 

HAROHALLI, KANAKAPURA ROAD – 562112

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(DATA SCIENCE)

**FODS PROJECT REPORT**

ON

#### 

#### "STRESS DETECTION BASED ON COMPOUND FACIAL EMOTIONS"

2024-2025

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)

**Submitted by**

Dhanushree N-ENG23DS0009

Sanitha M – ENG23DS0030

**Under The Supervision of:**

#### Prof. Sindhu A.

#### Assistant Professor

#### Department of CSE (Data Science), DSU

DAYANANDA SAGAR UNIVERSITY

#### DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**(DATA SCIENCE)**

#### SCHOOL OF ENGINEERING

**HAROHALLI, KANAKAPURA ROAD – 562112**



CERTIFICATE

#### It is certified that the mini project work entitled “Your Title” has been carried out at *Dayananda Sagar University*, Bangalore, by Sanitha M*-*ENG23DS0030,Dhanushree N-ENG23DS0009

#### Bonafide students of fourth Semester, B.Tech in partial fulfilment for the award of degree in *Bachelor of Technology in Computer Science & Engineering (Data Science)* during academic year *2024-25*. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in departmental library.

#### The project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

**Signature of the Guide Signature of the Chairperson**

**ACKNOWLEDGEMENT**

A project's successful completion offers a sense of satisfaction, but it is never finished without expressing gratitude to everyone who contributed to its accomplishment. We would like to convey our sincere gratitude to our esteemed university, Dayananda Sagar University, for offering the first-rate facilities.

We are especially thankful to our Chairperson, Dr. Shaila S G, for providing necessary departmental facilities, moral support and encouragement. The largest measure of our acknowledgment is reserved for Prof. Sindhu A. whose guidance and support made it possible to complete the project work in a timely manner.

We would want to thank everyone who has assisted us in successfully completing this project work, both directly and indirectly. The staff has provided us with a great deal of direction and cooperation.

Sanitha M-ENG23DS0030

Dhanushree N-ENG23DS0009

**DECLARATION**

We hereby declare that the project entitled **"Stress detection based on compound facial emotions"** submitted to Dayananda Sagar University, Bengaluru, is a bona fide record of the work carried out by me under the guidance of Prof. Sindhu A., Assistant Professor in the Dayananda Sagar University School of Engineering's Department of Computer Science and Engineering (Data Science). This work is submitted toward the partial fulfillment of the requirements for the award of a Bachelor of Technology in Computer Science and Engineering (Data Science).

Sanitha M-ENG23DS0030

Dhanushree N-ENG23DS0009

**ABSTRACT**

Stress is a growing concern in modern life, with significant impacts on both physical and mental health. Traditional stress detection methods often rely on self-reported data or physiological sensors, which can be intrusive and impractical in many settings. This project presents a non-invasive, image-based approach to stress detection using facial expressions, leveraging computer vision and machine learning techniques.

We use the publicly available FER2013 dataset, which consists of grayscale images categorized into seven basic emotions: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. Each emotion is mapped to a corresponding stress level — high, medium, or low — based on psychological research linking emotions to stress responses. Histogram of Oriented Gradients (HOG) is employed to extract relevant features from facial images, effectively capturing edge and gradient information that characterizes expressions.

Instead of using complex classifiers, we adopt a simple yet effective **manual centroid-based classifier**. For each stress class, the centroid (mean vector) of its training feature set is computed. A test image is then classified by finding the nearest centroid using Euclidean distance.

The system is implemented entirely in MATLAB, using built-in functions for image processing and classification. Results demonstrate that the proposed method is capable of detecting stress levels with reasonable accuracy, offering a lightweight and interpretable solution. While not as powerful as deep learning models, this approach is suitable for applications with limited resources and serves as a valuable educational tool.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CHAPTER** | **PG. NO.** |
| 1. INTRODUCTION | 7-7 |
| 1. OBJECTIVES AND SCOPE OF WORK | 8-8 |
| 1. DESCRIPTION OF WORK | 9-9 |
| 1. SOURCE CODE | 10-11 |
| 1. RESULT | 12-12 |
| 1. CONCLUSION | 13-13 |
| 1. REFERENCES | 14-14 |

**INTRODUCTION**

Stress is a psychological and physiological response to demanding or threatening situations. With the increasing pace of modern life, stress-related issues have become common, especially among students, professionals, and individuals in high-pressure environments. Early detection of stress is essential to prevent long-term health problems such as anxiety, depression, and cardiovascular diseases.

Traditional methods for detecting stress include self-assessment questionnaires, wearable biosensors, and medical evaluations. While effective, these methods can be intrusive, expensive, and impractical for regular use. In contrast, analysing facial expressions offers a non-contact and natural alternative for stress detection.

Facial expressions are widely recognized as powerful indicators of a person's emotional and mental state. Emotions such as anger, sadness, fear, and happiness can provide meaningful cues about stress levels. By analysing facial features captured in images, it becomes possible to infer underlying stress using computational techniques.

This project proposes a vision-based approach for detecting stress levels from facial expressions using classical machine learning techniques. The system utilizes the FER2013 dataset, which contains labelled facial images of seven basic emotions. These emotions are mapped to three stress levels — high, medium, and low — based on psychological theory.

To extract features from facial images, Histogram of Oriented Gradients (HOG) is used, which captures structural and edge information critical for emotion recognition. For classification, a simple manual centroid-based method is implemented, where the system calculates the average feature vector (centroid) of each stress class and assigns test samples to the nearest centroid.

The entire system is implemented using MATLAB, emphasizing interpretability, efficiency, and educational value. This project demonstrates that even without complex neural networks, meaningful stress detection can be achieved using classical approaches with reasonable accuracy.

**OBJECTIVE** **AND SCOPE OF WORK**

### ****Objectives****

 **To analyse the relationship between facial emotions and psychological stress** using established emotion-stress models.

 **To design a preprocessing pipeline** for facial image normalization, including grayscale conversion, resizing, and noise reduction.

 **To extract consistent and discriminative features** from facial images using Histogram of Oriented Gradients (HOG), ensuring that the model captures emotion-relevant information.

 **To reduce computational complexity** by using a lightweight manual classification method instead of deep learning models.

 **To construct a stress level classification system** based on the proximity of image features to precomputed stress-class centroids.

 **To ensure model interpretability and transparency**, making it suitable for academic and research learning environments.

 **To evaluate model accuracy, precision, recall, and confusion matrix** for better understanding of class-wise performance.

 **To demonstrate how classical image processing methods** can still be effective in tasks often dominated by neural networks.

 **To propose a mapping between categorical emotions and ordinal stress levels**, offering flexibility for real-world emotional analysis applications.

### ****Scope of Work****

 **Dataset Handling:**

* Utilize the **FER2013 dataset** from Kaggle, which includes thousands of labeled facial expression images.
* Organize the dataset into training and testing directories, each containing subfolders for the seven basic emotions: *Angry, Disgust, Fear, Happy, Sad, Surprise,* and *Neutral*.

 **Image Preprocessing:**

* Convert images to **grayscale** (if not already).
* **Resize all images** to a uniform 48x48 pixel dimension to ensure consistent feature extraction.
* Apply any basic image normalization techniques if needed.

 **Feature Extraction using HOG:**

* Implement **Histogram of Oriented Gradients (HOG)** to extract edge and orientation features from each facial image.
* Use consistent cell size and parameters across the dataset for uniformity.

 **Emotion to Stress Mapping:**

* Develop a mapping from **7 emotional categories** to **3 stress levels**:
  + *High stress*: Angry, Disgust
  + *Medium stress*: Fear, Sad
  + *Low stress*: Happy, Surprise, Neutral

 **Manual Classifier Implementation:**

* Compute **centroid vectors (mean HOG features)** for each of the three stress levels using training data.
* Classify each test image by computing the **Euclidean distance** from its features to each centroid and assigning the nearest one.

 **Model Evaluation:**

* Test the classifier on unseen images.
* Calculate overall **accuracy** and optionally generate a **confusion matrix** to understand class-wise performance.
* Analyse strengths and weaknesses of the manual classification approach.

 **MATLAB Development:**

* Write modular MATLAB scripts:
  + extractFeatures.m for preprocessing and feature extraction.
  + mainStressDetection.m for training, testing, and evaluation.
* Ensure code is readable, well-commented, and logically organized.

 **Documentation and Presentation:**

* Prepare a well-structured **report** detailing the methodology, implementation, results, and future scope.
* Design a visually appealing **PowerPoint presentation** summarizing the entire project flow and outcomes.

 **Future Extension Planning:**

* Define potential enhancements such as using CNNs, integrating physiological signals, or deploying in real-time systems.

### DESCRIPTION OF WORK

The project was implemented in a structured manner, starting from dataset preparation to classification and evaluation. The complete work can be described through the following workflow:

#### ****Data Collection and Preparation****

The FER2013 dataset was obtained from Kaggle. It contains thousands of 48x48 pixel grayscale facial images categorized into seven emotional classes. These images were organized into training and testing directories with subfolders for each emotion.

#### ****Emotion-to-Stress Mapping****

To focus on stress detection, a mapping from emotional labels to stress levels was defined based on psychological theory. Emotions were grouped into:

* High stress: Angry, Disgust
* Medium stress: Sad, Fear
* Low stress: Happy, Surprise, Neutral

#### ****Feature Extraction using HOG****

Each image was resized to 48x48 pixels (if needed), converted to grayscale, and then passed through the HOG (Histogram of Oriented Gradients) algorithm. HOG extracts local shape and edge features that are effective in capturing facial expression details.

#### ****Training Phase – Manual Classifier****

For each stress level, HOG features of all training images were averaged to form a centroid. These centroids represent the mean feature profile of each class.

#### ****Testing Phase – Classification****

During testing, each test image’s HOG features were compared to the class centroids using Euclidean distance. The class with the smallest distance was assigned as the predicted stress level.

#### ****Accuracy Evaluation****

The predicted stress labels were compared against the actual (mapped) labels from the dataset. Accuracy was calculated, and performance insights were recorded.

#### ****MATLAB Implementation****

The entire process was implemented using MATLAB. Modular scripts were written for feature extraction (extractFeatures.m) and the main classification workflow (mainStressDetection.m).

#### ****Documentation****

A detailed report and PowerPoint presentation were created to explain the methodology,results professional format.

### METHODOLOGY

The methodology of this project follows a step-by-step process involving data preparation, feature extraction, stress mapping, classification, and evaluation. The entire system is implemented using MATLAB. The steps are as follows:

#### ****Dataset Acquisition****

The **FER2013 dataset** from Kaggle is used, which contains thousands of grayscale facial images (48x48 pixels) labeled with one of seven basic emotions:  
**[Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral]**

#### ****Preprocessing****

* All images are resized to a uniform size of **48×48 pixels**.
* Images are converted to **grayscale** if they are in RGB format.
* This ensures consistency for feature extraction.

#### ****Emotion to Stress Mapping****

A mapping is defined to convert emotional labels into stress levels:

* **High Stress**: Angry, Disgust
* **Medium Stress**: Fear, Sad
* **Low Stress**: Happy, Surprise, Neutral

#### ****Feature Extraction using HOG****

* **Histogram of Oriented Gradients (HOG)** is applied to each image to extract features.
* HOG captures edge directions and intensity gradients, which are effective in representing facial structures.
* Each image is represented as a **HOG feature vector**.

#### ****Manual Classification (Nearest Centroid)****

* The **centroid (mean vector)** for each stress class is computed from the training data.
* For each test image, its HOG feature vector is compared to all centroids using **Euclidean distance**.
* The stress class with the **nearest centroid** is assigned as the prediction.

#### ****Evaluation****

* Predictions are compared with the true stress labels from the test set.
* **Accuracy** is calculated to measure the performance of the classifier.
* Optionally, a **confusion matrix** can be created to analyze class-wise results.

#### ****MATLAB Implementation****

* The methodology is implemented in MATLAB using two scripts:
  + extractFeatures.m: Handles preprocessing and HOG feature extraction.
  + mainStressDetection.m: Handles centroid calculation, prediction, and evaluation.

### SOURCE CODE

### MATLAB code to Convert Emotion-Labeled Images to Stress-Labeled

clc;clear;close all;

inputFolders={'fer2013/train’,'fer2013/test'};

outputPath='stress\_dataset';

stressMap=containers.Map({'angry','disgust','fear','sad','neutral','happy','surprise'},{'high\_stress','high\_stress','medium\_stress','medium\_stress','medium\_stress','low\_stress','low\_stress'} );

emotions=keys(stressMap);

imgCount=0;

for k = 1:length(inputFolders)

rootPath=inputFolders{k};

for i = 1:length(emotions)

emotion = emotions{i};

emotionFolder=fullfile(rootPath, emotion);

if ~exist(emotionFolder,'dir')

fprintf('Skipping missing folder: %s\n',emotionFolder);

continue;

end

files = [ dir(fullfile(emotionFolder, '\*.jpg'));];

stressLabel=stressMap(emotion);

stressFolder=fullfile(outputPath,stressLabel);

if~exist(stressFolder,'dir')

mkdir(stressFolder);

end

for j=1:length(files)

src=fullfile(emotionFolder, files(j).name);

[~,~,ext]=fileparts(files(j).name);

dst=fullfile(stressFolder,sprintf('%s\_img\_%05d%s',emotion,imgCount, ext));

copyfile(src, dst);

imgCount=imgCount + 1;

end

fprintf('Copied %d "%s" images to "%s"\n', length(files), emotion, stressLabel);

end

end

disp('All images grouped by stress levels.');

### extractFeatures.m

This script handles image preprocessing and HOG feature extraction

function [features, labels] = extractFeatures(baseFolder, emotions, stressMap)

totalImages = 0;

for i = 1:length(emotions)

folderPath = fullfile(baseFolder, emotions{i});

imgFiles = dir(fullfile(folderPath, '\*.jpg'));

totalImages = totalImages + length(imgFiles);

end

% Check sample image for HOG length

imgFilesSample = dir(fullfile(baseFolder, emotions{1}, '\*.jpg'));

if isempty(imgFilesSample)

error('No images found in %s', fullfile(baseFolder, emotions{1}));

end

sampleImgPath = fullfile(baseFolder, emotions{1}, imgFilesSample(1).name);

sampleImg = imread(sampleImgPath);

if size(sampleImg,3) == 3

sampleImg = rgb2gray(sampleImg);

end

sampleImg = imresize(sampleImg, [48 48]);

hogFeatureSample = extractHOGFeatures(sampleImg, 'CellSize', [8 8]);

hogLength = length(hogFeatureSample);

features = zeros(totalImages, hogLength);

labels = strings(totalImages,1);

idx = 0;

for i = 1:length(emotions)

emotion = emotions{i};

folderPath = fullfile(baseFolder, emotion);

imgFiles = dir(fullfile(folderPath, '\*.jpg'));

for j = 1:length(imgFiles)

imgPath = fullfile(folderPath, imgFiles(j).name);

img = imread(imgPath);

if size(img,3) == 3

img = rgb2gray(img);

end

img = imresize(img, [48 48]);

hogFeature = extractHOGFeatures(img, 'CellSize', [8 8]);

idx = idx + 1;

features(idx, :) = hogFeature;

labels(idx) = string(stressMap(emotion));

end

end

% Trim to actual number of images

features = features(1:idx, :);

labels = labels(1:idx);

end

### 

### mainStressDetection.m

This script performs classification using the nearest centroid method and evaluates the model.

clear; clc;

% Define emotions and stress level mapping

emotions = {'surprise','sad','neutral','happy','fear','disgust','angry'};

stressMap = containers.Map( ...

{'angry','disgust','fear','happy','sad','surprise','neutral'}, ...

{'high\_stress','high\_stress','medium\_stress','low\_stress','medium\_stress','low\_stress','low\_stress'});

% Extract features for training

trainFolder = fullfile('fer2013', 'train');

fprintf('Extracting training features...\n');

[featuresTrain, labelsTrain] = extractFeatures(trainFolder, emotions, stressMap);

fprintf('Training samples: %d\n', size(featuresTrain,1));

% Extract features for testing

testFolder = fullfile('fer2013', 'test');

fprintf('Extracting testing features...\n');

[featuresTest, labelsTest] = extractFeatures(testFolder, emotions, stressMap);

fprintf('Testing samples: %d\n', size(featuresTest,1));

% Unique stress classes

classes = unique(labelsTrain);

numClasses = length(classes);

% Calculate centroid of each class

centroids = zeros(numClasses, size(featuresTrain, 2));

for c = 1:numClasses

centroids(c, :) = mean(featuresTrain(labelsTrain == classes(c), :), 1);

end

fprintf('Training completed. Starting prediction...\n');

% Predict using nearest centroid

predictions = strings(size(featuresTest,1), 1);

for i = 1:size(featuresTest,1)

dists = sum((centroids - featuresTest(i, :)).^2, 2);

[~, idx] = min(dists);

predictions(i) = classes(idx);

end

% Calculate accuracy

accuracy = sum(predictions == labelsTest) / numel(labelsTest);

fprintf('Stress detection accuracy: %.2f%%\n', accuracy\*100);

**eda.m**

clear; clc;

% Define emotions and stress level mapping

emotions = {'surprise','sad','neutral','happy','fear','disgust','angry'};

stressMap = containers.Map( ...

{'angry','disgust','fear','happy','sad','surprise','neutral'}, ...

{'high\_stress','high\_stress','medium\_stress','low\_stress','medium\_stress','low\_stress','low\_stress'});

% Extract features for training

trainFolder = fullfile('fer2013', 'train');

fprintf('Extracting training features...\n');

[featuresTrain, labelsTrain] = extractFeatures(trainFolder, emotions, stressMap);

fprintf('Training samples: %d\n', size(featuresTrain,1));

% Extract features for testing

testFolder = fullfile('fer2013', 'test');

fprintf('Extracting testing features...\n');

[featuresTest, labelsTest] = extractFeatures(testFolder, emotions, stressMap);

fprintf('Testing samples: %d\n', size(featuresTest,1));

%% ------------------------- EDA Section ------------------------- %%

fprintf('Performing Exploratory Data Analysis (EDA)...\n');

% Plot class distribution in training data

figure;

histogram(categorical(labelsTrain));

title('Stress Level Distribution in Training Data');

xlabel('Stress Level'); ylabel('Number of Samples');

grid on;

% Plot class distribution in test data

figure;

histogram(categorical(labelsTest));

title('Stress Level Distribution in Testing Data');

xlabel('Stress Level'); ylabel('Number of Samples');

grid on;

% Mean and variance of features

meanFeatures = mean(featuresTrain);

varFeatures = var(featuresTrain);

% Plot first 20 mean feature values (or modify according to feature count)

figure;

bar(meanFeatures(1:20));

title('Mean of First 20 Features (Train Data)');

xlabel('Feature Index'); ylabel('Mean Value');

% Plot variance of first 20 features

figure;

bar(varFeatures(1:20));

title('Variance of First 20 Features (Train Data)');

xlabel('Feature Index'); ylabel('Variance');

% PCA visualization (2D projection for clustering insight)

[coeff,score,~,~,explained] = pca(featuresTrain);

figure;

gscatter(score(:,1), score(:,2), labelsTrain);

title('PCA Projection (First 2 Components)');

xlabel(['PC1 - ' num2str(explained(1), '%.2f') '%']);

ylabel(['PC2 - ' num2str(explained(2), '%.2f') '%']);

legend('Location','best');

%% --------------------- Classification Section --------------------- %%

% Unique stress classes

classes = unique(labelsTrain);

numClasses = length(classes);

% Calculate centroid of each class

centroids = zeros(numClasses, size(featuresTrain, 2));

for c = 1:numClasses

centroids(c, :) = mean(featuresTrain(labelsTrain == classes(c), :), 1);

end

fprintf('Training completed. Starting prediction...\n');

% Predict using nearest centroid

predictions = strings(size(featuresTest,1), 1);

for i = 1:size(featuresTest,1)

dists = sum((centroids - featuresTest(i, :)).^2, 2);

[~, idx] = min(dists);

predictions(i) = classes(idx);

end

% Calculate accuracy

accuracy = sum(predictions == labelsTest) / numel(labelsTest);

fprintf('Stress detection accuracy: %.2f%%\n', accuracy\*100);

% Confusion Matrix

figure;

confusionchart(labelsTest, predictions);

title('Confusion Matrix for Stress Detection');

### 

### 

### RESULT

### 

### 

### 

### 

### 

### 

### 

### CONCLUSION

This project successfully demonstrated a stress detection system by leveraging facial emotion recognition using the FER2013 dataset. By mapping seven basic emotions into three stress levels—high, medium, and low—we effectively transformed a multi-class emotion classification problem into a simpler stress classification task. Using Histogram of Oriented Gradients (HOG) features to capture facial structural information, combined with a straightforward nearest centroid classifier, the system was able to identify stress levels from facial images with reasonable accuracy.

The methodology emphasized simplicity and interpretability, making it suitable for real-time or resource-constrained applications. However, the accuracy could potentially be improved by exploring more advanced classifiers, incorporating temporal information, or using deeper feature extraction methods like convolutional neural networks.

Overall, this work lays a foundation for automatic, non-intrusive stress monitoring through facial expressions, which can be beneficial in fields such as mental health, workplace safety, and human-computer interaction.

### 

### 

### 

### REFERENCES

1. **Goodfellow, I., Erhan, D., Carrier, P.L., Courville, A., Mirza, M., Hamner, B.,... Bengio, Y.**  
   “Challenges in Representation Learning: A Report on Three Machine Learning Contests.” *Neural Networks*, 2013.  
   [FER2013 dataset origin]
2. **Dalal, N., & Triggs, B.**  
   “Histograms of Oriented Gradients for Human Detection.” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.  
   https://doi.org/10.1109/CVPR.2005.177
3. **Ekman, P., & Friesen, W.V.**  
   “Constants Across Cultures in the Face and Emotion.” *Journal of Personality and Social Psychology*, 1971.
4. **Jolliffe, I. T.**  
   *Principal Component Analysis.* Springer, 2002.  
   [General feature reduction reference, optional]
5. **MATLAB Documentation**  
   “extractHOGFeatures — Extract Histogram of Oriented Gradients (HOG) Features.” MathWorks.  
   https://www.mathworks.com/help/vision/ref/extracthogfeatures.html
6. **Scherer, K. R.**  
   “Psychological Models of Stress and Emotion.” *The Oxford Handbook of Stress, Health, and Coping*, 2011.