IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, June 2025
Impact Factor: 7.67

Stress Detection in IT Professionals

Adarsh Tiwari, Prajwal Kute, Palak Yede, Shruti Ratnaparkhi, Prof. Shweta Junghare
Professor, Department of Computer Engineering
Students, Department of Computer Engineering
Government College of Engineering Yavatmal, Maharashtra, India

Abstract: *The objective of this project is to identify stress levels in IT professionals by employing advanced machine learning and image processing methodologies. This system improves upon previous models by incorporating real-time detection and regular assessments, which were not features of older systems. In addition to identifying both physical and mental stress, it also facilitates periodic feedback through surveys to recommend suitable coping mechanisms. The aim is to foster a healthier and more dynamic work environment, thereby maximizing employee productivity during working hours.*

Keywords: Machine Learning, Image Processing, Stress Detection, IT Professionals, Real-Time Monitoring

I. INTRODUCTION

Stress monitoring systems are crucial for identifying stress levels that affect individuals' personal and professional lives. According to the World Health Organization (WHO), mental stress impacts one in every four people. Prolonged stress may cause emotional, social, and financial difficulties, leading to reduced work clarity, poor interpersonal relationships, depression, and in severe cases, suicidal tendencies. As a result, there is a growing need for accessible stress evaluation and support tools. While total elimination of stress is not feasible, early detection and preventive strategies can significantly reduce its adverse effects. Traditionally, stress detection has relied on self-reported questionnaires, which are subjective and not always reliable.

A. Problem Statement

With the rapid expansion of IT industries introducing innovative technologies and services, the mental health of employees has become a growing concern. Despite numerous mental wellness programs offered by organizations, stress levels among employees continue to rise. This study focuses on understanding stress indicators among employees by utilizing image processing and machine learning techniques. Through the classification of stress patterns, the project aims to pinpoint key factors contributing to elevated stress. Algorithms such as K-Nearest Neighbour (KNN) are employed to effectively categorize stress levels.

B. Overview

Machine learning enables systems to automatically improve through experience without explicit programming. It allows software to access and learn from data, creating predictive or decision-making models based on training data. Image mining, as part of image processing, helps uncover patterns and extract hidden features in visual data, assisting in identifying stress through facial expressions and subtle cues.

II. LITERATURE SURVEY

This section outlines various research studies that explore stress detection using physiological signals, facial recognition from videos, heart rate variability analysis, and digital signal processing. Each study contributes unique methodologies to understanding stress and anxiety through technology-driven systems, emphasizing non-intrusive monitoring and improved classification of emotional states.

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, June 2025
Impact Factor: 7.67

III. SYSTEM ANALYSIS

A. Existing systems for stress detection often rely on physiological signals such as skin response, blood volume, and temperature. These systems, though effective, tend to be intrusive and less comfortable for practical use. Stress levels are assessed by comparing sensor data with predefined threshold values.

B. The proposed system leverages machine learning for classifying stress levels and uses image processing to analyse facial features from captured images. This non-intrusive approach enhances detection accuracy and comfort for users.

IV. INPUT AND OUTPUT DESIGN

A. Input Design: Focuses on creating secure, user-friendly interfaces that simplify the data entry process. Validation procedures ensure accuracy and minimize errors.

B. Objectives: To transform user-provided inputs into system-readable formats efficiently, enabling accurate processing.

C. Output Design: Aims to present results clearly and effectively, supporting decision-making and system interaction. Outputs include real-time analysis, reports, and alerts.

V. IMPLEMENTATION

A. Feasibility Study: The feasibility of the proposed system is evaluated based on technical, economic, and social criteria. This ensures that implementation does not pose a burden and aligns with project objectives.

B. System Architecture: Describes how various components interact, including data acquisition, image processing, machine learning classification, and result visualization.

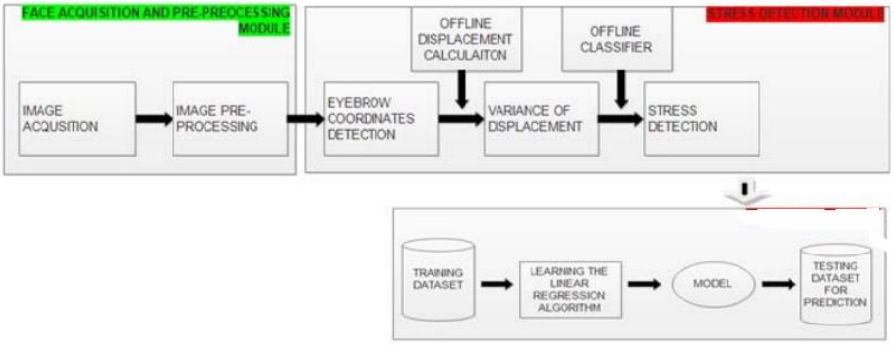


Figure 1. System Architecture

C. Data Flow Diagram: Illustrates how data moves through the system, highlighting key processes and their interactions.

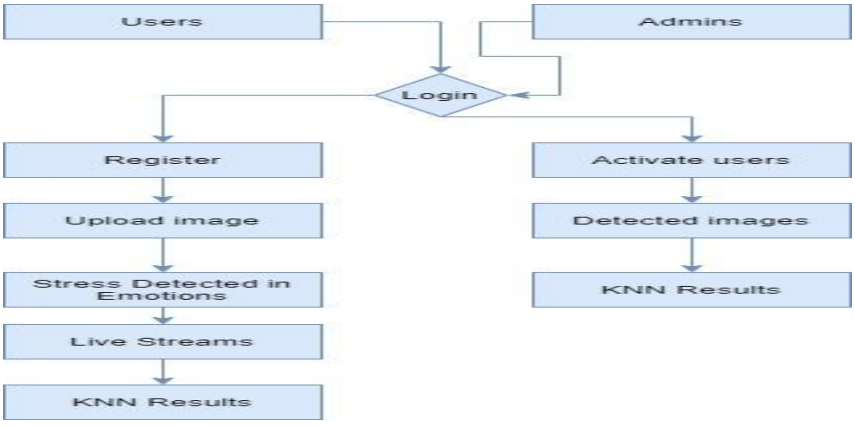


Figure 2. Data Flow Diagram

DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, June 2025
Impact Factor: 7.67

D. Use Case Diagram: Depicts system functionality and how different user roles interact with features.

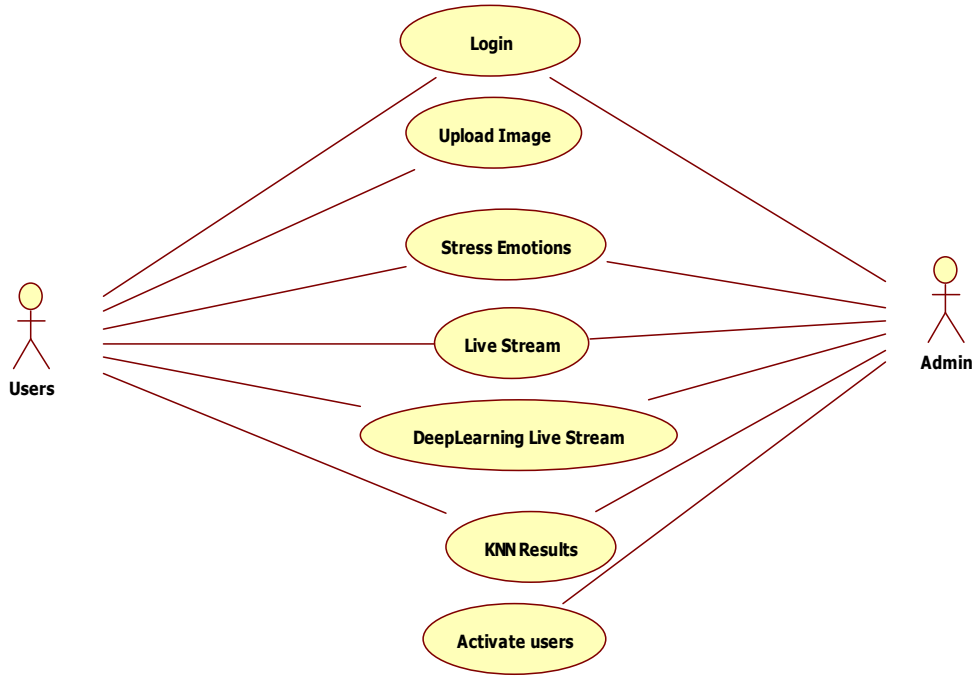


Figure 3. Use Case Diagram

E. Activity Diagram: Shows the sequence of operations and workflows within the system, aiding in understanding the overall process.

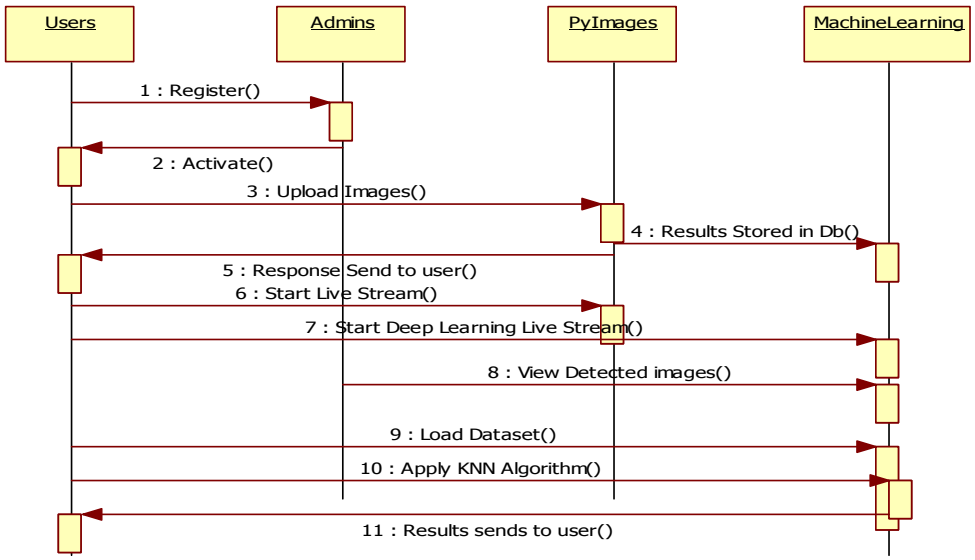


Figure 4. Activity Diagram

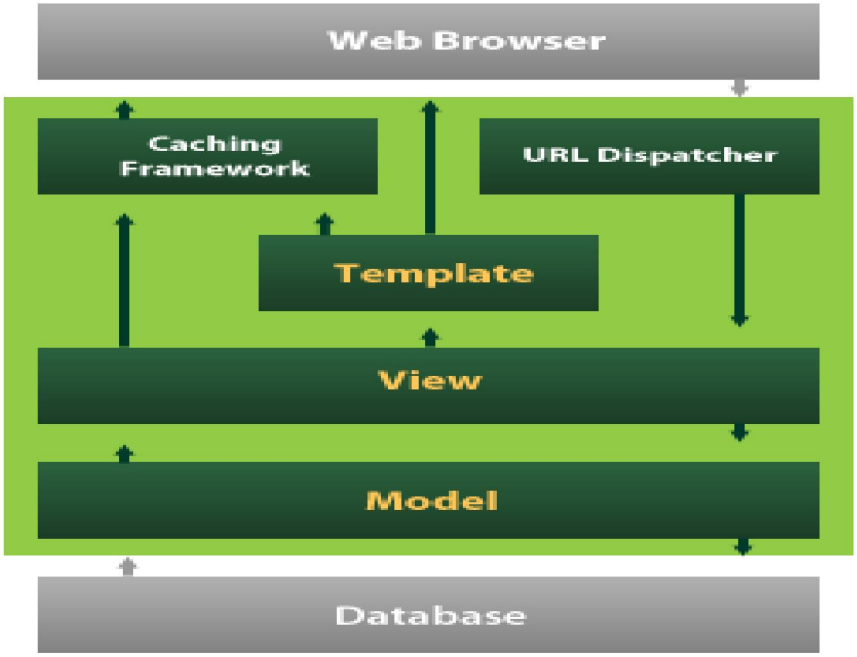
DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, June 2025
Impact Factor: 7.67

VI. SOFTWARE ENVIRONMENT

The system is developed using Python, a high-level, versatile programming language known for its simplicity and readability. The Django framework is used for web development, facilitating a structured and scalable backend. Python supports object-oriented and functional programming paradigms and includes extensive libraries suitable for machine learning and image processing applications.

A. *Django*



VII. SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies , and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing focuses on verifying the functionality of individual components or modules of a software system. The goal is to ensure that each unit performs as expected and produces correct outputs for given inputs. This type of testing is typically done immediately after the development of a single unit, but before integration with other parts. As a form of structural testing, it requires detailed knowledge of the internal code and is often intrusive in nature. Unit tests check the behaviour of specific application functions, business logic, or configurations. They validate all logical paths and decision points to confirm that each segment operates according to the predefined specifications.

Integration testing

Integration testing evaluates how different software components work together when combined. Although individual modules may pass unit testing, integration testing ensures they interact correctly as a whole. It focuses on identifying

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, June 2025
Impact Factor: 7.67

issues that may arise from combining components. This form of testing is driven by events or user interactions and typically assesses the correctness of data flow across interfaces, screens, or forms.

Functional testing

Functional testing is performed to verify that the system's features behave according to the specified business and technical requirements. It systematically checks that all intended functionalities are accessible and operate as outlined in the system documentation or user manuals. This testing ensures that the application delivers the expected results in real-world usage scenarios.

System Test

System testing is conducted on the complete, integrated system to confirm that it meets the defined requirements. It tests the software in its entirety, focusing on overall system behaviour and reliability. A common approach within this category is configuration-based system integration testing, which emphasizes testing workflows, process sequences, and integration touchpoints as defined in the process documentation.

White Box Testing

White box testing involves a thorough inspection of the internal logic and structure of the application. Testers have full visibility into the code and often utilize this insight to test hidden or complex code paths that may not be accessible through standard interfaces. This technique is suitable for verifying internal operations and ensuring that code behaves correctly under all conditions.

Black Box Testing

Black box testing treats the software as a closed system. Testers are unaware of the internal workings or structure of the code. Instead, tests are based solely on input and output without concern for how the processing is done internally. These tests are usually derived from requirement specifications and aim to validate the software's external functionality by simulating user behaviour.

VIII. CONCLUSION

The developed stress detection system efficiently monitors employee stress through periodic image capture and analysis. Utilizing machine learning algorithms, it provides accurate classification and promotes mental well-being in IT professionals.

REFERENCES

[1]. Giannakakis, D. Manousos, F. Chiarugi, "Stress and anxiety detection using facial cues from videos," Biomedical Signal Processing and Control, vol. 31, pp. 89-101, 2017.
[2]. T. Jick and R. Payne, "Stress at work," Journal of Management Education, vol. 5, no. 3, pp. 50-56, 1980.
[3]. Nisha Raichur, Nidhi Lonakadi, Priyanka Mural, "Detection of Stress Using Image Processing and Machine Learning Techniques," vol.9, no. 3S, 2017.
[4]. Bakker, J., et al., 'Stress@work: From measuring stress to its understanding', ACM Intl. health informatics symposium, 2012.
[5]. Deng, Y., et al., 'Sensor feature selection and combination for stress identification', Int. J. of Advanced Robotic Systems, 2013.
[6]. Villarejo, M.V., et al., 'A stress sensor based on GSR controlled by Zigbee', Sensors, vol. 12(5), 2012.
[7]. Tanev, G., et al., 'Classification of acute stress using HRV', IEEE EMBC, 2014.
[8]. Gjoreski, M., Gjoreski, H., Lustrek, M., Gams, M.. Continuous stress detection using a wrist device: in laboratory and real life. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct.
[9]. Koldijk, S., et al., 'The SWELL knowledge work dataset for stress research', ACM ICMI, 2014.

DOI: 10.48175/568