

CSE (AIML)

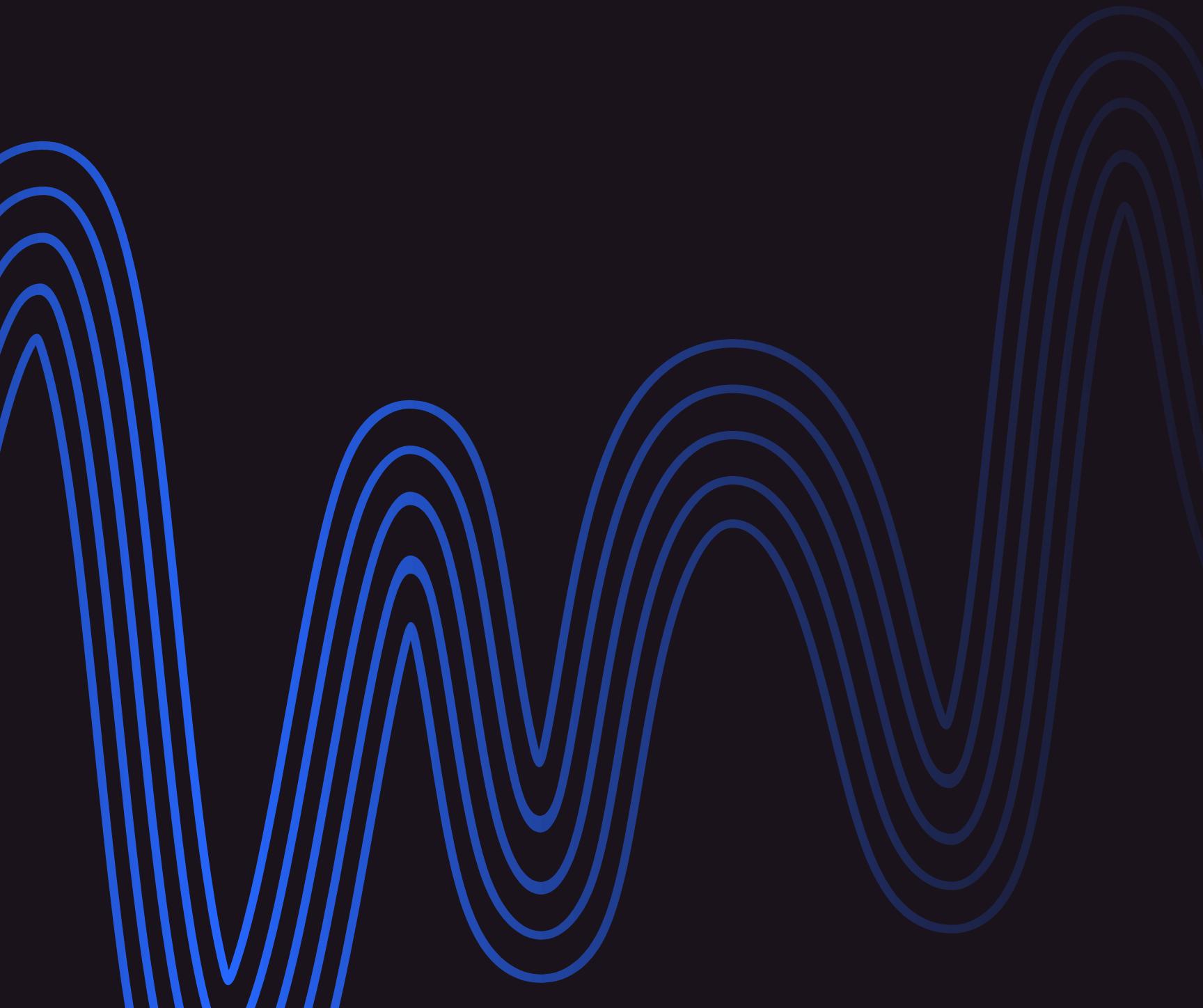
SENTIMENTAL ANALYSIS

AI Project

Presented By Abhimanyu & Deepak Kumar

22AM003 | 22AM013

AGENDA

A large, abstract graphic on the left side of the slide consists of several concentric, wavy blue lines that curve upwards and outwards from the bottom left corner towards the center.

Problem Statement

3

Framework

4

Procedure

5

Datasets

6

Data info

7

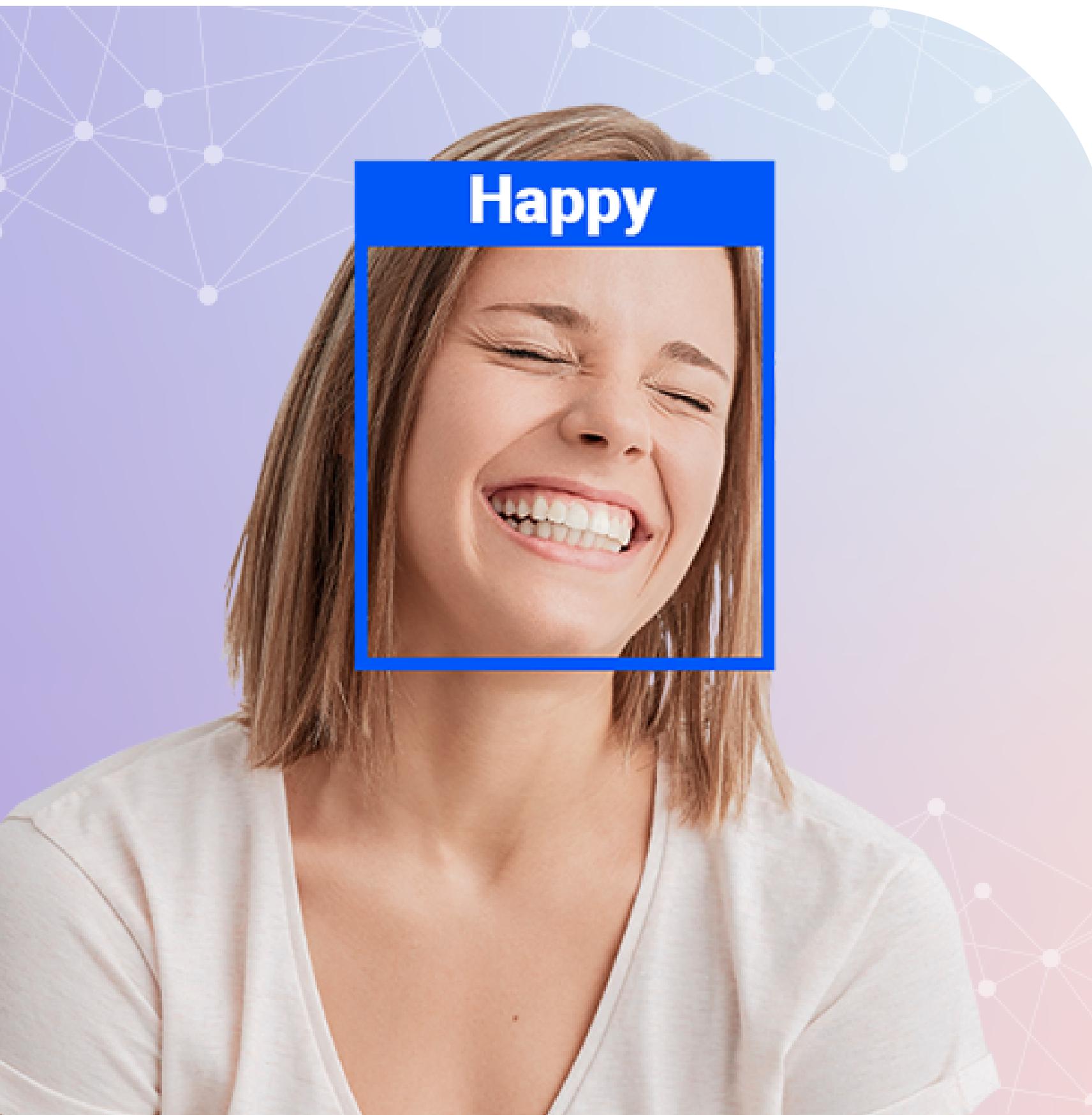
project coding

11

Final Output

13

Problem Statement



to capture and analyze facial expressions in order to determine the predominant emotion displayed, providing valuable insights for various applications such as user experience research, customer feedback analysis, and emotion-driven interactive systems.

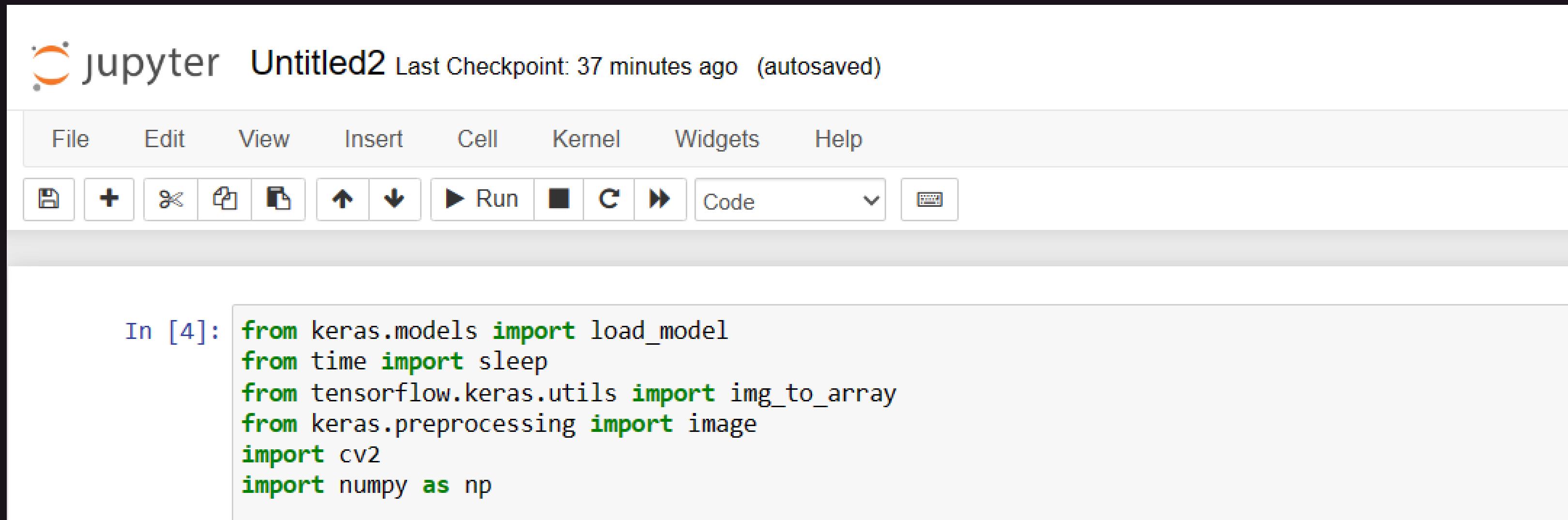
FRAMEWORK

Our project aims to Design and develop a system that utilizes a **webcam to perform real-time emotion detection** on human faces. The system should be able to accurately detect and classify emotions such as **anger, happiness, sadness, neutrality, and surprise** from the facial expressions captured by the webcam.



STEP BY STEP PROCEDURE

import the required libraries :



The screenshot shows a Jupyter Notebook interface with the title "jupyter Untitled2 Last Checkpoint: 37 minutes ago (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for file operations like save, new, and delete, and code execution options like Run, Cell, and Kernel. The code cell In [4] contains the following Python code:

```
In [4]: from keras.models import load_model  
from time import sleep  
from tensorflow.keras.utils import img_to_array  
from keras.preprocessing import image  
import cv2  
import numpy as np
```

Data sets :

```
In [10]: import pandas as pd  
face_rec=pd.read_csv("E:/AI_Project/fer2013/fer2013.csv")  
face_rec.head()
```

Out[10]:

	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

```
In [11]: face_rec.tail()
```

Out[11]:

	emotion	pixels	Usage
35882	6	50 36 17 22 23 29 33 39 34 37 37 37 39 43 48 5...	PrivateTest
35883	3	178 174 172 173 181 188 191 194 196 199 200 20...	PrivateTest
35884	0	17 17 16 23 28 22 19 17 25 26 20 24 31 19 27 9...	PrivateTest
35885	3	30 28 28 29 31 30 42 68 79 81 77 67 67 71 63 6...	PrivateTest
35886	2	19 13 14 12 13 16 21 33 50 57 71 84 97 108 122...	PrivateTest

In []:



jupyter Untitled Last Checkpoint: 3 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help



35882	6	50 36 17 22 23 29 33 39 34 37 37 37 39 43 48 5...	PrivateTest
35883	3	178 174 172 173 181 188 191 194 196 199 200 20...	PrivateTest
35884	0	17 17 16 23 28 22 19 17 25 26 20 24 31 19 27 9...	PrivateTest
35885	3	30 28 28 29 31 30 42 68 79 81 77 67 67 71 63 6...	PrivateTest
35886	2	19 13 14 12 13 16 21 33 50 57 71 84 97 108 122...	PrivateTest

In [12]: `face_rec.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35887 entries, 0 to 35886
Data columns (total 3 columns):
 #   Column   Non-Null Count   Dtype  
--- 
 0   emotion   35887 non-null    int64  
 1   pixels    35887 non-null    object 
 2   Usage     35887 non-null    object 
dtypes: int64(1), object(2)
memory usage: 841.2+ KB
```

```
In [13]: face_rec.isnull().sum()
```

```
Out[13]: emotion      0  
pixels        0  
Usage         0  
dtype: int64
```

In [15]: `face_rec.describe()`

Out[15]:

	emotion
count	35887.000000
mean	3.323265
std	1.873819
min	0.000000
25%	2.000000
50%	3.000000
75%	5.000000
max	6.000000

```
In [17]: face_rec['emotion'].value_counts()
```

```
Out[17]: 3    8989  
6    6198  
4    6077  
2    5121  
0    4953  
5    4002  
1     547  
Name: emotion, dtype: int64
```

The main labels of it can be divided into 7 types: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral

Coding 1st part:

jupyter run.py 6 hours ago

File Edit View Language

```
1 from keras.models import load_model
2 from time import sleep
3 from tensorflow.keras.utils import img_to_array
4 from keras.preprocessing import image
5 import cv2
6 import numpy as np
7
8 face_classifier = cv2.CascadeClassifier('E:\AI_Project\face_detection.xml')
9 classifier = load_model('E:\AI_Project\Emotion_Detection.h5')
10
11 class_labels = ['Angry','Happy','Neutral','Sad','Surprise']
12
13 cap = cv2.VideoCapture(0)
14
15
16
17 while True:
18     # Grab a single frame of video
19     ret, frame = cap.read()
```

Coding 2nd part:

jupyter run.py 6 hours ago

Logout

File Edit View Language Python

```
16
17 while True:
18     # Grab a single frame of video
19     ret, frame = cap.read()
20     labels = []
21     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
22     faces = face_classifier.detectMultiScale(gray, 1.3, 5)
23
24     for (x,y,w,h) in faces:
25         cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0), 2)
26         roi_gray = gray[y:y+h, x:x+w]
27         roi_gray = cv2.resize(roi_gray, (48,48), interpolation=cv2.INTER_AREA)
28
29
30         if np.sum([roi_gray])!=0:
31             roi = roi_gray.astype('float')/255.0
32             roi = img_to_array(roi)
33             roi = np.expand_dims(roi, axis=0)
34
35         # make a prediction on the ROI, then lookup the class
36
37         preds = classifier.predict(roi)[0]
38         print("\nprediction = ",preds)
39         label=class_labels[preds.argmax()]
40         print("\nprediction max = ",preds.argmax())
41         print("\nlabel = ",label)
42         label_position = (x,y)
43         cv2.putText(frame,label,label_position, cv2.FONT_HERSHEY_SIMPLEX, 2, (0,255,0), 3)
44     else:
45         cv2.putText(frame, 'No Face Found', (20,60), cv2.FONT_HERSHEY_SIMPLEX, 2, (0,255,0), 3)
46     print("\n\n")
47     cv2.imshow('Emotion Detector',frame)
48     if cv2.waitKey(1) & 0xFF == ord('q'):
49         break
50
51 cap.release()
52 cv2.destroyAllWindows()
```

Final Output

