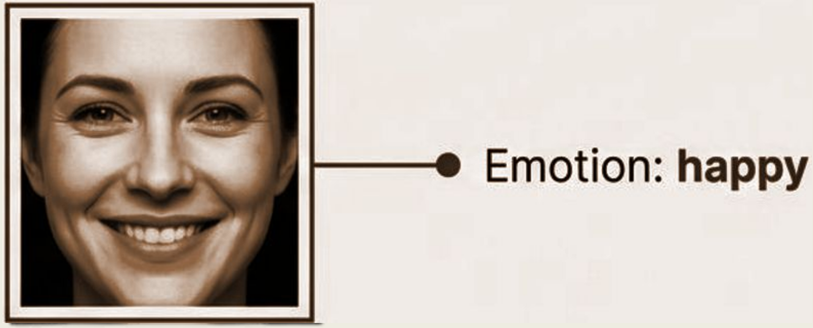
Emotion Detection from Images using ML

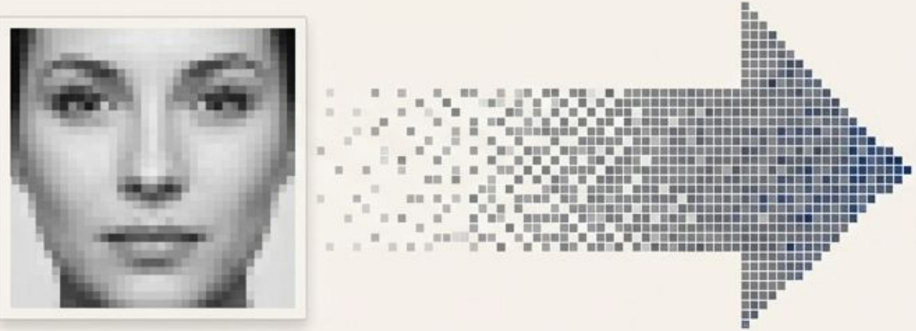


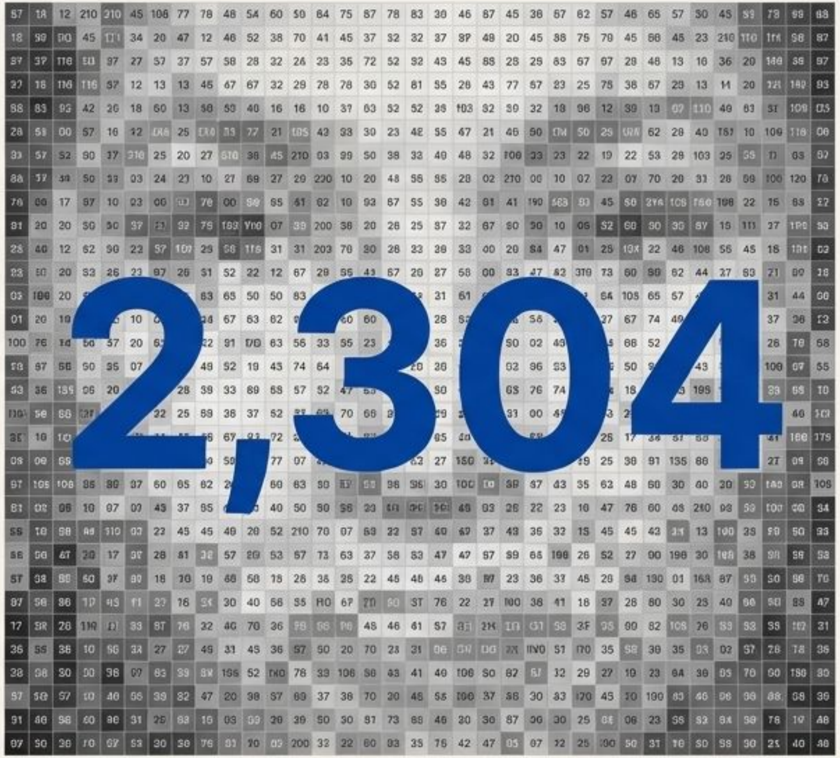
Presented By:
Advanced Signal and Image Processing Lab (ASIP Lab)
Dept. of Data Science and Engineering
IISER Bhopal

Facial Emotion Recognition

The human face is one of our most powerful communication tools, broadcasting unfiltered emotional responses. Modern AI, specifically Facial Emotion Recognition (FER), aims to interpret this complex language.

The Challenge: A Single 48x48 Image Contains 2,304 Raw Features

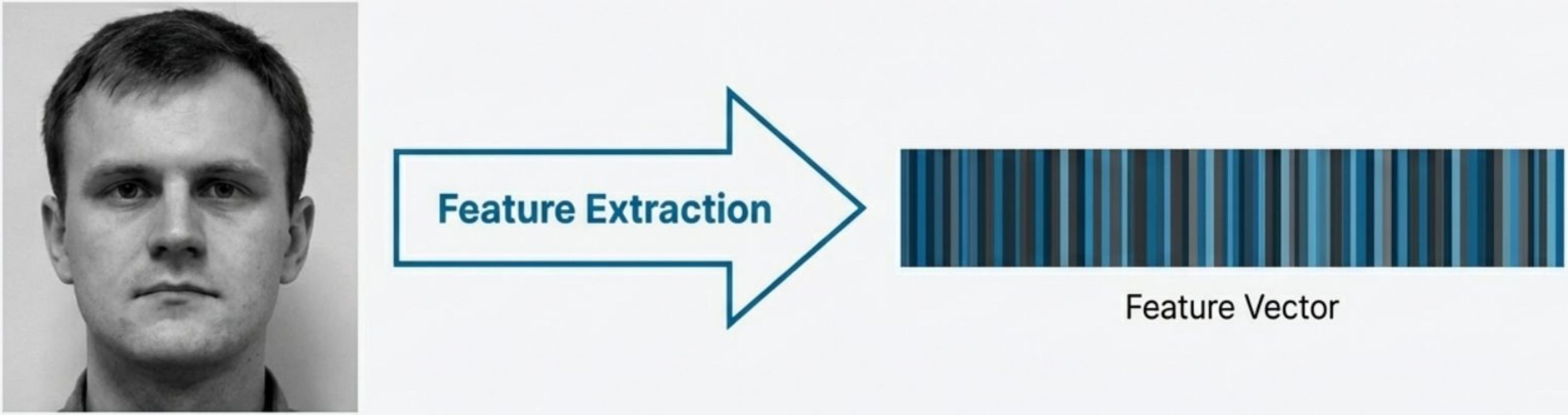




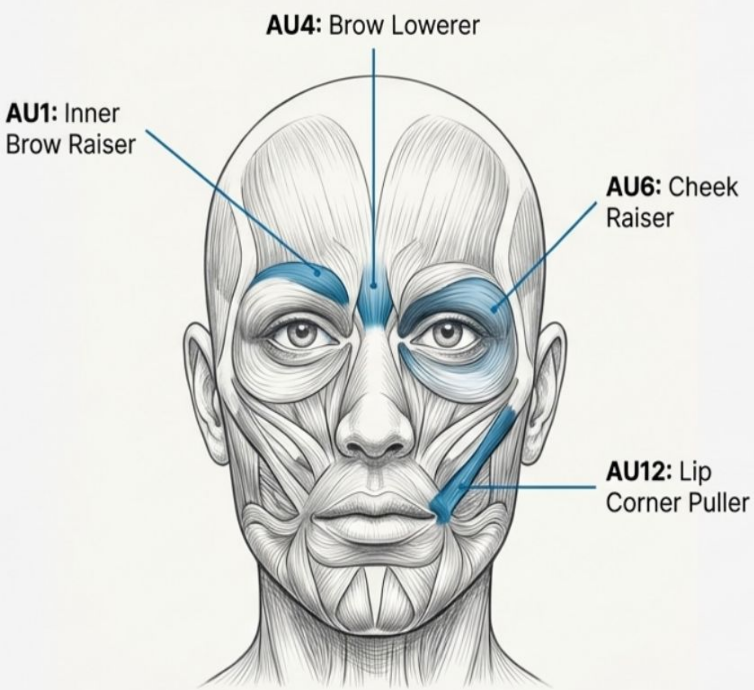
When a model looks at an image, it doesn't see a face; it sees a grid of numbers. A small 48×48 pixel grayscale image translates to **2,304 individual data points**. This high dimensionality overwhelms most models, making it difficult to distinguish meaningful patterns from random noise.

Too many features, So Need Feature Extraction (HoG/LBP)

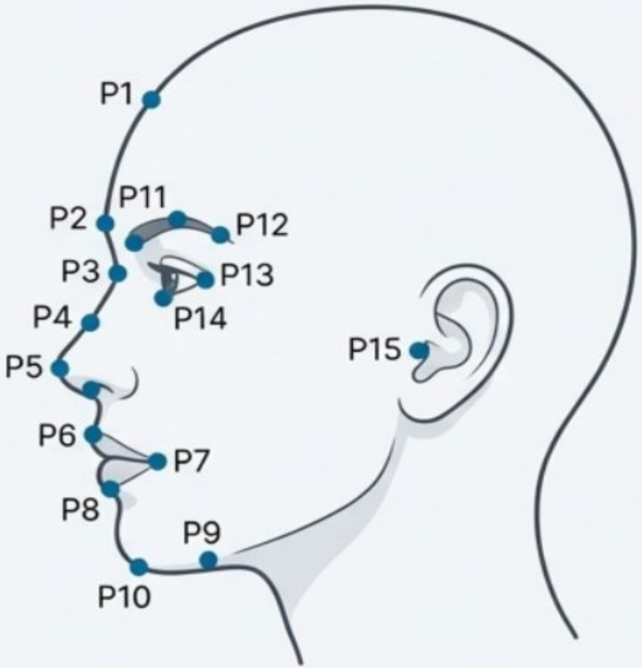
Feature Detection from Image



Identifying Fiducial Points



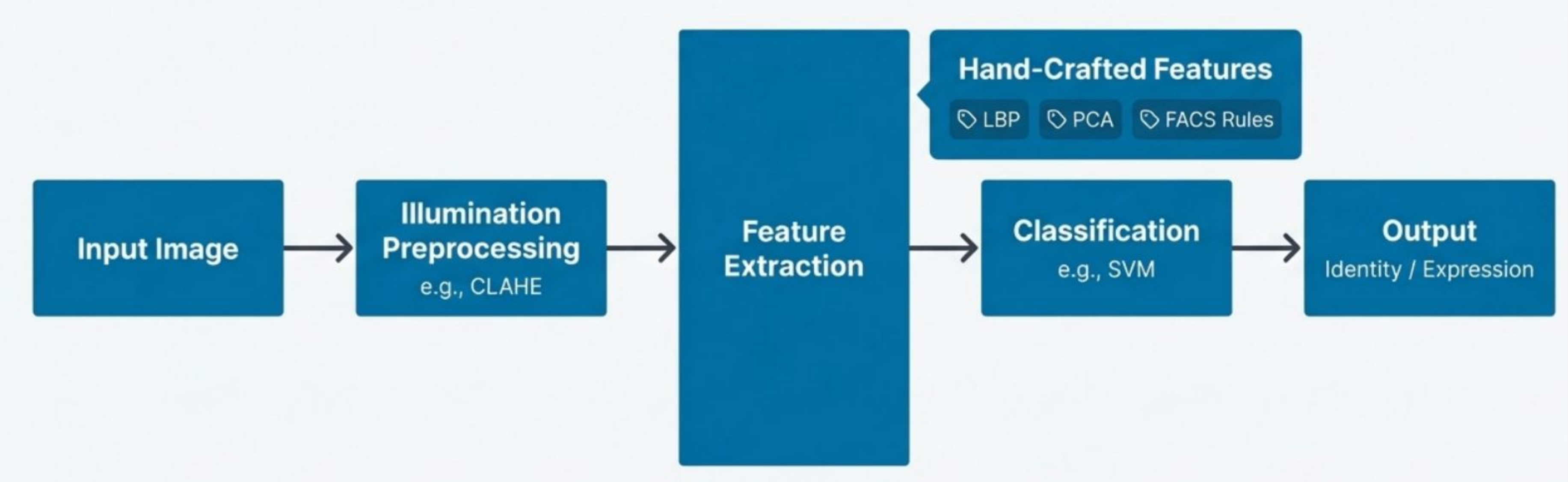
P1 (top of forehead)
P2 (eyebrow arcade)
P3 (root of nose)
P4 (tip of nose)
P5 (nostril)
P6 (upper lip)
P7 (mouth corner)
P8 (lower lip)
P9 (lower jaw)
P10 (tip of chin)



Classic (Traditional) Pipeline

Basic schematic which involves

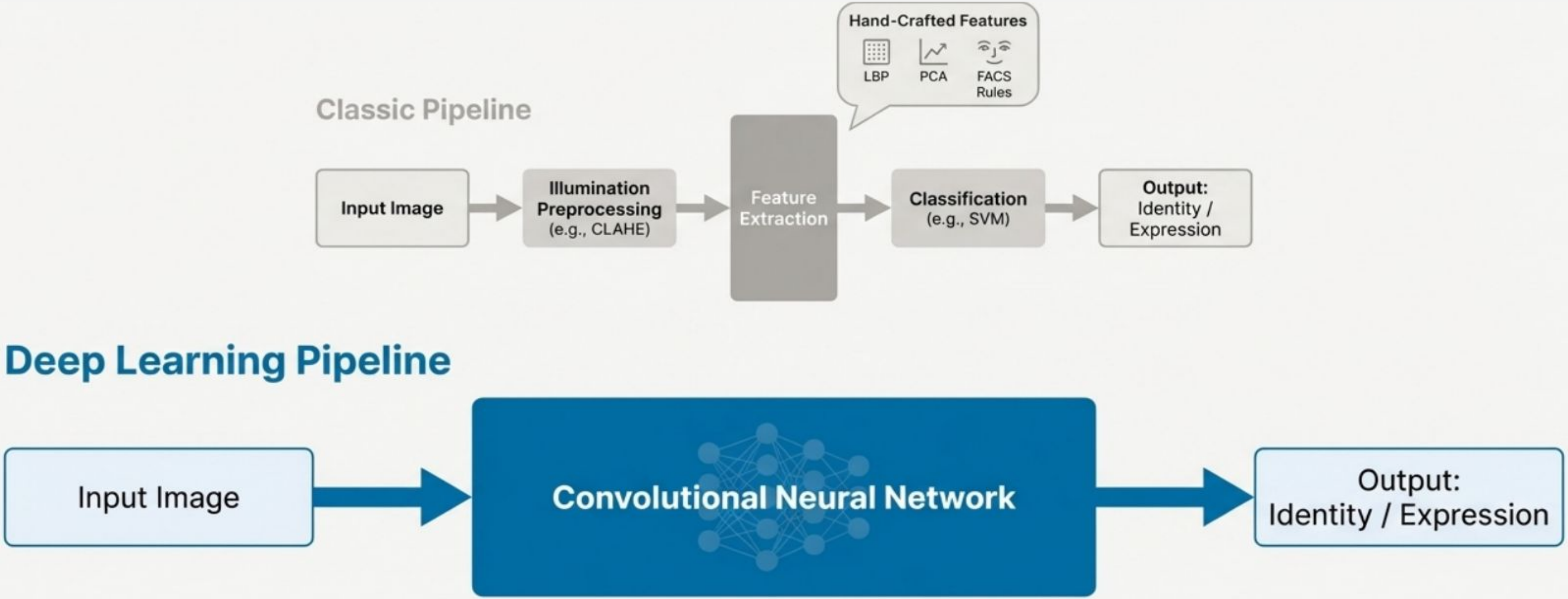
1. Pre-processing (Illumination correction, Scaling, Denoising, etc.)
2. Extraction of features (LBP, HoG, etc)
3. Feeding the feature into ML Classification Models



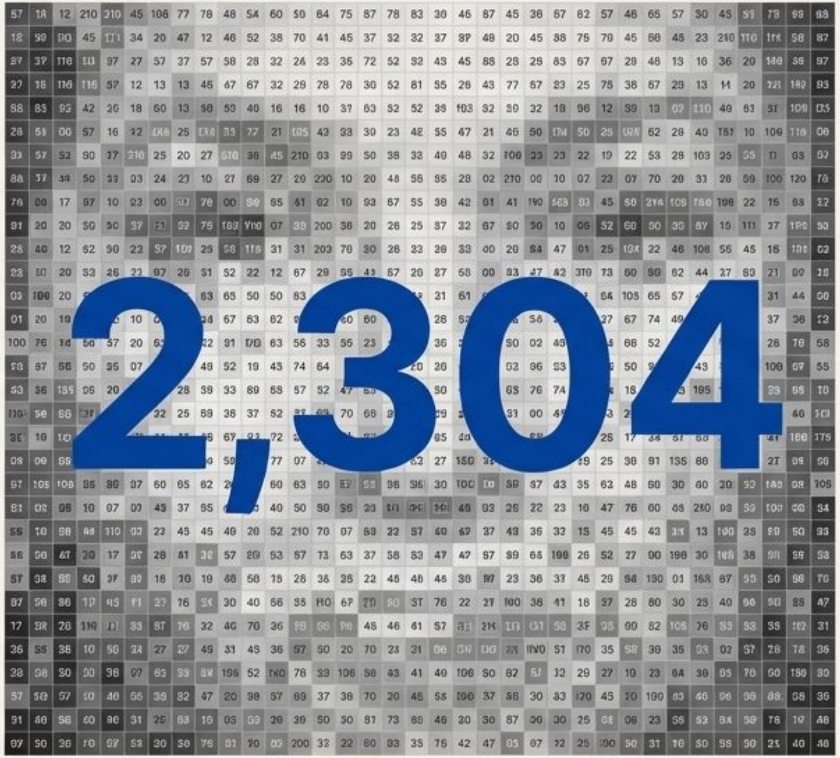
Basic schematic for emotion classification

Classic (Traditional) Pipeline To Deep Neural Network

A DNN-like CNN learns features automatically.



The Challenge: A Single 48x48 Image Contains 2,304 Raw Features



When a model looks at an image, it doesn't see a face; it sees a grid of numbers. A small 48×48 pixel grayscale image translates to **2,304 individual data points**. This high dimensionality overwhelms most models, making it difficult to distinguish meaningful patterns from random noise.

Too many features, So Need Feature Extraction (HoG/LBP)

Emotion Recognition Implementation

We will implement a Facial Emotion Recognition system, starting with the foundational theory, moving through the practical build using Python, evaluating its performance

01 THEORY & TOOLS

We Will use different models using Python.

02 DATA UNDERSTADING & PRE PROCESSING

Understanding and prepare the dataset

03 BUILD & TRAIN

Assembling and training the model architecture.

04 EVALUATION & REALITY CHECK

Measuring performance.

Emotion Recognition Implementation

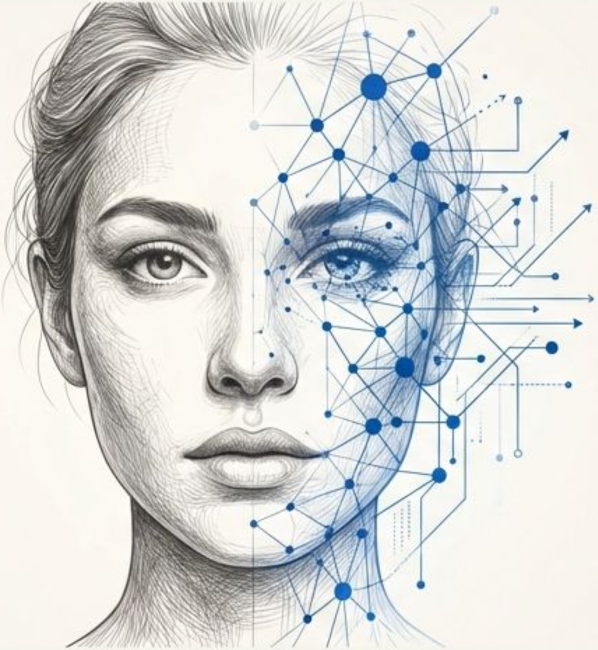
https://sorts.pro/emotion

Let's Implement

Scan QR Code for Code or Follow the Link:

Emotion

Decoding Emotion



Scan QR Code for Code or Follow the Link:

Emotion

A Machine Learning Journey from Pixels to Predictions

An analysis of feature extraction and model performance for facial emotion recognition.

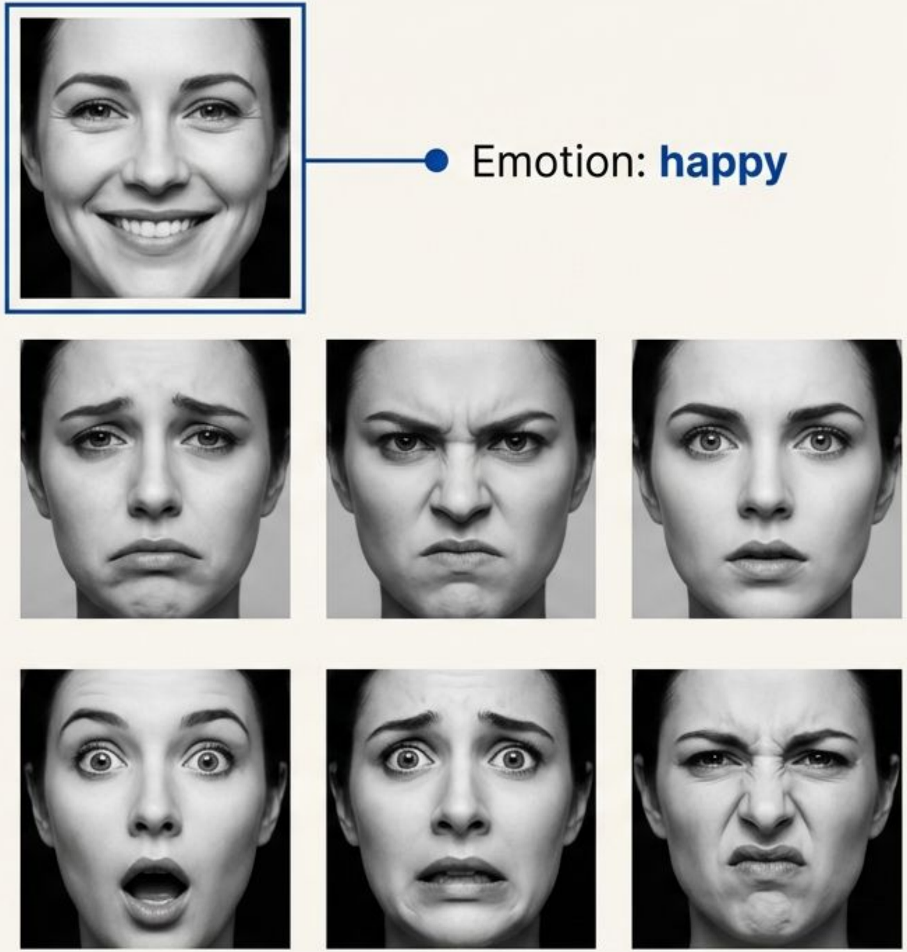
https://sorts.pro/emotion

The Task: Can an Algorithm Read a Human Face?

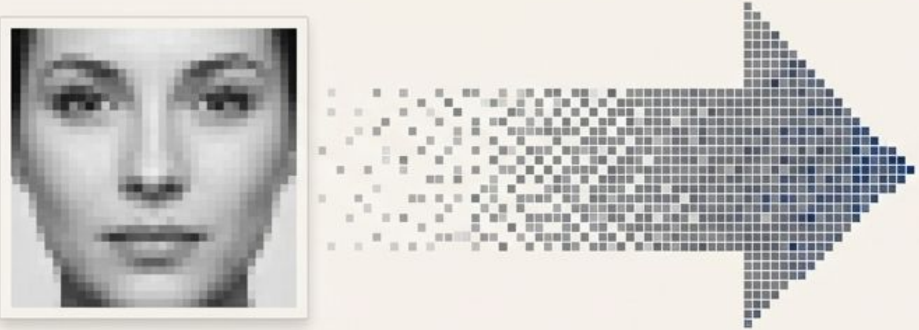
The project goal is to classify images of human faces into distinct emotional categories. We begin with a dataset of 1,575 labeled images, each representing a specific emotion.

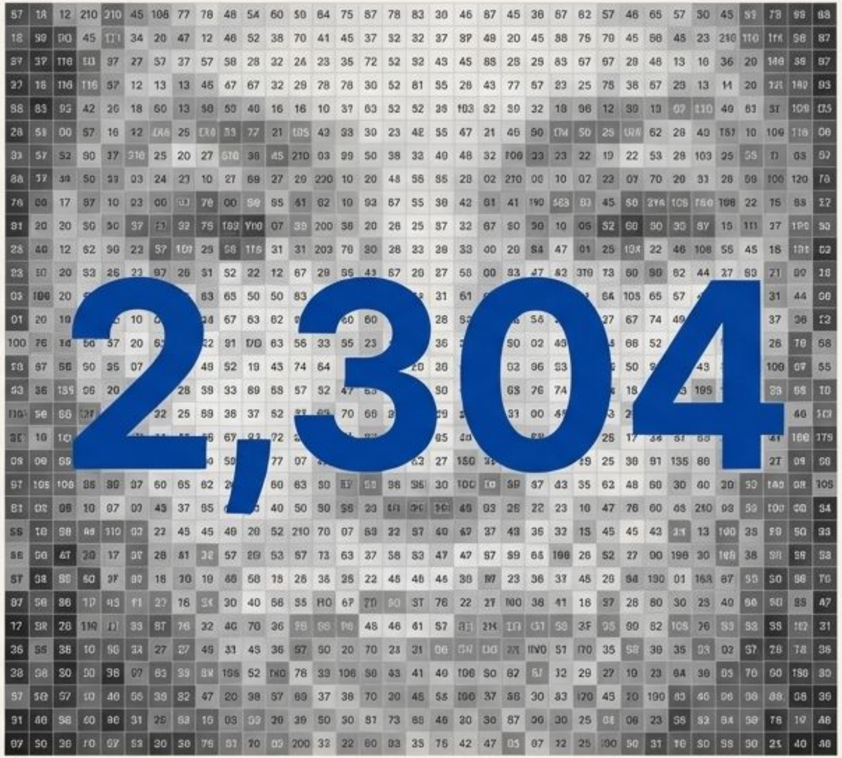
Dataset Size

1,575 images



The Challenge: A Single 48x48 Image Contains 2,304 Raw Features





When a model looks at an image, it doesn't see a face; it sees a grid of numbers. A small 48×48 pixel grayscale image translates to **2,304 individual data points**. This high dimensionality overwhelms most models, making it difficult to distinguish meaningful patterns from random noise.

Too many features, So Need Feature Extraction (HoG/LBP)

Our Solution: Isolate the Signal by Extracting Shape and Texture

Instead of feeding the model raw pixels, we will engineer features that describe the essential visual elements of an expression. We use two powerful techniques:

Histogram of Oriented Gradients (HOG)

To capture the shape of facial features (e.g., the curve of a smile, the furrow of a brow).

Local Binary Patterns (LBP)

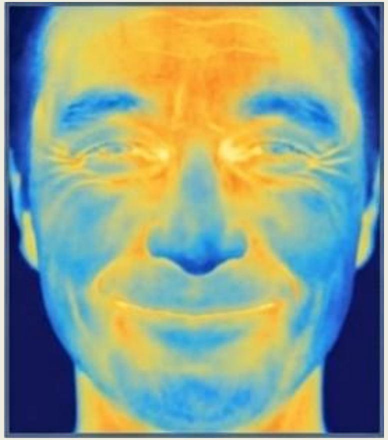
To capture the texture of the skin and micro-patterns.

Histogram of Oriented Gradients (HoG)

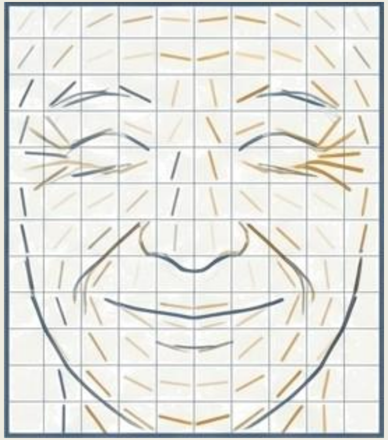
Core Idea: HoG is a feature descriptor that captures an object's essential structure by analyzing the orientation and magnitude of local intensity gradients (i.e., edges). Such as the curves formed by the mouth or eyebrows.

How It Works

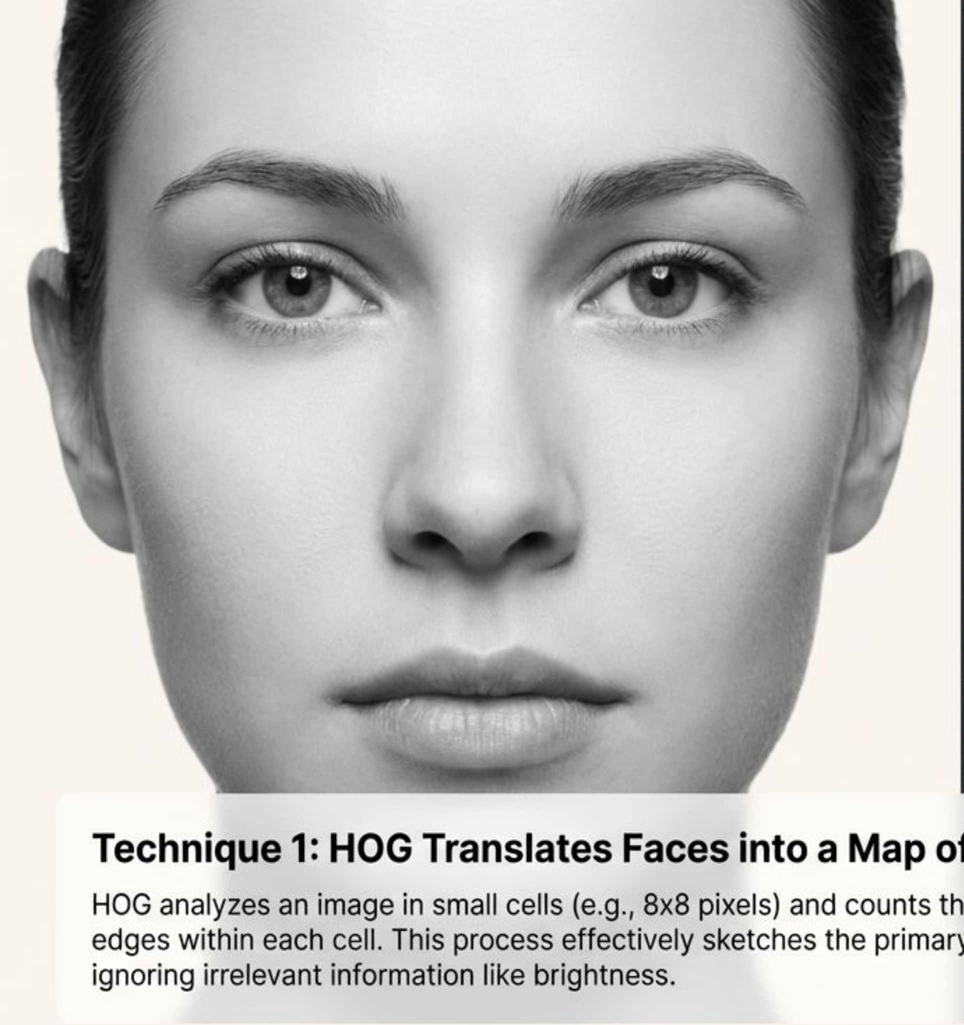
1. Gradient Calculation: The gradient magnitude (edge strength) and orientation (angle) computed/pixel.
2. Cell Aggregation: The image is divided into small regions called "cells" (e.g., 8X8)
3. Block Normalization: Neighboring cells are grouped into larger "blocks" (e.g., 2X2 cells). To improve robustness against local illumination.
4. Final Vector: All normalized block histograms are concatenated into a single, comprehensive feature vector.

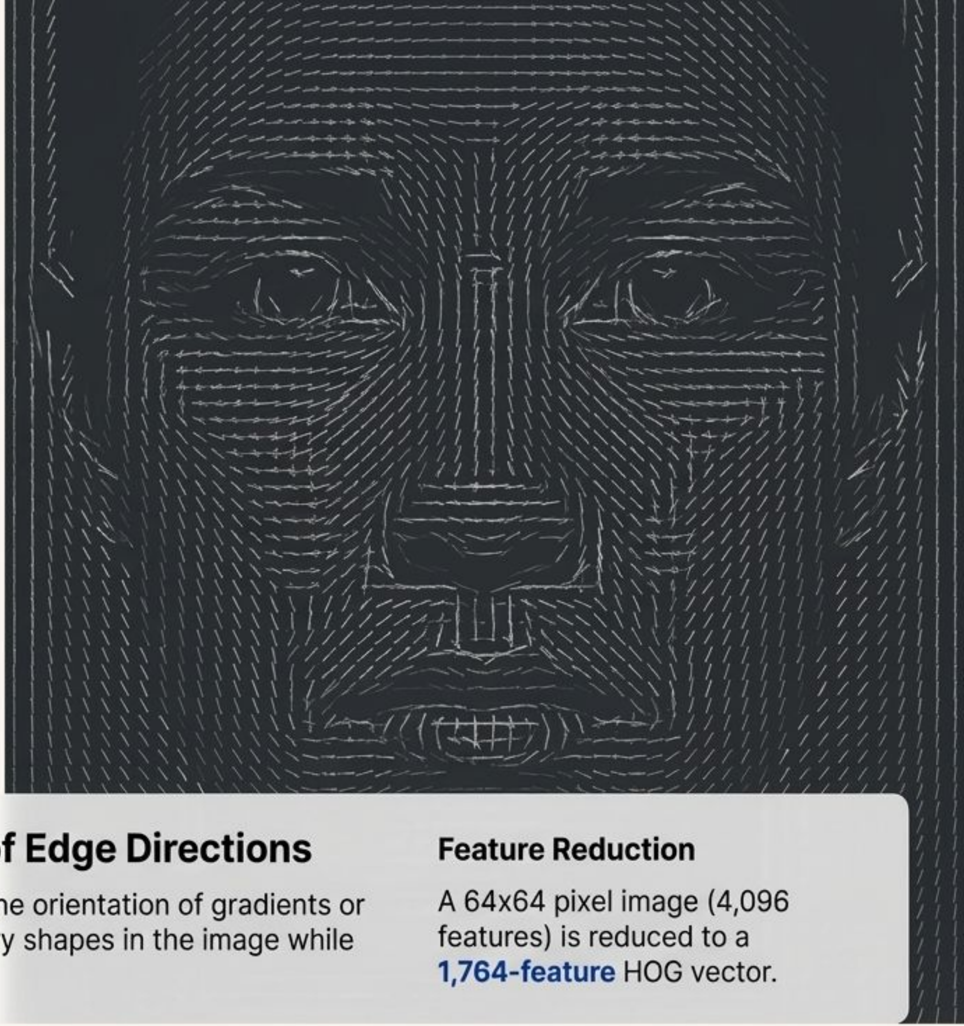


Thermal Image (Smiling Face)



HoG Visualization (Gradient Grid)





Technique 1: HOG Translates Faces into a Map of Edge Directions

HOG analyzes an image in small cells (e.g., 8x8 pixels) and counts the orientation of gradients or edges within each cell. This process effectively sketches the primary shapes in the image while ignoring irrelevant information like brightness.

Feature Reduction

A 64x64 pixel image (4,096 features) is reduced to a **1,764-feature** HOG vector.

Original face

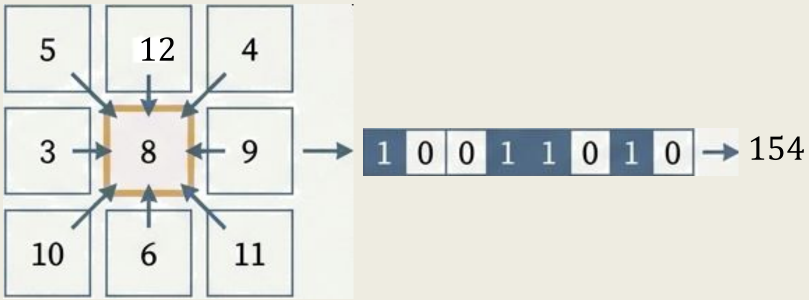
HOG edges

Local Binary Patterns (LBP)

LBP is a powerful texture descriptor known for its computational simplicity and, critically, its robustness to monotonic gray-scale changes caused by illumination variations. It excels at characterizing local micro-patterns, like skin wrinkles or muscle strain.

How It Works

1. For each pixel, a circular neighborhood is selected (e.g., P=8 surrounding points at radius R=l).
2. The intensity of each neighbour pixel is thresholded against the centre pixel's value, generating a binary pattern (e.g., `10011010`).
3. This binary pattern is converted to a decimal number, or LBP code. So-called ‘uniform patterns' (those with at most two bitwise transitions) are often used to reduce the feature vector’s length.
4. The final LBP feature vector is the normalised histogram of these LBP codes over the entire image.



LBP Computation: Center pixel compared with neighbors.

The macro-structure (HoG) and the micro-texture (LBP) of the face are its primary characteristics.

Technique 2: LBP Captures the Micro-Textures of the Face

LBP looks at each pixel and compares it to its immediate neighbors. It generates a binary code based on whether the neighbors are brighter or darker, creating a detailed map of local textures. This is highly effective for capturing subtle skin patterns and wrinkles.

Compact Representation

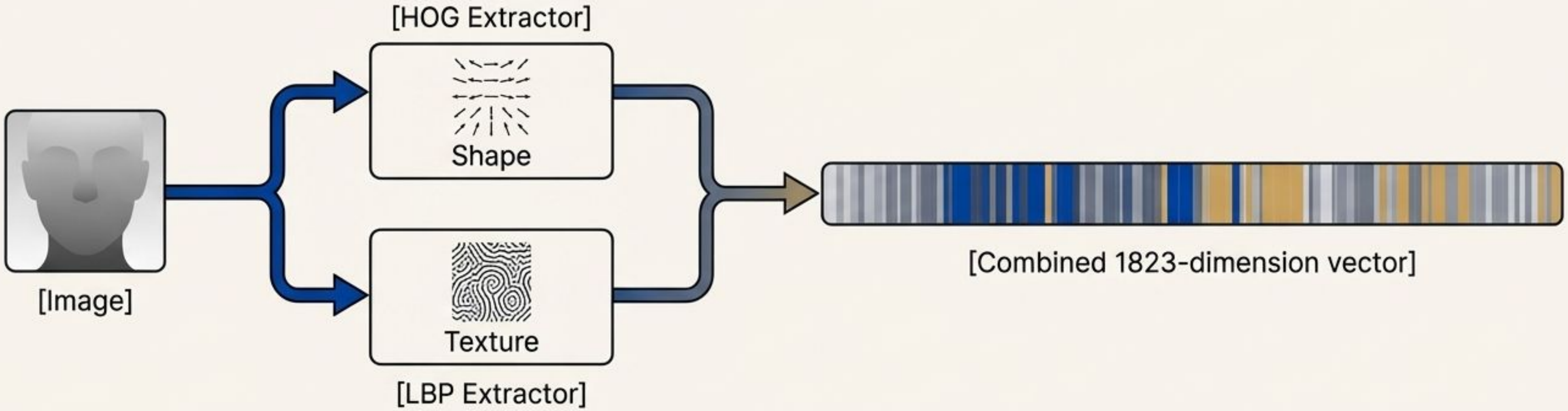
LBP generates a highly compact histogram of **59 features**.

Original

LBP Texture

The Final Fingerprint: A Combined Vector of 1,823 Features

By combining the shape information from HOG with the texture information from LBP, we create a rich, descriptive feature vector for each image. This hybrid approach gives our models a much more nuanced and robust understanding of the facial expression.



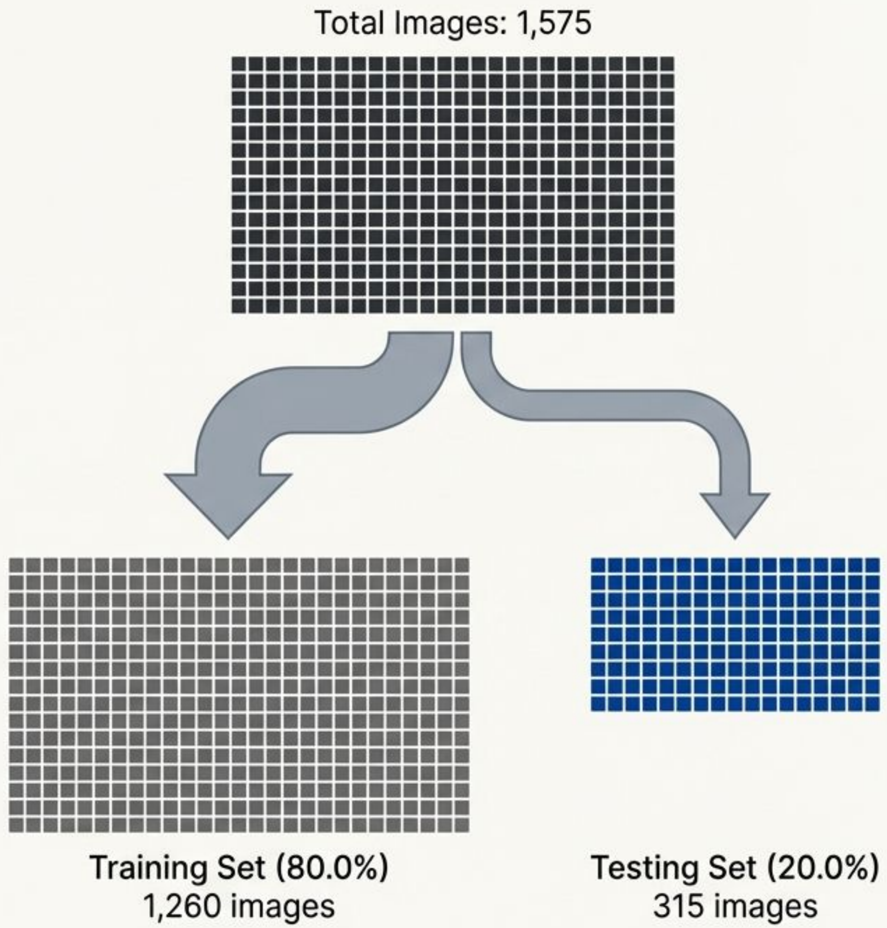
[HOG Features] + [LBP Features] = [Combined Feature Vector]

1,764 + 59 = **1,823 Features per Image**

Building a Fair Test: Splitting Data for Training and Evaluation

To honestly evaluate our models, we separate our dataset into two parts. The models learn patterns from the "training set" and are then tested on the "testing set"—images they have never seen before.

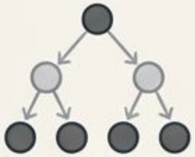
This prevents the model from simply memorizing the data and ensures it can generalize to new examples.



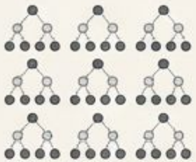
The Contenders: Four Algorithms Compete to Classify Emotions

We will train and evaluate four distinct machine learning models, each with a different strategy for classifying the feature vectors.

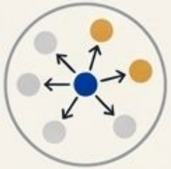
Support Vector Machine (SVM): Finds the best line or plane to separate emotion categories in a high-dimensional space.



Decision Tree: Makes a series of “yes/no” decisions based on feature values, like a flowchart.

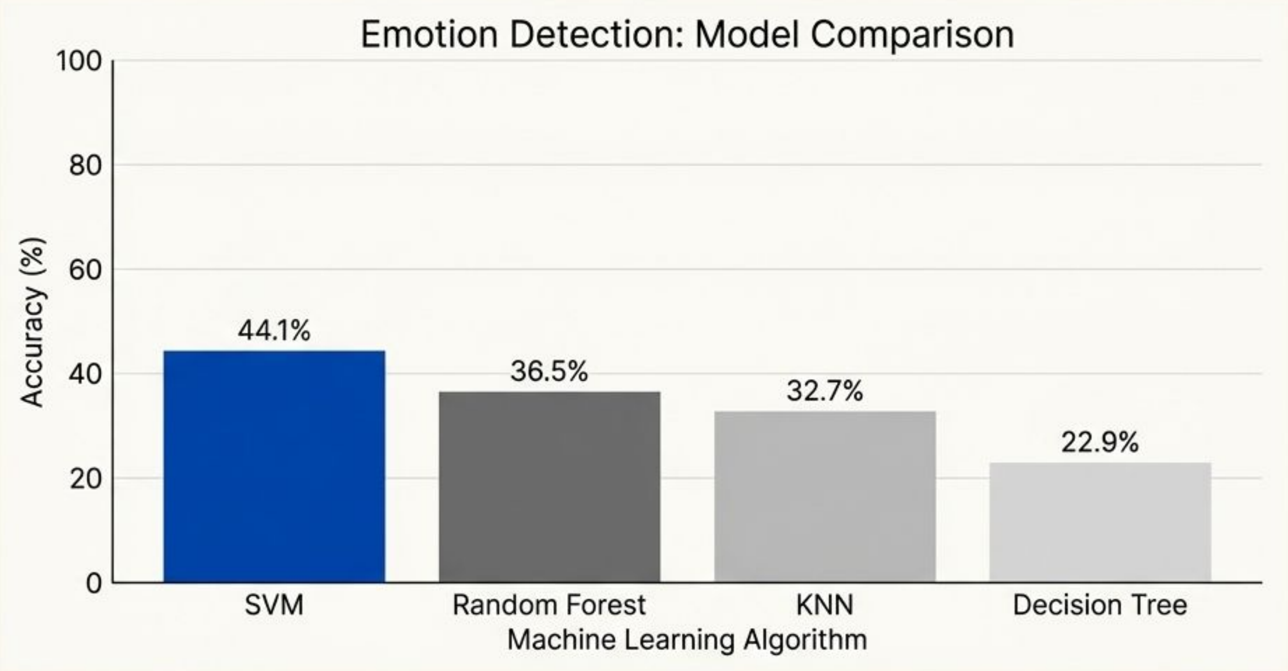


Random Forest: An ensemble of many decision trees that vote on the final classification.



K-Nearest Neighbors (KNN): Classifies an image based on the majority emotion of its 5 closest neighbors in the feature space.

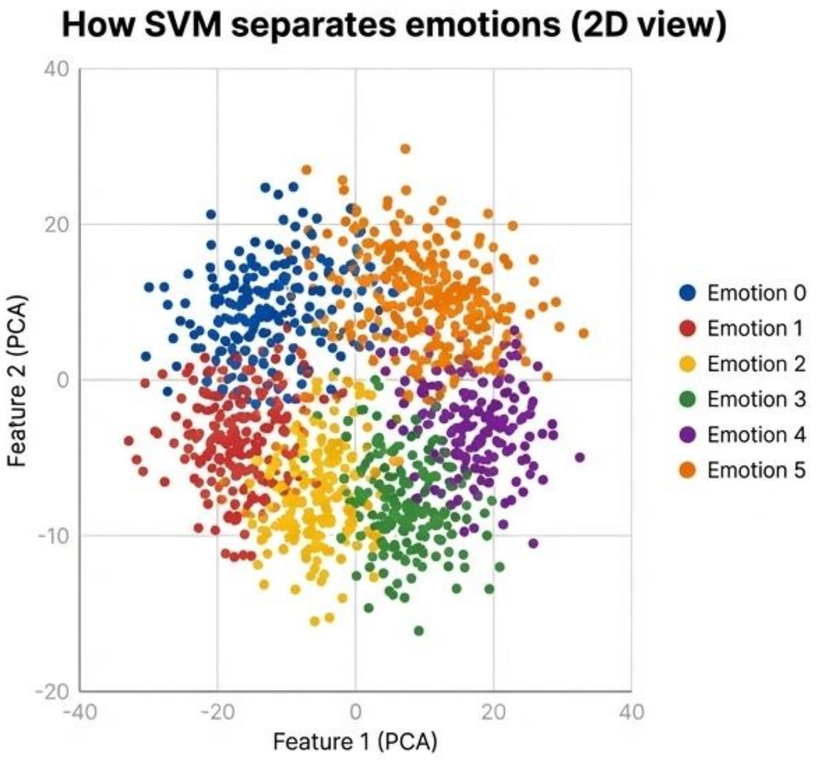
The Verdict: SVM Outperforms Other Models in Classification Accuracy



Best Model: SVM with an accuracy of 44.13% on the test set.

Inside the Winner: How SVM Separates Emotions in Feature Space

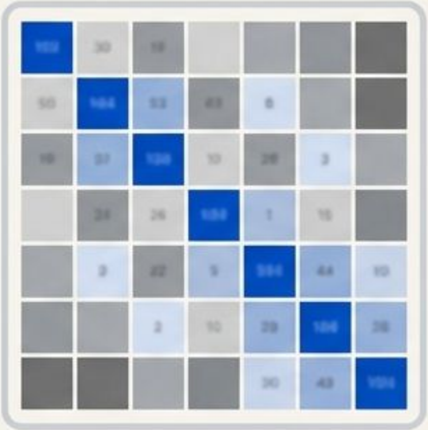
SVM's strength lies in its ability to find an optimal boundary between complex data clusters. The visualization on the right shows a 2D simplification of our 1,823-dimension feature space. Each dot is an image, and SVM attempts to draw lines that best separate the different colored emotion groups.



Beyond Overall Accuracy: Where Exactly Does the Model Succeed or Fail?

A single accuracy score doesn't tell the whole story. To truly understand the model's behavior, we need to see which emotions are correctly identified and which are commonly confused with one another. The confusion matrix provides this detailed, class-by-class breakdown.

Overall Accuracy



Detailed Breakdown

Measuring Performance: Why Accuracy Isn't Enough

To truly diagnose a model's performance, especially with potentially imbalanced emotion classes, we must look beyond simple accuracy and use a more holistic suite of metrics.

The Key Metrics

Precision

Of all the times the model predicted "Happy," how many times was it actually correct? Note: High precision is crucial when a false positive is costly.

Formula: TP / (TP + FP)

Recall (or Sensitivity)

Of all the truly "Sad" faces in the dataset, how many did our model successfully identify?

Note: High recall is critical when a false negative is costly (e.g., failing to detect).

Formula: TP / (TP + FN)

F1-Score

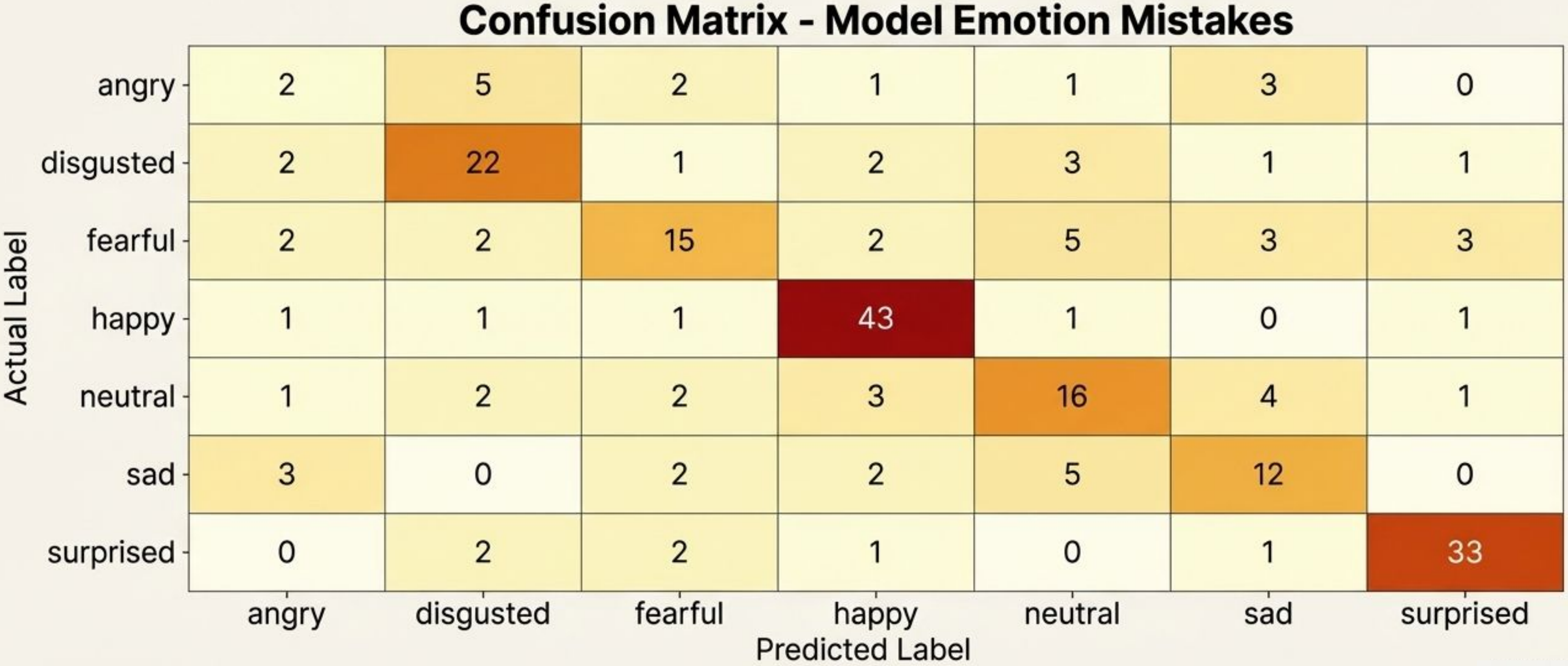
The harmonic mean of precision and recall. It provides a single, balanced score that is preferable to accuracy for imbalanced datasets, as it penalizes models severely if either precision or recall is low.

Formula: 2 * (Precision * Recall) / (Precision + Recall)

	Predicted: Positive	Predicted: Negative
Actual: Positive	True Positive (TP)	False Negative (FN)
Actual: Negative	False Positive (FP)	True Negative (TN)

The Confusion Matrix: Mapping Predictions Against Reality

The diagonal from top-left to bottom-right shows correct predictions. All other cells represent errors, revealing specific confusion patterns. For example, the model correctly identified 43 "happy" faces but struggled significantly with "angry" faces, only getting 2 correct.



A Deeper Analysis: Balancing Precision and Recall

Precision: When the model predicts an emotion, how often is it correct? (High precision for "disgusted" at 0.600)

Recall: Of all the actual instances of an emotion, how many did the model find? (High recall for "happy" at 0.717)

The F1-Score provides a balanced measure. "Happy" and "surprised" are the strongest classes, while "fearful" and "angry" are the weakest.

	precision	recall	f1-score	support
angry	0.143	0.143	0.143	14
disgusted	0.600	0.688	0.641	32
fearful	0.556	0.429	0.484	35
happy	0.705	0.717	0.711	60
neutral	0.333	0.333	0.333	48
sad	0.500	0.414	0.453	29
surprised	0.846	0.917	0.880	36
accuracy			0.438	254
macro avg	0.526	0.520	0.521	254
weighted avg	0.458	0.438	0.447	254

Face Detection

https://sorts.pro/face

Face Emotion Detection

https://sorts.pro/emotionface

Scan for ASIP Lab & PI - Dr. Samiran Das

Mr Sajjan Singh

Mr Ramen Ghosh

Q & A Session