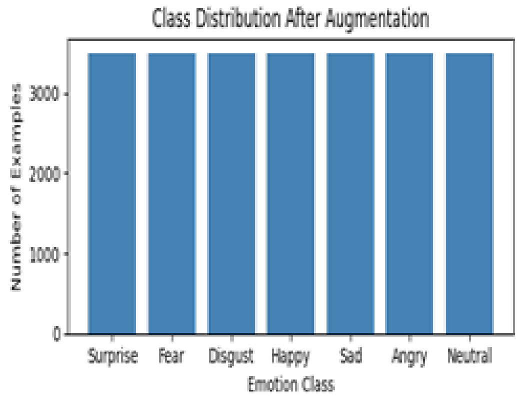

Figure 5: Augmenting all the emotions except the Happy class

V. DATA PREPROCESSING

In image processing, converts unrefined image data into a suitable and relevant format. After all the emotions are changed uniformly, it is mandatory to remove the unwanted distortions that enhance the image characteristics. It is the primary step in image processing. Applying preprocessing techniques on an image such as Normalizing, Resizing, converting labels to categorical, and Data Augmentation.

Resizing:

Resizing is useful for reducing or increasing the size of the images. It neither eliminates the important information nor misrepresents the image ratio. It adjusts the dimensions of the images. While dealing with the deep learning models, especially for CNN require the same input size of images. It fastens the training of the model.

Normalizing:

In an image, the foremost integral is a pixel. Each pixel represents its gray levels which shows how bright or dark a pixel is. The process of adjusting the gray levels to a desired range, between 0 and 1 is called normalization. After resizing, the normalization is performed which leads to more stable convergence during the optimization. It not only makes the data consistent but also reduces the lighting conditions too.

Converting the Labels into Categorical:

The model acknowledges easily, by converting the numerical data into categorical format by using Python libraries and functions. Classification can be performed with the help of output labels that is emotion in neural network. Each class is proportionate with a unique category by employing categorical encoding to transform class labels into a categorical format.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 6, June 2025

Data Augmentation:

It is an image preprocessing technique where the image data can be equally distributed when it has different categories. It generates the new image data by applying the transformations without modifying the original images. The transformations that are applied in this phase are: Rotation_range, width_shift_range, height_shift_range, vertical_flip, horizontal_flip, and fill_mode.

Rotation_range: Rotates the image into a certain angle that is specified degrees.

Width_shift_range: Shift the image to left or right horizontally.

Height_shift_range: shift the image top or bottom vertically.

Vertical_flip: It flips the image vertically during the training.

Horizontal_flip: It flips the image horizontally during the training.

flip_mode: After the image is rotated and flipped, the empty spaces in the image are filled. If the mode is nearest, then it fills with the nearest common value.

Splitting:

The image dataset was subdivided into two parts: The training dataset and the Testing dataset. The training dataset was considered 75% and the testing dataset was 25%. Each dataset contains seven subfolders that represent the emotions and the images concerning the specific class were stored in it.

Deep Learning Model:

An image contains important features such as edges, colors, corners, textures, and shapes. These characteristics are learned from the multiple layers of deep learning models with the help of image pixels. The models prefer the small details first, then bigger patterns, and finally decide which class it belongs to. Three deep learning models are used: Convolutional Neural Network, DenseNet, and EfficientNet+Squeeze Extractor. The main model is CNN.

Model Architecture:

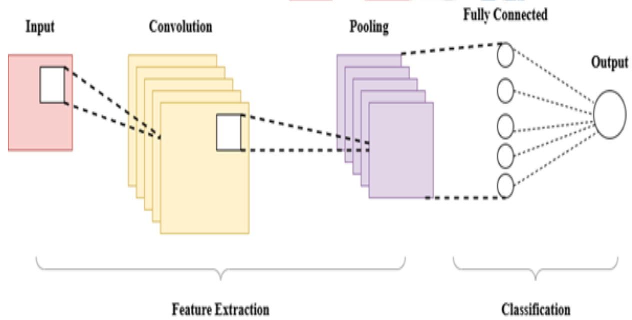

Figure 6: Model Architecture for Convolutional Neural Network.

Figure 6 shows the different layers of the convolutional neural network starting from input layer to the output layer in deep learning.

Convolutional Neural Network:

For image classification tasks, the convolutional neural network is utilized. it is useful for efficiently extracting the features from the input data. It fundamentally recognizes the edges or shapes and then puts them together to get the classification or the output. Layers in a convolutional neural network:

- Input Layer
- Convolutional Layer
- Pooling Layer
- Fully Connected Layer
- Output Layer

Input Layer:

A 3D array of numbers is taken as an input for the input layer. The input shape for the RAF-dB dataset was (100,100,3). The input shape is calculated as follows:

$$Input\ shape = (H\ x\ W\ x\ C)$$

Where,
H = Height (number of rows in pixels),
W=Weight (Number of columns in pixels),
C=Number of Channels,
Therefore,
For RGB images of size 100 X 100: Input Shape =100 x 100 x 3= 30,000-pixel values were fed into the input layer.

Convolutional Layer:

In this layer, the pixel values or features were taken from the input layer. It identifies the basic patterns such as edges and shapes can be done using filters. A new feature map is created from each filter. The layer contains the convolutions, kernel (filters), input, padding, and strides.

Kernel: It finds out the features in the image.

Stride: It looks over how pixels move at each step.

Padding: Equalizing the input and output dimensions.

Formula:

$$Hout = \frac{Hin - F + 2P}{S} + 1$$

$$Wout = \frac{Win - F + 2P}{S} + 1$$

Where,
Hout = Output Height
Wout = Output Weight
Hin = Number of the rows in pixels
Win = Number of the columns in pixels
F= filter with 3x3
P = Padding (By default it is 0 if it is not mentioned)
S = Stride (By default it is 1)

Pooling Layer:

In this layer, the required features were continued by decreasing the height and weight. Extracting the features reduces the possibility of overfitting. The sludge is with the 2x2 used.

Fully Connected Layers:

In fully connected layers, all the features that were abstracted from the preceding layers were integrated. It connects every neuron from one layer to every neuron in the succeeding layer. One-dimensional vectors of feature maps are created from multi-dimensional vectors. The 1D vector is passed through the various fc layers, which are used for classification tasks.

Formula:

$$Zi = \sum_{i}^{n}(Wij\ x\ Xi) + bi$$

Where,
Wij = it is the weight matrix,
Xi = Flattened vector from the preceding layer,
Bi = Bias,
N= number of neurons,
i = Number of the steps.

Dropout Layer:

In this layer, without any prearrangement neurons were made inactive to avoid overfitting. It makes the model more robust by encouraging it to learn disturbed representations.

Formula:

$$\tilde{x} = mask * X * \frac{1}{1 - dropout\_rate}$$

Where,
dropout_rate is the probability of the dropout layer,
The mask is the binary mask with 1 with probability (1 - dropout_rate) and 0 with probability dropout_rate,
X is the input tensor.

Activation Function:

It facilitates to representation of difficult relationships in the data. It put forward the non-linearity of the model. The Rectified Linear Unit sets the Zero for the negative values and the input for positive values.

Formula:

$$RELU = \max(0, x)$$

X is the input to the neuron,
If $X > 0$ then the X value remains the same, otherwise it sets to Zero.

Output Layer:

It is the finishing layer in the Convolutional Neural Network. It generates final predictions after all the preceding layers. It is useful for solving the classification tasks. SoftMax function is used in the output layer, as it is a multiclass classification which means more than 2 classes.

Formula for SoftMax:

$$Softmax\ (Zi) = \frac{e^{zi}}{\sum_{j=1}^{n} e^{zj}}$$

Where,
Zi is the score for class i,
n is the number of classes.

DenseNet:

DenseNet (Dense Convolutional Network) is a deep-learning model in which every layer acquires the output of the feature maps related to all preceding layers as an input. For performing the classification tasks, a pre-trained model of DenseNet121 was employed, and a custom dense layer was passed with it. The model used weights learned from the ImageNet dataset and was useful for learning features such as edges, texts, shapes, and patterns. working the learned knowledge not only saves time but also requires less data. This process is called Transfer learning. To make the model simpler, the Global Average Pooling 2D was utilized where each feature map average was taken as a single value. It is efficient and faster training. A dense layer with 512 neurons was attached and the activation function RELU was applied after the dense layer to learn hidden patterns. The Dropout layer will randomly drop out 40% of neurons during training which prevents overfitting. To predict the emotions, each class of emotion relates to each neuron and the number of classes is equal to the number of neurons. To attain the highest probability of the prediction, SoftMax activation was implemented where each class gets the probability between 0 and 1. The custom output layer and input layer were linked to each other for model training.

EfficientNet+Squeeze Extractor:

To enhance the model's efficiency, integrating the two powerful deep-learning techniques was employed such as EfficientNetV2S and Squeeze Extractor. A peculiar version of EfficientV2S, where “S” recommends the small variant. The features such as objects were abstracted from the input images. It performs hierarchical representations and several convolutions at individual levels of an image. The backbone for the squeeze extractor is an efficientnet where the raw features are extracted. The output of EfficientNetV2S is taken as an input for the SE block. **Squeeze:** for an image, the dimensions such as height, width, and depth of the image are figured out into feature maps. The feature maps are then converted into a single value using Global Average Pooling. Each channel's information is collected by taking the average of all values. The values were passed into the dense layer which modernizes the original number of channels. **Extractor:** The significance of each channel is learned here by reducing the channels or dimensionality with the help of fully connected layers and activation function RELU. The

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 6, June 2025

sigmoid function is applied after restoring the size, which changes the value range between 0 and 1. For instance, if the output of the sigmoid is 0.80, then the channel has 80% of importance.

Formula for the Sigmoid Function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Where,
$\sigma(x)$ for the input x, the sigmoid function output,
e is the Euler's Number,
x is the input for the function of the sigmoid.

VI. RESULTS

The proposed system has been designed with three deep learning models: a Convolutional Neural Network, a Dense Net, and an EfficientNet+squeeze Extractor. The Convolutional Neural Network performed better than the other deep learning models. The evaluation metrics taken as key considerations for the comparison are Accuracy, Precision, Recall, F1 score, Support, and an ROC curve. The Convolutional Neural Network attained an accuracy of 85.43%.

Accuracy: It shows how often a machine predicts outcomes correctly.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} = \frac{TP + TN}{TP + TN + FP + FN}$$

Table 1: Comparison of training accuracy and testing accuracy of the deep learning models

Model	Training Accuracy	Testing Accuracy
Convolutional Neural Network	90.69	85.24
DenseNet	77.15	66.78
EfficientNet + Squeeze Extractor	80.68	77.54

Table 1 shows that the convolutional neural network outperformed apart from the DenseNet and EfficientNet + Squeeze Extractor. Even though the DenseNet was developed based on the limitations of CNN and combining the two powerful techniques of deep learning models, CNN accomplished the best part in the stress detection system

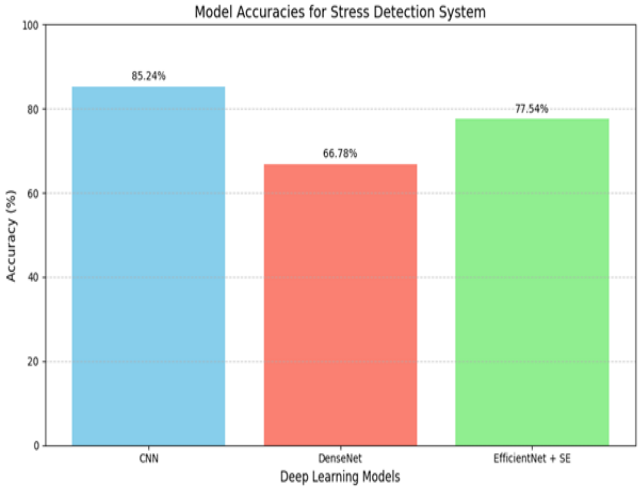

Figure 7: Comparison of three deep learning models in terms of accuracy

Figure 7 shows that the convolutional neural network attained the best accuracy apart from the dense net and Efficient Net and Squeeze-and-Excitation.

On the raf-db dataset, different splitting was carried out to check for the training and testing accuracies of the convolutional neural network of epochs 100.

Table 2: Comparison of training and testing accuracies at different splits

Splitting	Epochs	Training Accuracy	Testing Accuracy	Early Stopping
70 - 30	100	91	83	56
80 - 20	100	88	81	52
75 - 25	100	93	85	99

Table 2 shows that the best accuracy when the training set is 75% and the testing set is 25%.

Precision: It is useful for showing how many examples predicted positive for actual positive labels.

$$Precision = \frac{TP}{TP + FP}$$

Recall: It shows the proportion of true positives detected out of all the actual positive instances.

$$Recall = \frac{TP}{TP + FN}$$

Where,

TP (True Positive): The model predicts Happy when the actual label is Happy class.

TN (True Negative): The model predicts sadness when the actual label is a neutral class or any other emotion.

FP (False Positive): The model predicts Anger when the actual label is Surprise class or any other emotion.

FN (False Negative): The model predicts neutrality when the actual label is Happy class.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 6, June 2025

F1 Score: It is an evaluation metric where the combination of precision and recall, provides better model performance.

$$F1\ Score = 2\ X\ \frac{Precision\ x\ Recall}{Precision + Recall}$$

Table 3: Classification report for Convolutional Neural Network showing precision, recall, F1-score, and support for each emotion class.

Emotion	Precision	Recall	F1-Score	Support
Angry	0.89	0.90	0.90	859
Disgust	0.84	0.77	0.80	910
Fear	0.96	0.95	0.96	840
Happy	0.83	0.87	0.85	887
Neutral	0.74	0.75	0.74	894
Sad	0.78	0.78	0.78	865
Surprise	0.88	0.91	0.90	870

Table 3: shows the performance metrics of the proposed system stress detection system by various emotion classes: Angry, Disgust, Fear, Happy, Neutral, Sad, and Surprise. The model attained the highest evaluation metrics for fear, surprise, and anger. The overall Accuracy of the Convolutional Neural Network was 85%.

ROC Curve: ROC indicates the Receiver Operating Characteristic is a picturesque rendering that shows the arrangement of true positive rate across the different thresholds.

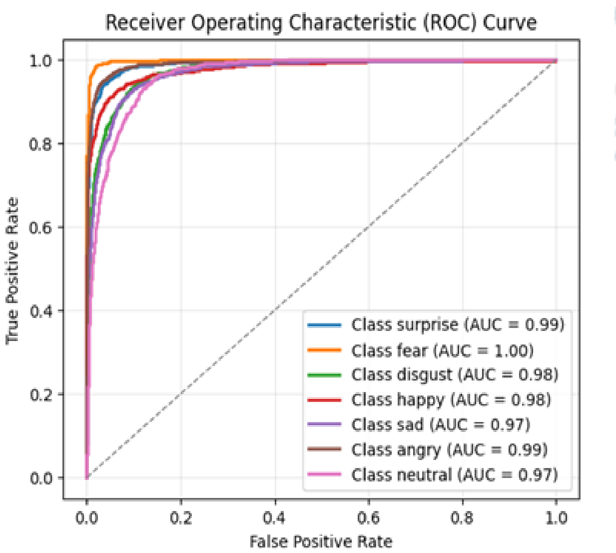

Figure 8: Multiclass ROC Curve for each emotion class in the Stress Detection System with AUC Scores

Figure 8 shows the ROC curves for seven facial emotion classes: surprise, fear, disgust, happy, sad, anger, and neutral. Each curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) for different decision thresholds.

- The AUC score for each class quantifies how well the model separates that class from the others:
  1. Fear achieves a perfect AUC of 1.00, indicating flawless distinction from other emotions.
  2. Surprise, Angry, Happy, and Disgust also show near-perfect AUCs (≥ 0.98).
  3. Sad and Neutral yield slightly lower AUCs (0.97), but still indicate strong classification performance.
- The curves lying close to the **top-left corner** demonstrate that the model has high sensitivity and low false positive rates.

VII. CONCLUSION

In this study, a deep learning-based method was introduced for detecting stress in IT professionals using facial emotion recognition from image data. Deep learning models: Convolutional Neural Network (CNN), DenseNet, and EfficientNet with a Squeeze Extractor with Real-world Affective Face Database (RAF-DB) were used for training and testing.

Intensive preprocessing of data in the form of resizing, normalization, data augmentation, and balancing was used to improve model performance. Among all the models, CNN performed best among all the others with a maximum testing accuracy of 85.24%, followed by EfficientNet+SE with 77.54%, and DenseNet with 66.78%. These results were also substantiated by excellent F1-scores and ROC AUC scores, especially for fear, surprise, and anger.

The results validate the early-ever capacity of facial expressions to serve as non-invasive yet scalable prediction indices for stress during the initial years. The proposed system could potentially become a tool for monitoring an individual's welfare in extremely heavily stressful parts of the world, such as the IT sector, where people typically fall prey to work. Future research could include real-time deployment, integration of video streams, and further use of attention mechanisms or hybrid models to improve accuracy and stability.

REFERENCES

[1] Suresh Kumar, Surekha, Lakshmi Priya, Bhavya, Beula Mungara, "Detection Of Stress in IT Employees Using Machine Learning Techniques", International Conference on Applied Artificial Intelligence and Computing; 2023. https://ieeexplore.ieee.org/document/9793195

[2] Disha Sharma, Sumit Chaudhary, "Stress Prediction of Professional Students using Machine Learning", International Journal of Engineering and Advanced Technology;2020.

[3] Gabriele Rescio, Andrea Manni, Andrea Caroppo, Alessandro Leone, Marianna Ciccarelli, Alessandra Papetti, "A Deep Learning-Based Platform for Workers' Stress Detection Using Minimally Intrusive Multisensory Devices", Multidisciplinary Digital Publishing Institute;2024. https://www.mdpi.com/1424-8220/24/3/947

[4] E. Padma, Talapaneni Praveen, and Shaik Karimulla,

"STRESS DETECTION USING MACHINE LEARNING AND IMAGE PROCESSING", Journal Of Engineering Sources, 2022. Pdf format

[5] Abdul Gaffar Khani, DR. D. Krishna Madhuri, "STRESS DETECTION IN IT PROFESSIONAL BY MACHINE LEARNING", NeuroQuantology,2022. https://www.neuroquantology.com/open-access/STRESS+DETECTION+IN+IT+PROFESSIONAL+BY++MACHINE+LEARNING_3502/

[6] Dr. K.L.S. Soujanya, B. Mohan, K. Sai Kiran Reddy, Mohammed Waqaas Adeem, "STRESS DETECTION IN IT PROFESSIONALS BY USING IMAGE PROCESSING AND MACHINE LEARNING", Journal Of Critical Reviews; 2023. Pdf format

[7] Dr. R. Sundar, Praneeth. K. B, Sai Teja. C, Sai Kumar. S, Vamsi. D, " Stress Detection in IT Professionals by Image Processing and Machine Learning", International Journal of Advanced Research in Science, Communication and Technology (IJARSCT); 2023. Pdf format

[8] Lakshmi Shree M S, Ranjitha M, Rebecca D, Shirisha H N, Shradha Sania J E, "HUMAN STRESS DETECTION BASED ON SLEEPING HABITS USING MACHINE LEARNING ALGORITHMS", International Journal of Creative Research Thoughts;2024 pdf format

[9] Swayprabha Alias Megha Mane, Arundhati Shinde, "StressNet: Hybrid model of LSTM and CNN for stress detection from electroencephalogram signal (EEG)", Results in Control and Optimization; 2023. https://www.sciencedirect.com/science/article/pii/S2666720723000334?via%3Dihub

[10] Yara Badr, Usman Tariq, Fares Al-Shargie, Fabio Babiloni, Fadwa Al Mughairbi, Hasan Al-Nashash, "A review on evaluating mental stress by deep learning using EEG signals", Neural Computing and Applications (Springer); 2024, https://link.springer.com/article/10.1007/s00521-024-09809-5

[11] Z. Zainudin, S. Hasan, S.M. Shamsuddin, and S. Argawal, "Stress Detection using Machine Learning and Deep Learning", Asian Conference on Intelligent Computing and Data Sciences; 2021.

[12] N. Balakrishna, M. B. Mukesh Krishnan, B. Veerabhadrudu, A. Rishika, B. Sruthika, B. Sanjay Kumar, **"**Facial Image Analysis for Stress Detection in IT Professionals using Machine Learning Techniques**"**, Institute of Electrical and Electronics Engineers, 2024. https://ieeexplore.ieee.org/document/10575715

[13] Induja S, Pavan Kumar G R, Rita Mary, Rama Dhananjaya V, "Stress Detection in IT Professionals Using Machine Learning", International Journal For Research in Applied Science and Engineering Technology, 2024. https://www.ijraset.com/research-paper/stress-detection-in-it-profesionals-using-machine-learning

[14] Tushar Patil, Videet Jadhav, Minal Jungare, Rupali Adhau, Abhishek Rajegaonkar, Prajwal Thombare, "STRESS DETECTION IN IT PROFESSIONAL BY IMAGE PROCESSING AND MACHINE LEARNING", International Journal of Computer Techniques, 2025. https://ijctjournal.org/indexing/stress-detection-in-it-professionals-ijct-journal/

[15] R.D. Mane, Sahil Sutar, Tanmay Patil, Shraddha Salokhe, Reenal Shah, "Stress Detection in it Professionals by Face & Speech", International Journal of Scientific Research in Engineering and Management, 2024. https://ijsrem.com/download/stress-detection-in-it-professionals-by-face-speech/

[16] Ramya Srikanteswara, P. Rithicka, Y. V. Shilpha Kala, Shreya Rangaraj, Varsha Devaiah, "Machine Learning-Based Stress Detection in IT Employees: A Data-Driven Approach for Workplace Well-being", Institute of Electrical and Electronics Engineers, 2024. https://ieeexplore.ieee.org/document/10594118

[17] K.Madhuri Reddy, K.Rajani, M.Ruchitha Reddy, M.Fathima, "Stress Detection in IT Professionals by Image Processing and Machine Learning**"**, 2022. https://ijirt.org/Article?manuscript=155909

[18] M. S. D. Perera, S. M. D. A. R Jayathilake, J. D. Ranasinghe, S. V. Bartholomeusz, H. M. Samadhi Chathuranga, Samitha Vidhanaarachchi, Thilanga Jayarathne, "Stress Monitoring and Relieving Application for IT Professionals", International Research Journal of Innovations in Engineering and Technology, 2023. https://irjiet.com/Volume-7/Issue-10-October-2023/Stress-Monitoring-and-Relieving-Application-for-IT-Professionals/1912

[19] Laura Fontes, Pedro Machado, Doratha Vinkemeier, Salisu Yahaya, Jordan J. Bird, Isibor Kennedy Ihianle, "Enhancing Stress Detection: A Comprehensive Approach through rPPG Analysis and Deep Learning Techniques", Multidisciplinary Digital Publishing Institute, 2024. https://www.mdpi.com/1424-8220/24/4/1096

[20] ANUJA PINGE, Vinaya Rajendra Gad, Dheryta Jaisinghani, Surjya Ghosh, Sougata Sen, "Detection and monitoring of stress using wearables: a systematic review", frontiers, 2024. https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2024.1478851/full