

# Emotion Detection from Images using ML



• Emotion: happy

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.



Gauging raw emotional responses to products and advertisements, providing insights beyond traditional surveys.



Assisting in clinical psychology and autism therapy by quantifying and tracking emotional states.



Evaluating user frustration or satisfaction with software interfaces and websites to create more intuitive designs.



Ensuring safer roads by detecting driver drowsiness or distress through facial cues.

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



2,304

When a model looks at an image, it doesn't see a face; it sees a grid of numbers. A small 48x48 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.

57	18	12	210	210	45	106	77	78	48	54	60	50	84	75	87	78	83	30	46	87	45	38	67	62	57	46	65	57	30	45	98	98	98			
18	99	90	45	11	34	20	47	12	46	52	38	70	41	45	37	32	32	37	89	48	20	45	38	75	70	45	66	45	23	210	110	110	98	87		
37	37	116	116	116	37	27	57	57	58	28	22	24	23	35	72	52	52	43	45	88	28	25	83	67	67	28	48	13	16	36	20	108	38	87		
37	16	116	116	116	37	12	13	13	45	67	67	32	29	78	78	30	81	81	55	28	43	77	57	23	25	78	36	57	23	13	11	20	110	140	93	
88	83	95	42	26	18	60	13	98	50	40	16	10	37	63	52	52	38	103	32	50	32	18	96	12	39	19	97	110	49	83	57	109	15			
28	58	00	57	16	42	006	25	010	19	77	21	015	42	33	30	23	42	55	47	21	46	50	110	50	28	40	161	10	106	116	06					
93	57	52	90	37	316	25	20	27	610	38	45	210	03	99	50	38	33	40	48	32	106	33	23	22	19	22	53	28	103	25	35	17	63	87		
88	17	99	50	31	03	24	23	10	27	58	27	29	220	10	20	48	56	56	28	02	210	00	10	02	22	07	70	20	31	26	98	100	120	70		
76	06	17	97	10	03	00	00	00	76	00	98	61	61	02	10	93	57	55	36	42	81	41	190	68	83	45	50	216	108	108	22	15	83	27		
91	20	20	50	30	37	E3	92	75	193	Y10	07	39	200	36	20	26	25	37	32	67	80	90	10	05	52	60	60	30	67	15	111	27	190	53		
28	46	12	52	90	22	57	101	29	56	116	31	202	70	80	23	29	33	00	20	S4	47	01	25	101	22	46	108	55	45	18	101	03				
23	50	20	32	26	23	97	26	31	52	22	12	67	28	56	43	67	26	27	58	00	93	47	42	310	73	60	89	92	44	27	93	21	89	18		
03	100	20	50	30	37	83	65	50	83	31	61	31	61	31	61	31	61	31	61	31	61	31	61	31	61	31	61	31	61	31	61	31	44	60		
01	20	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10			
100	76	14	50	37	20	62	91	07	63	56	33	55	23	36	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23			
18	87	56	50	35	07	48	92	19	43	74	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64			
03	36	156	06	20	28	39	93	88	68	57	52	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47			
03	56	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37	37			
31	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10			
08	09	69	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77			
97	105	106	85	89	97	60	85	82	29	60	63	80	83	63	98	98	30	100	10	98	47	43	55	62	48	60	30	00	30	50	100	98	108			
81	02	06	10	07	09	45	37	95	01	40	50	50	56	20	11	06	10	49	03	25	22	23	10	47	76	60	05	210	02	50	04	34				
58	10	98	48	10	03	22	45	45	49	20	52	210	70	97	69	22	97	40	67	37	43	36	32	15	45	45	42	31	13	100	28	98	50	93		
56	66	60	60	17	97	28	41	38	57	93	53	67	13	63	37	58	83	47	47	97	99	66	196	26	52	27	90	198	30	108	38	91	98	58		
57	98	99	50	39	18	10	19	68	58	16	26	35	26	22	45	46	38	87	23	36	31	45	29	98	190	01	168	87	95	90	97	70				
87	56	86	10	45	11	27	16	54	30	40	56	55	H0	67	70	60	37	76	22	27	100	36	41	18	57	28	80	30	25	40	06	90	89	47		
17	9R	26	110	11	93	67	76	22	40	70	26	78	08	10	48	46	41	57	57	2M	11	01	98	3F	95	90	82	108	70	53	33	95	10	31		
36	58	36	10	96	34	27	57	49	31	45	36	57	50	20	70	23	31	06	06	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
38	08	08	30	90	97	63	39	8X	165	52	07	76	33	108	58	43	41	40	100	50	62	57	12	29	27	10	23	84	36	05	76	60	150	80		
57	50	57	10	46	35	32	47	20	98	57	68	37	38	70	40	55	100	37	38	30	33	70	45	70	100	63	66	96	88	06	98	96	36			
91	46	98	60	68	31	26	88	16	09	39	39	50	50	50	57	73	68	46	30	38	37	30	30	25	08	06	28	53	56	94	98	78	10	46		
97	50	50	10	07	53	30	38	76	31	03	03	200	32	22	60	03	35	15	42	47	05	07	32	25	00	50	31	16	10	50	98	50	25	40	48	

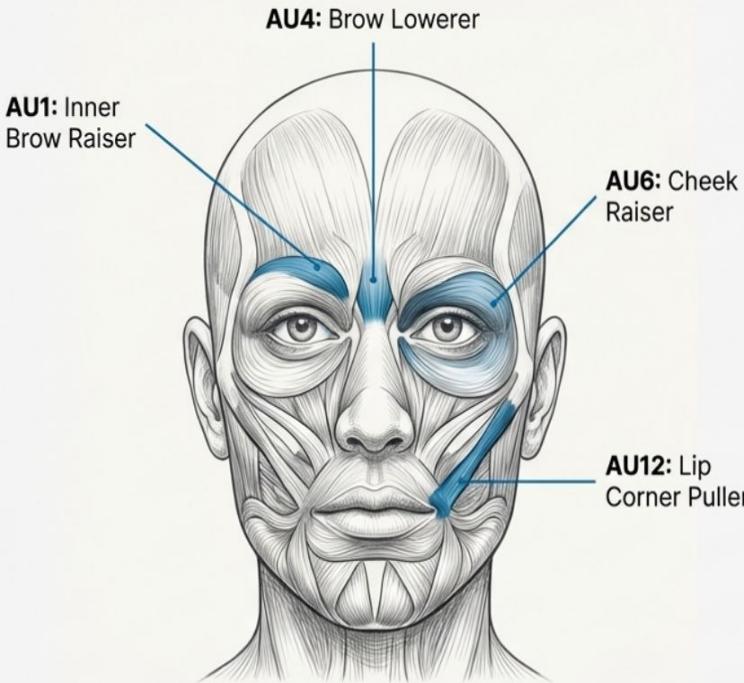
*Too many features, So Need Feature Extraction (HoG/LBP)*

# Feature Detection from Image

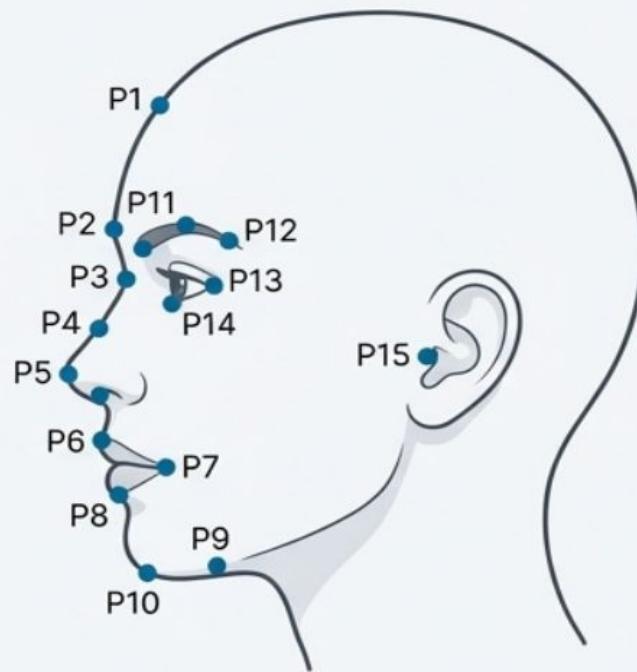


Feature Vector

# Identifying Fiducial Points



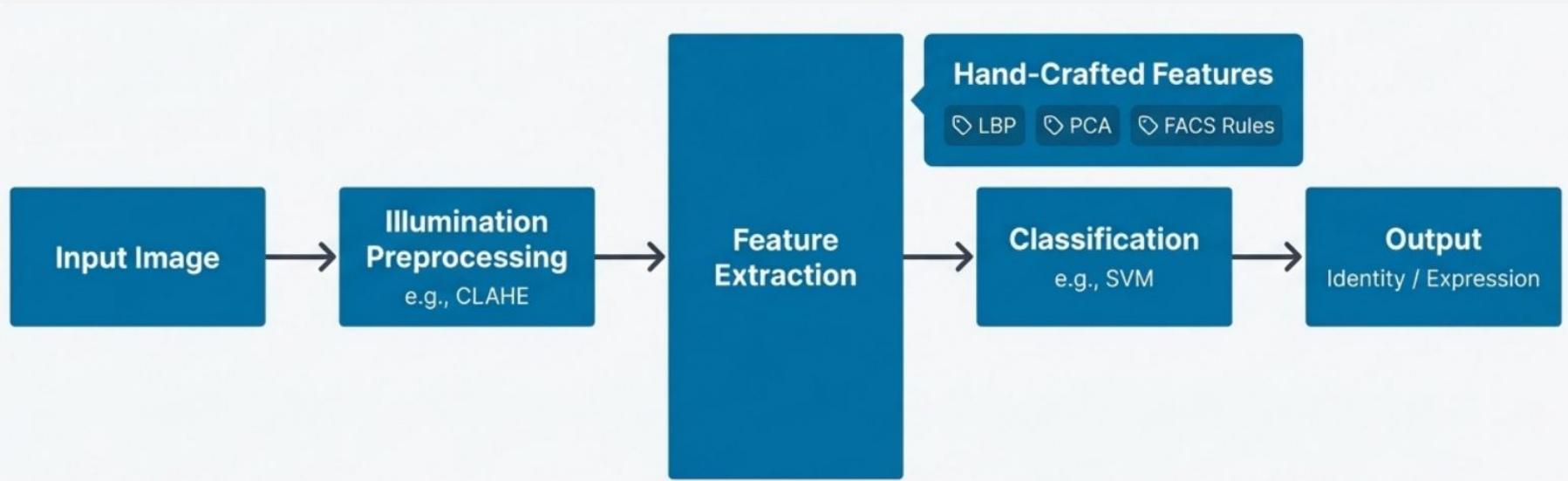
- P1** (top of forehead)
- P2** (eyebrow arcade)
- P3** (root of nose)
- P4** (tip of nose)
- P5** (nostril)
- P6** (upper lip)
- P7** (mouth corner)
- P8** (lower lip)
- P9** (lower jaw)
- P10** (tip of chin)



# Classic (Traditional) Pipeline

Basic schematic which involves

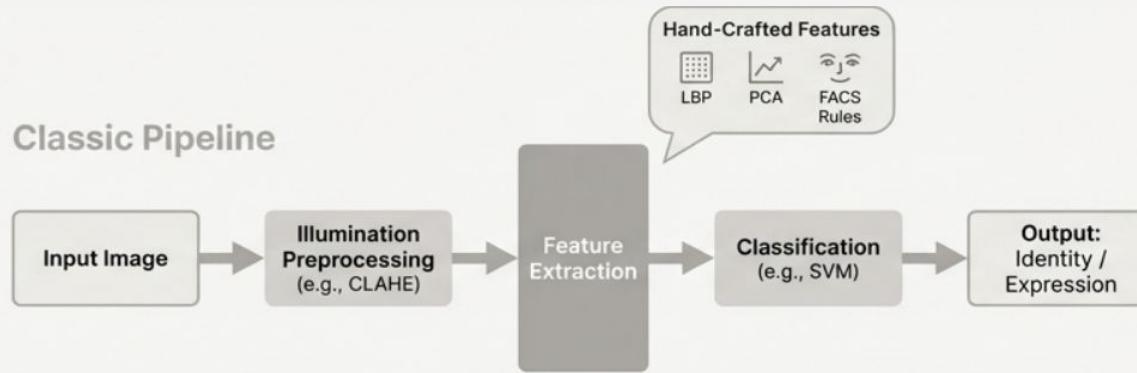
1. Pre-processing (Illumination correction, Scaling, Denoising, etc.)
2. Extraction of features (LBP, HoG, etc)
3. Feeding the feature into ML Classification Models



**Basic schematic for emotion classification**

# Classic (Traditional) Pipeline To Deep Neural Network

A DNN-like CNN learns features automatically.



## Deep Learning Pipeline



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



2,304

When a model looks at an image, it doesn't see a face; it sees a grid of numbers. A small 48x48 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.

57	18	12	210	210	45	106	77	78	48	54	60	50	84	75	87	78	83	30	46	87	45	38	67	62	57	46	65	57	30	45	98	98	98		
18	99	90	45	11	34	20	47	12	46	52	38	70	41	45	37	32	32	37	89	48	20	45	38	75	70	45	66	45	23	210	110	110	98	87	
37	37	116	116	116	37	27	57	37	58	28	22	24	23	35	72	52	52	43	45	88	28	25	83	67	67	29	48	13	16	36	20	108	38	87	
37	16	116	116	116	37	12	13	13	45	67	67	32	29	78	78	30	81	81	55	28	43	77	57	23	25	78	36	57	23	13	11	20	110	140	93
88	83	95	42	26	18	60	13	98	50	40	16	10	37	63	52	52	38	103	32	50	32	18	96	12	39	19	97	110	49	83	57	109	15		
28	58	00	57	16	42	006	25	010	19	77	21	015	42	33	30	23	42	55	47	21	46	50	110	50	28	40	161	10	106	116	06				
93	57	52	90	37	316	25	20	27	610	38	45	210	03	99	50	38	33	40	48	32	106	33	23	22	19	22	53	28	103	25	35	17	63	87	
88	17	99	50	31	03	24	23	10	27	58	27	29	220	10	20	48	56	56	28	02	210	00	10	02	22	07	70	20	31	26	99	100	120	70	
76	06	17	97	10	03	00	00	76	00	99	65	61	02	10	93	57	55	36	42	81	41	190	68	83	45	50	216	108	108	22	15	83	27		
91	20	20	50	30	37	13	92	75	193	190	07	39	200	36	20	25	87	32	67	90	50	10	05	52	60	50	30	67	15	111	27	190	53		
28	46	12	52	90	22	57	101	29	56	116	31	202	70	30	28	23	29	33	00	20	84	47	01	25	101	22	46	108	55	45	18	101	03		
23	50	20	32	26	23	97	26	31	52	22	12	67	28	56	43	67	26	27	58	00	93	47	42	310	73	60	89	92	44	27	93	21	89	18	
03	100	20	50	30	37	13	92	75	193	190	07	39	200	36	20	25	87	32	67	90	50	10	05	52	60	50	30	67	15	111	27	190	53		
100	76	14	50	37	20	62	22	91	07	63	56	33	55	23	36	21	31	61	31	64	105	65	57	37	67	74	49	37	96	12					
18	87	56	50	35	07	48	92	19	43	74	64	36	12	96	53	50	9	48	36	50	02	48	68	52	50	30	67	15	111	27	190	53			
03	36	156	06	20	28	39	93	89	68	57	52	47	43	50	50	68	76	74	31	30	00	3	2	40	103	33	69	10							
03	66	37	37	22	25	89	36	37	52	52	07	70	68	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93			
31	10	10	10	10	10	66	67	93	72	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03	03			
08	09	69	69	69	69	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77			
97	105	106	85	97	60	85	82	29	60	63	80	83	63	98	98	30	100	10	98	47	43	55	62	48	60	30	00	30	50	100	98	108			
81	02	06	10	07	09	45	37	95	01	40	50	50	56	20	11	06	10	49	03	25	22	23	10	47	76	60	05	210	02	50	04	34			
58	10	98	48	10	03	22	45	45	49	49	20	52	210	70	97	69	22	97	40	67	37	43	36	32	15	45	45	42	31	13	100	28	98	50	93
96	66	60	17	97	28	41	38	57	93	53	67	13	63	37	58	83	47	47	97	99	66	196	26	52	27	90	198	30	108	38	91	98	98		
57	98	99	50	30	19	19	68	58	15	26	35	26	22	45	46	38	87	23	36	31	45	29	98	190	01	168	87	95	90	98	70				
87	56	86	10	45	11	27	16	54	30	40	56	55	100	67	70	60	37	76	22	27	100	36	41	18	97	28	80	30	25	40	98	90	89	47	
17	98	26	110	11	93	67	76	22	40	70	26	18	08	10	48	46	41	57	87	24	11	01	98	3F	95	90	82	108	70	53	33	95	93	101	
36	88	36	10	96	34	27	27	49	31	45	36	57	50	20	70	23	31	06	04	04	02	01	00	51	20	35	58	39	35	03	02	57	78	78	96
38	08	80	50	97	63	39	8X	165	52	07	76	33	108	58	43	41	40	100	50	62	57	12	29	27	10	23	84	30	05	76	60	150	80		
57	10	97	10	46	30	24	47	20	98	57	68	37	38	70	48	55	100	37	88	30	33	10	45	70	100	63	46	96	88	98	96				
91	46	98	60	46	31	26	88	16	09	39	39	50	50	51	73	68	46	30	38	87	30	30	25	08	06	23	56	53	94	98	78	10	46		
97	50	90	10	07	53	30	38	76	31	03	03	200	32	22	60	03	35	15	42	47	05	07	32	25	00	50	31	16	10	50	98	50	25	40	48

*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 UNDERSTANDING &  
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](https://sorts.pro/emotion)

## A Machine Learning Journey from Pixels to Predictions

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

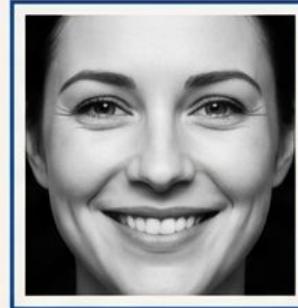
<https://sorts.pro/emotion>

# The Task: Can an Algorithm Read a Human Face?

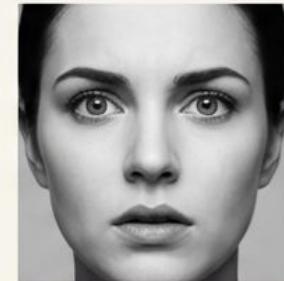
The project goal is to classify images of human faces into distinct emotional categories. We begin with a dataset of 1,575 labeled images, each representing a specific emotion.

Dataset Size

**1,575 images**



Emotion: **happy**



## 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.

A large, bold, blue digital watermark reading "2,34" is centered over a grid of smaller, grayish-blue numbers. The grid consists of approximately 100 rows and 100 columns of numbers, creating a pattern that looks like a barcode or a complex code. The numbers are arranged in a grid format, with some cells containing larger digits than others, suggesting a hierarchical or weighted data structure.

*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)

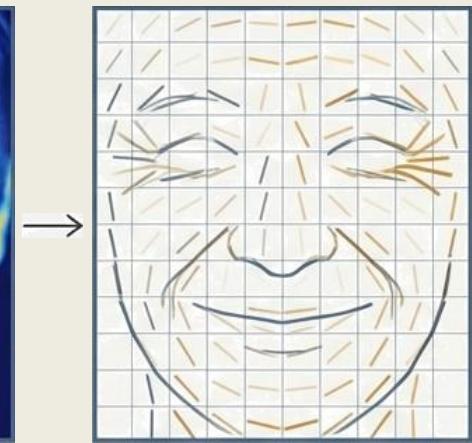
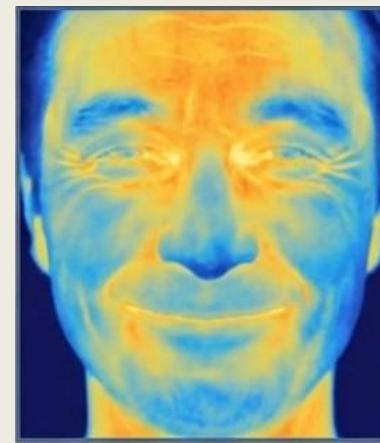
To capture the texture of the skin and micro-patterns.

# 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)

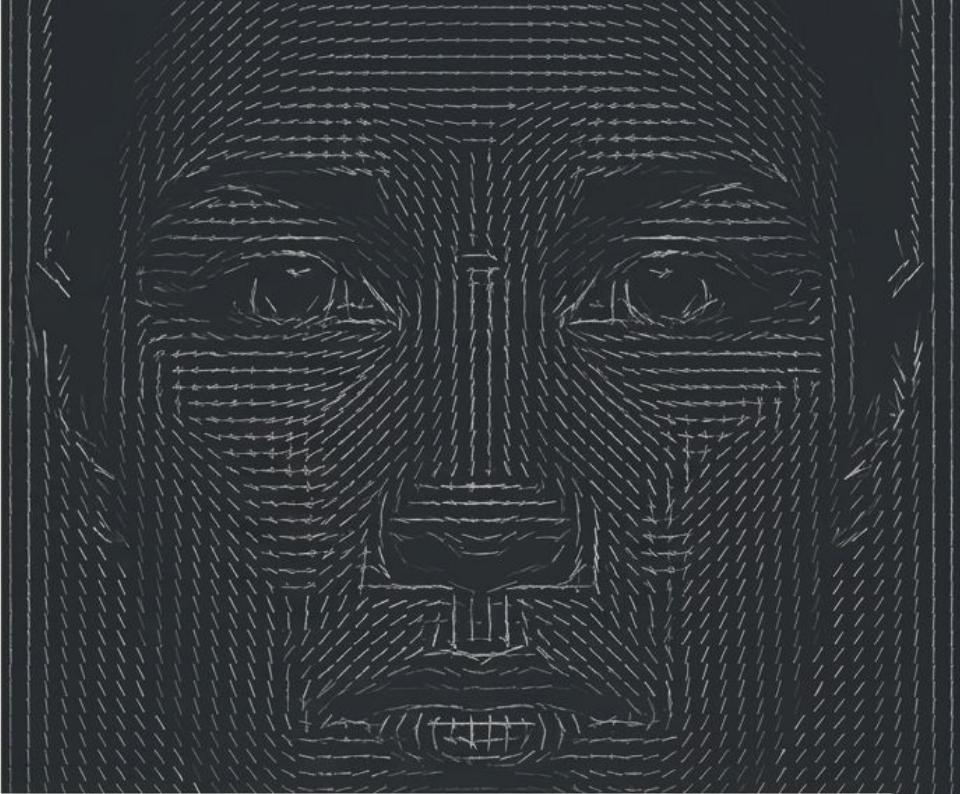
HoG Visualization  
(Gradient Grid)



## 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.

Original face



## Feature Reduction

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

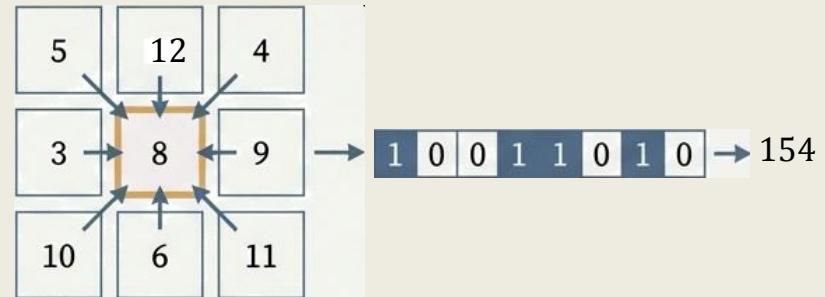
HOG edges

# 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.

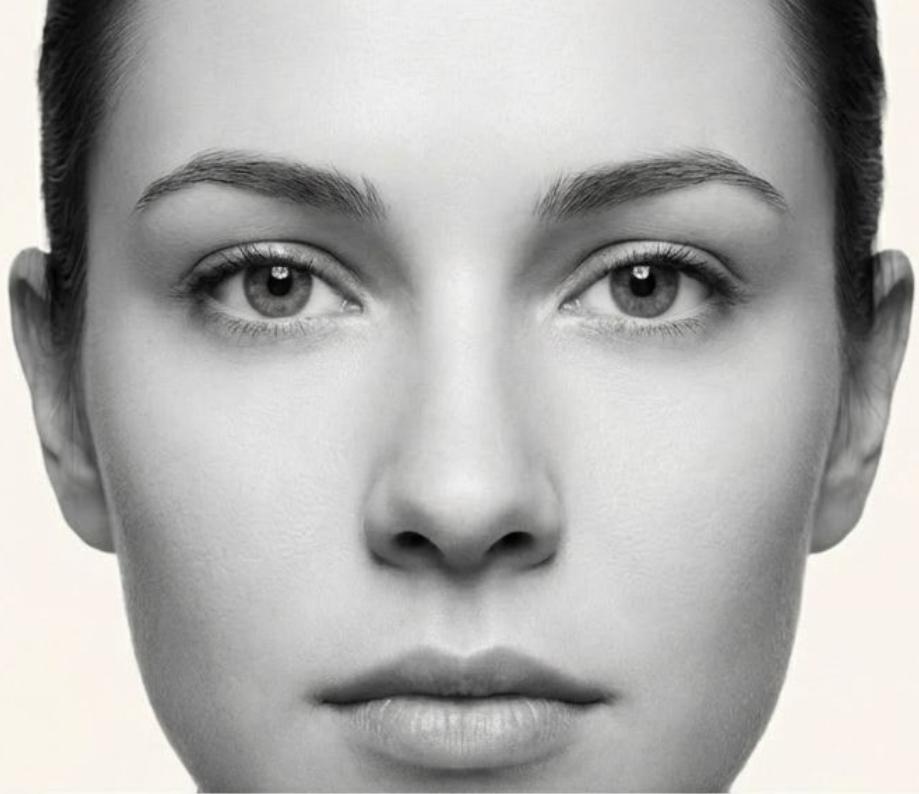
## How It Works

1. For each pixel, a circular neighborhood is selected (e.g., P=8 surrounding points at radius R=1).
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.

Original



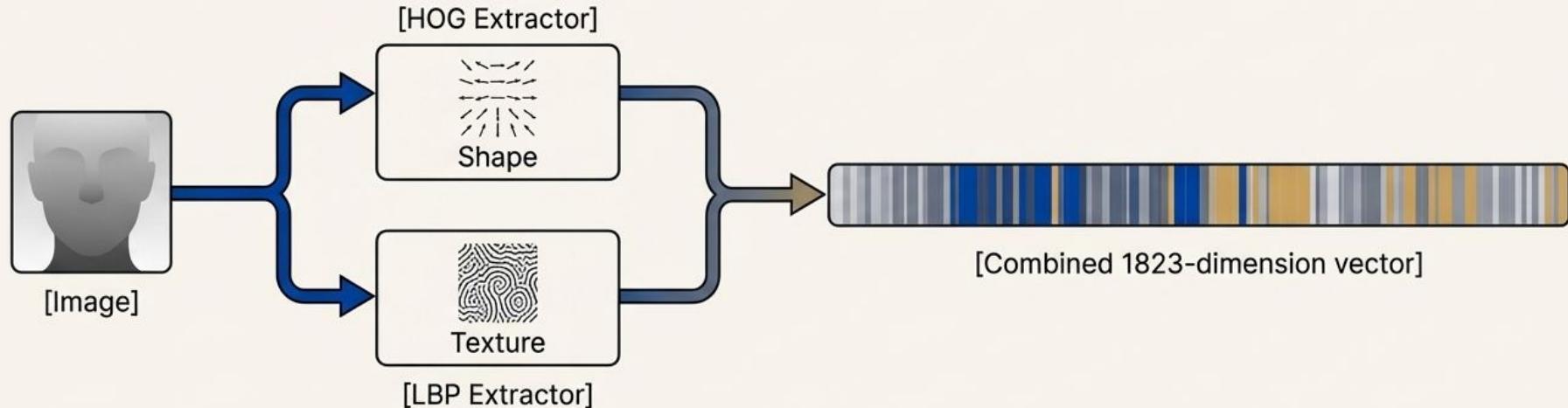
## Compact Representation

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

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.



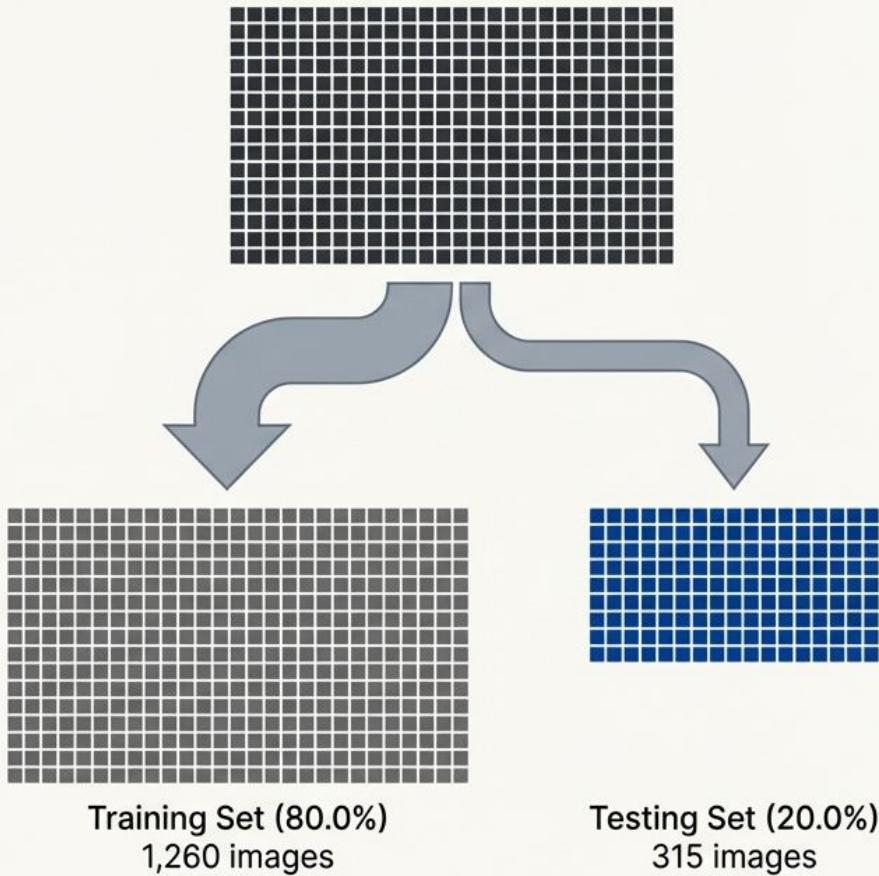
$$\begin{array}{ccc} \boxed{\text{[HOG Features]}} & + & \boxed{\text{[LBP Features]}} \\ 1,764 & + & 59 \end{array} = \boxed{\text{[Combined Feature Vector]}} = \boxed{\text{1,823 Features per Image}}$$

# 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.

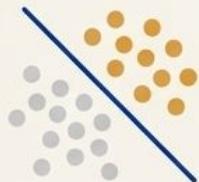
This prevents the model from simply memorizing the data and ensures it can generalize to new examples.

Total Images: 1,575

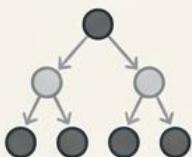


# The Contenders: Four Algorithms Compete to Classify Emotions

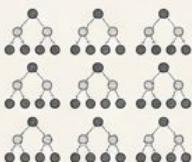
We will train and evaluate four distinct machine learning models, each with a different strategy for classifying the feature vectors.



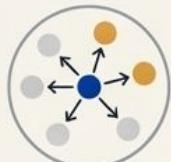
**Support Vector Machine (SVM)**: Finds the best line or plane to separate emotion categories in a high-dimensional space.



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

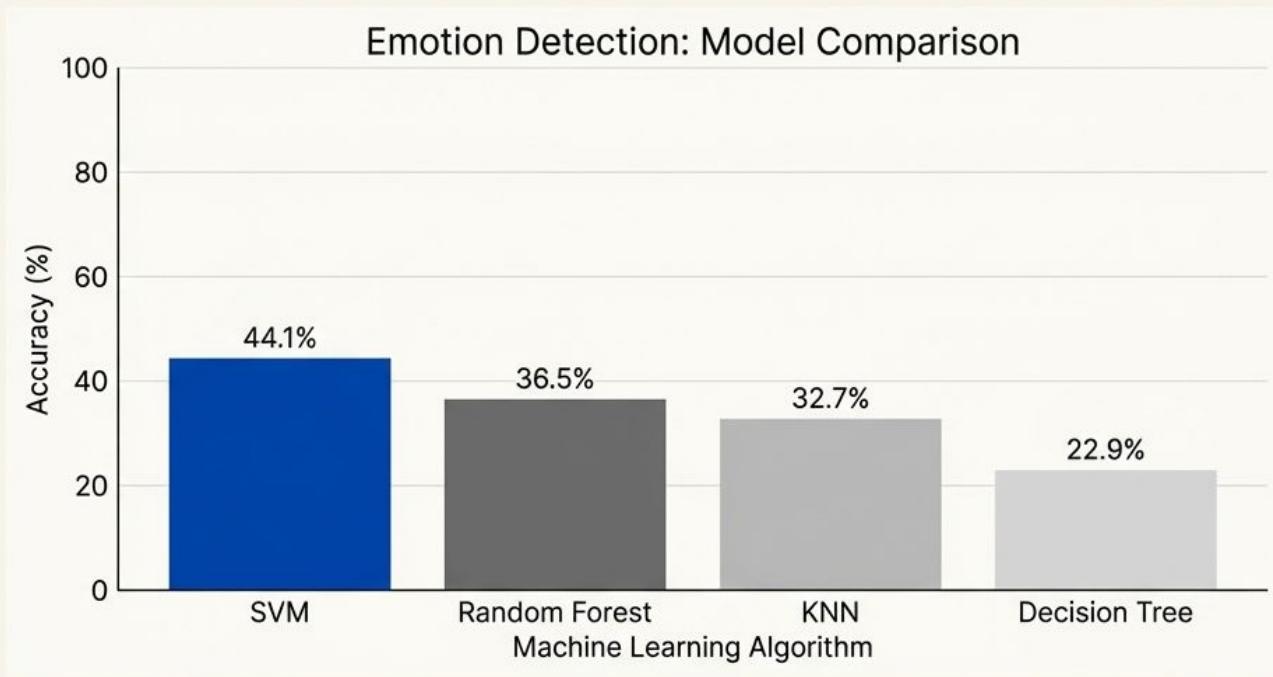


**Random Forest**: An ensemble of many decision trees that vote on the final classification.



**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

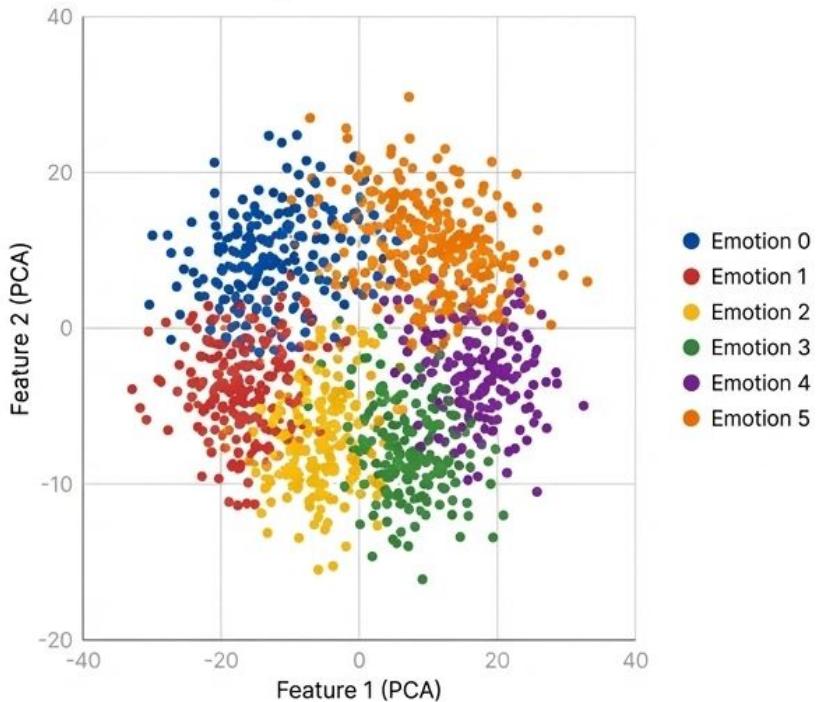


Best Model: SVM with an accuracy of 44.13% on the test set.

# 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.

How SVM separates emotions (2D view)



# 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



True	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral
Anger	100	30	10	5	5	5	5
Disgust	50	100	30	10	5	5	5
Fear	10	30	100	10	20	5	5
Happiness	5	10	10	100	10	10	10
Sadness	5	5	20	10	100	10	10
Surprise	5	5	5	10	10	100	10
Neutral	5	5	5	10	10	10	100

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)	
	False Positive (FP)	True Negative (TN)	
Actual: Negative			

# 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.

**Confusion Matrix - Model Emotion Mistakes**

Actual Label	Predicted Label						
	angry	disgusted	fearful	happy	neutral	sad	surprised
angry	2	5	2	1	1	3	0
disgusted	2	22	1	2	3	1	1
fearful	2	2	15	2	5	3	3
happy	1	1	1	43	1	0	1
neutral	1	2	2	3	16	4	1
sad	3	0	2	2	5	12	0
surprised	0	2	2	1	0	1	33

# A Deeper Analysis: Balancing Precision and Recall

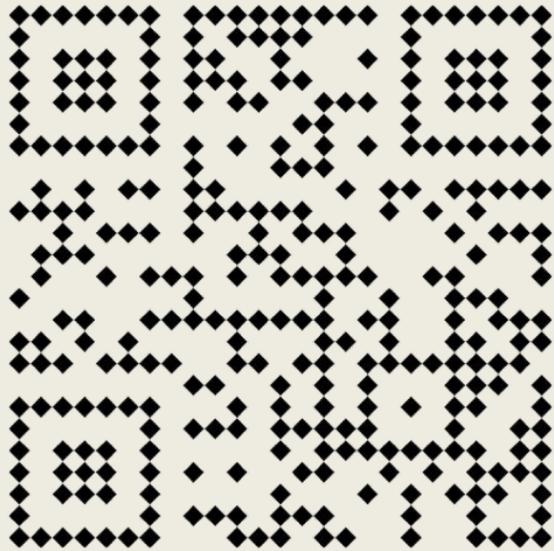
**Precision:** When the model predicts an emotion, how often is it correct? (High precision for "disgusted" at 0.600)

**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

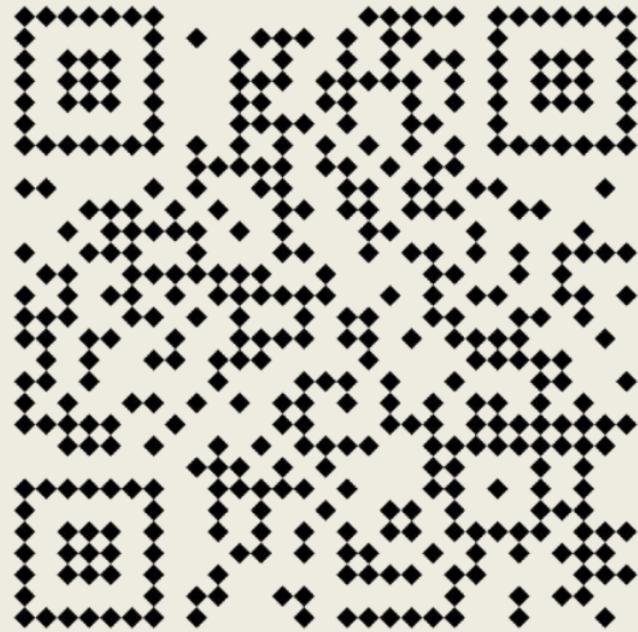
# Face Detection

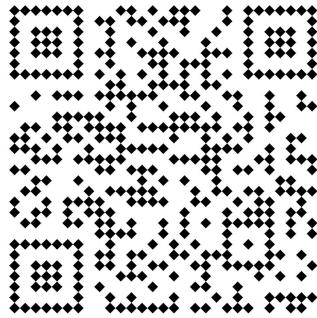


<https://sorts.pro/face>

# Face Emotion Detection

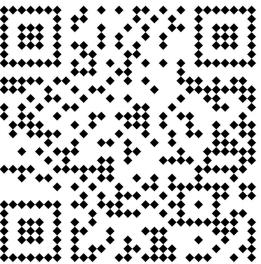
<https://sorts.pro/emotionface>





Scan for ASIP  
Lab & PI -  
Dr. Samiran Das

Mr Sajjan Singh



Mr Ramen Ghosh

# Q & A Session

