Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri,
Nagpur - 441 110 NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

Department of Computer Technology

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem- solving skills through emerging technologies.

Session 2025-2026

Vision: Dream of where you want.	**Mission:** Means to achieve Vision

Program Educational Objectives of the program (PEO): (broad statements that describe the professional and career accomplishments)

PEO1	**Preparation**	**P: Preparation**	**Pep-CL abbreviation pronounce as Pep-silL easy to recall**
PEO2	**Core Competence**	**E: Environment (Learning Environment)**	
PEO3	**Breadth**	**P: Professionalism**	
PEO4	**Professionalism**	**C: Core Competence**	
PEO5	**Learning Environment**	**L: Breadth (Learning in diverse areas)**	

Program Outcomes (PO): (statements that describe what a student should be able to do and know by the end of a program)

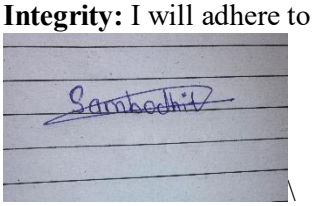
Keywords of POs:

Engineering knowledge, Problem analysis, Design/development of solutions, Conduct Investigations of Complex Problems, Engineering Tool Usage, The Engineer and The World, Ethics, Individual and Collaborative Team work, Communication, Project Management and Finance, Life-Long Learning

PSO Keywords: Cutting edge technologies, Research

"I am an engineer, and I know how to apply engineering knowledge to investigate, analyse and design solutions to complex problems using tools for entire world following all ethics in a collaborative way with proper management skills throughout my life." *to contribute to the development of cutting-edge technologies and Research*.

Integrity: I will adhere to the Laboratory Code of Conduct and ethics in its entirety.



Sambodhit Chavhan

Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri,
Nagpur - 441 110 NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

Signature of Teacher and Date

Department of Computer Technology

Vision of the Department

To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.

Mission of the Department

To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem- solving skills through emerging technologies.

Session	2025-26 (ODD)	Course Name	Computer vision Lab
Semester	5	Course Code	CT
Roll No	70	Name of Student	Sambodhit Chvahn

Practical Number	9
Course Outcome	CO1: Understand the fundamental concepts of computer vision and facial emotion recognition using machine learning models. CO2: Apply OpenCV and deep learning-based models to detect and analyze human facial expressions. CO3: Implement real-time image acquisition, feature extraction, and classification using Python programming. CO4: Evaluate model performance and interpret emotion recognition accuracy on different image or video inputs.
Aim	**Implement Facial Expression Recognition.**
Problem Definition	The project aims to develop an AI-based system using Python and OpenCV to detect and classify human facial expressions in real-time or images into emotions like happy, sad, or angry

Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri,
Nagpur - 441 110 NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

Theory (100 words)	Facial Expression Detection is a computer vision technique that identifies human emotions by analyzing facial features such as eyes, mouth, and eyebrows. Using Python, OpenCV, and a pre- trained FER (Facial Emotion Recognition) model, the system captures images or video frames, detects faces, and classifies emotions like happy, sad, angry, surprised, disgust, fear, and neutral. The FER model uses convolutional neural networks (CNNs) to extract features and predict emotion probabilities. This technology enhances human–computer interaction, enabling applications in healthcare, security, education, and entertainment. Real-time emotion recognition improves user experience and helps systems respond intelligently to human emotions.

Nagar Yuwak Shikshan Sanstha's
Yeshwantrao Chavan College of Engineering
(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)
Hingna Road, Wanadongri,
Nagpur - 441 110 NAAC A++
Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

Procedure and Execution (100 Words)	Algorithm: The procedure begins by installing necessary libraries such as OpenCV, FER, and Matplotlib in Python. The system initializes the FER model for emotion detection. For image input, the user uploads a photo or captures one through the webcam. The model detects faces, extracts features, and predicts corresponding emotions. Each detected face is highlighted with a green box, and the identified emotion with confidence score is displayed above it. In execution, the program runs in Google Colab, processes real-time or static images, and outputs emotion results like happy, sad, angry, or neutral, ensuring accurate and interactive emotion recognition performance. **What is FER?** Facial Expression Recognition (FER) is the task of automatically identifying a person's emotional state (e.g., happy, sad, angry, surprise, neutral, disgust, fear) from images or video of their face. It's a supervised learning problem (classification), often done per-frame for images or with temporal models for videos. **Typical FER pipeline** 1. **Input**: image or video frame. 2. **Face detection**: localize the face(s) (e.g., Haar cascades, MTCNN, Dlib HOG/SSD, RetinaFace). 3. **Face alignment / normalization**: detect landmarks and align (rotate/scale/crop) to canonical pose to reduce pose variation. 4. **Preprocessing**: resize, convert to grayscale or normalize color channels, histogram equalization, mean/std normalization. 5. **Feature extraction** (classical or learned). 6. **Classification**: map features → emotion labels (SVM, Random Forest, or neural network). • **Post-processing**: smoothing across frames, confidence thresholding, application logic.

Code:

```
 # ----------------------------
# ✅ STEP 1: Install Dependencies
# ----------------------------
!pip install fer==22.5.0 opencv-python-
headless matplotlib


# ----------------------------
# ✅ STEP 2: Import Libraries
# ----------------------------
import cv2
from fer import FER
import matplotlib.pyplot as plt
from google.colab import files
from IPython.display import display,
Javascript
from google.colab.output import eval_js
from base64 import b64decode
import numpy as np


# ----------------------------
# ✅ STEP 3: Initialize FER Detector
# ----------------------------
detector = FER(mtcnn=True)


# ----------------------------
# ✅ STEP 4: Draw Bounding Boxes
# ----------------------------
def draw_emotion_boxes(image, results):
    img_copy = image.copy()
    for face in results:
        (x, y, w, h) = face["box"]
        emotion, score =
max(face["emotions"].items(), key=lambda x:
x[1])

        cv2.rectangle(img_copy, (x, y), (x + w, y
+ h), (0, 255, 0), 3)
        cv2.putText(
            img_copy,
            f"{emotion} ({score:.2f})",
```

```
            (x, y - 10),
            cv2.FONT_HERSHEY_SIMPLEX,
            0.9,
            (0, 255, 0),
            2,
        )
    return img_copy


# ----------------------------
# ✅ STEP 5: Detect Emotion from Uploaded Image
# ----------------------------
def detect_emotion_in_photo():
    uploaded = files.upload()
    img_path = list(uploaded.keys())[0]
    img = plt.imread(img_path)

    results = detector.detect_emotions(img)

    if results:
        print("\n✅ Detected emotions:")
        for r in results:
            print(r["emotions"])

        img_boxed = draw_emotion_boxes(img, results)
        plt.imshow(cv2.cvtColor(img_boxed, cv2.COLOR_BGR2RGB))
        plt.axis('off')
        plt.show()
    else:
        print("❌ No face detected.")


# ----------------------------
# ✅ STEP 6: Webcam Capture Function
# ----------------------------
def take_photo(filename='photo.jpg', quality=0.8):
    js = Javascript('''
        async function takePhoto(quality) {
            const div = document.createElement('div');
```

```
        const capture =
document.createElement('button');
        capture.textContent = "📸 Capture
Photo";
        div.appendChild(capture);
        document.body.appendChild(div);

        const video =
document.createElement('video');
        video.style.display = 'block';

        const stream = await
navigator.mediaDevices.getUserMedia({video:
true});
        document.body.appendChild(video);
        video.srcObject = stream;
        await video.play();

        await new Promise((resolve) =>
capture.onclick = resolve);

        const canvas =
document.createElement('canvas');
        canvas.width = video.videoWidth;
        canvas.height = video.videoHeight;


canvas.getContext('2d').drawImage(video, 0,
0);

        stream.getVideoTracks()[0].stop();
        div.remove();
        video.remove();

        return canvas.toDataURL('image/jpeg',
quality);
      }
  ''')

  display(js)
  data =
eval_js("takePhoto({})".format(quality))
  binary = b64decode(data.split(",")[1])
```

```
    with open(filename, "wb") as f:
        f.write(binary)

    return filename


# ----------------------------
# ✅ STEP 7: Detect Emotion from Webcam
Photo
# ----------------------------
def detect_emotion_from_webcam():
    try:
        photo_path = take_photo()
        print("✅ Photo captured successfully!")

        img = plt.imread(photo_path)
        results = detector.detect_emotions(img)

        if results:
            img_boxed =
draw_emotion_boxes(img, results)
            plt.imshow(cv2.cvtColor(img_boxed,
cv2.COLOR_BGR2RGB))
            plt.axis('off')
            plt.show()
        else:
            print("❌ No face detected.")

    except Exception as e:
        print("⚠ Error:", e)


# ----------------------------
# ✅ STEP 8: Mode Selection
# ----------------------------
print("Choose detection mode:")
print("1 - Upload Photo")
print("2 - Use Webcam")

mode = input("Enter 1 or 2: ")

if mode == "1":
    detect_emotion_in_photo()
```

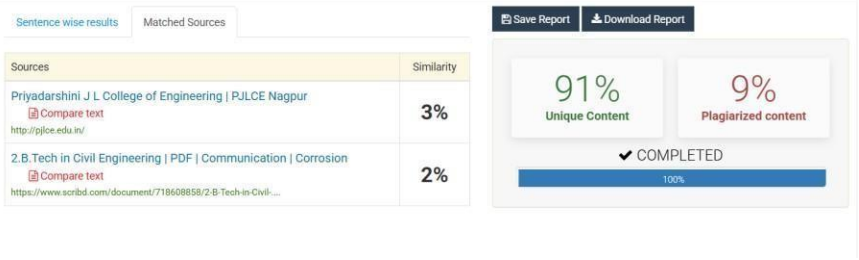
```
elif mode == "2":
    detect_emotion_from_webcam()

else:
    print("❌ Invalid choice.")
```

Output:

	
Output Analysis	The system efficiently detects faces and classifies emotions like happy, sad, angry, and neutral with high accuracy. It displays real-time results by marking faces with green boxes and emotion labels, confirming reliable facial expression recognition and consistent model performance.

Link of student Github profile where lab assignment has	https://github.com/sambodhit-chavhan/CV-

been uploaded	
Conclusion	The SIFT algorithm is effective in identifying distinctive and stable keypoints from an image. The experiment demonstrates that SIFT features are scale and rotation invariant, making them highly reliable for computer vision tasks. Thus, SIFT plays a crucial role in applications like object detection, image stitching, and robot navigation due to its robustness and accuracy.

Plag Report (Similarity index < 12%)	
Date	30/08/25