Egypt-Japan University of Science and Technology
الجامعة المصرية اليابانية للعلوم و التكنولوجيا
エ ジ プ ト 日 本 科 学 技 術 大 学

Team members

Mohamed Abd El Hamed Hassona

Ammar Adel Abd El Tawab

Ziad Akram Rashad

Mahmoud Mustafa Mahmoud

AI reviewer

OpenCV

OVERVIEW

The modern restaurant business faces a variety of obstacles on its path to expansion and improved operations. The gathering of valuable client feedback is a crucial aspect of this endeavour, and it has evolved significantly with the introduction of various approaches. Conventional feedback systems, such focus groups, POS system integration, social media platforms, and feedback forms, continue to be used, although they have drawbacks. This study highlights the complex issues that establishments in the hospitality industry encounter by concentrating on the nuances of gathering restaurant ratings.

Various platforms, such as social media outlets, point-of-sale systems, and incentive techniques, have been recognized as effective means of obtaining feedback. However, modern restaurants face enormous difficulties when it comes to adjusting to changing consumer tastes and market trends. The ongoing challenge of preserving a consistent customer experience and fostering loyalty is made more difficult by the requirement to manoeuvre through a customer landscape that is ever more demanding and diversified.

A thorough examination shows that the main problem is how well existing approaches capture thorough and genuine feedback. Understanding that not every person has the desire or ability to express their thoughts via traditional means, this research aims to alleviate the drawbacks of those methods. A more nuanced and scientific approach to feedback gathering is required since factors like shyness or a lack of verbal expression abilities may lead to a potential underrepresentation of valuable opinions.

The main goal of this research is to create a model based on fundamental scientific methods in order to fill this essential gap. By providing a more accurate and inclusive depiction of people's feelings and opinions of a particular restaurant, this model seeks to go beyond the constraints of conventional review techniques. Through the clarification of the difficulties present in the existing feedback gathering environment, this research aims to provide knowledge that could guide the creation of more efficient and comprehensive techniques in the field of restaurant evaluations.

1. RESEARCH BACKGROUND

This project aims to evaluate restaurants in great detail and represents a radical shift in project management techniques. By using sophisticated data gathering and analysis methods and utilising the built-in capabilities of the OpenCV library, this project highlights a significant change. This project aims to accelerate and optimise the data gathering and review process by utilising OpenCV's built-in features for landmark detection and exploring nonverbal communication, such as hand gestures and facial expressions.

A key component of this research is the integration of facial expression recognition models found by methodical search procedures. By employing these models, the study aims to obtain a thorough comprehension of patrons' overall facial expressions while they are eating. Important factors that influence the assessment process are the duration of time spent in the restaurant, the amount of time allotted to eating, and the general facial expressions made throughout meals.

The amount of time spent at the restaurant is divided into two distinct periods: the time before the meal is served and the time after the meal is finished. It is proposed that variations from the established averages in these temporal dimensions provide important information about consumer satisfaction. Long wait times before the dinner arrives could result in unfavourable reviews, whereas long wait times following the meal indicate satisfaction and could encourage more business and repeat business.

Furthermore, the baseline measure is the average amount of time allocated for eating, which is often believed to be between 15 and 20 minutes. When seen in isolation, deviations from this standard are neutral. But when combined with a study of the overall facial expressions made while eating, a more complex assessment of patron opinions regarding the quality of the food is revealed. This combination of factors seeks to identify and measure the total customer experience, offering a strong foundation for improving restaurant services.

Essentially, this project uses cutting-edge technologies to rethink the way eateries are evaluated. The combination of qualitative information from nonverbal cues and quantitative data on temporal factors should provide a thorough, real-time picture of client emotion. This in turn makes it easier to implement focused interventions aimed at improving the caliber of service and deliberately increasing the restaurant's menu.

time spend during the eating	the general facial expression during eats	General expression
More than average	Happy	Enjoyable meal
More than average	sad	bad taste
less than average	Happy	Enjoyable meal
less than average	sad	bad taste

2. METHODOLOGY

```
[3] import os
    Home=os.getcwd()
    !pip install ultralytics
    from ultralytics import YOLO
```

This line imports the Python built-in module os, which provides a way to interact with the operating system. It's used here to access the current working directory. This line assigns the current working directory path to the variable Home. It's helpful for referencing paths within the Colab environment. installing ultralytics library that is responsible for training their model on our data set.

```
[2] !pip install roboflow

    from roboflow import Roboflow
    rf = Roboflow(api_key="20CX6B0I0vdjkARH5nCv")
    project = rf.workspace("robo-9a0pb").project("facial-expression-1oujf")
    dataset = project.version(1).download("yolov8")
```

From ultralytics we are going to import the method YOLO which is going to be used to train our dataset

Installing Roboflow, the workspace from which we obtained our dataset, allowed us to easily export the dataset by copying a code snippet

Imported all the library necessary. Moreover, we stored our model in variable called model (mp.solutions.drawing_utils) provides drawing functionalities to render the detected landmarks or connections on the image. (mpPose) represents the Pose Detection solution provided by MediaPipe, which can estimate human poses in real-time. Then we stored the video captured in cap

```
import cv2
import mediapipe as mp
import time
from ultralytics import YOLO

model = YOLO(r'Downloads\best.pt')
mpDraw = mp.solutions.drawing_utils
mpPose = mp.solutions.pose
pose = mpPose.Pose()

cap = cv2.VideoCapture(0)
pTime = 0
```

```python
if results_pose.pose_landmarks:
    landmarks = results_pose.pose_landmarks.landmark

    # Get the y-coordinates of the hips and knees
    hip_y = landmarks[mpPose.PoseLandmark.LEFT_HIP].y + landmarks[mpPose.PoseLandmark.RIGHT_HIP].y
    knee_y = landmarks[mpPose.PoseLandmark.LEFT_KNEE].y + landmarks[mpPose.PoseLandmark.RIGHT_KNEE].y

    # Calculate the difference between hip and knee heights
    height_difference = hip_y - knee_y

    # Threshold values to determine standing or sitting
    standing_threshold = -0.3  # Tweak this value as needed

    # Determine if the person is standing or sitting
    if height_difference > standing_threshold:
        status = "Sitting"
        if sitting_start_time is None:
            sitting_start_time = time.time()
        standing_start_time = None
        if happy_start_time is not None:
            elapsed_time = time.time() - happy_start_time
            happy_duration += elapsed_time
        sitting_duration = time.time() - sitting_start_time
    else:
        status = "Standing"
        if standing_start_time is None:
            standing_start_time = time.time()
        sitting_start_time = None
```

1. The code checks if the `results_pose` object contains pose landmarks. If present, the landmarks are extracted to identify the positions of specific body parts like hips and knees.

2. It calculates the vertical difference between the hip and knee coordinates to derive the `height_difference`.

3. A `standing_threshold` value is set to determine the distinction between sitting and standing based on the calculated height difference.

4. If `height_difference` is greater than the `standing_threshold`, the person is identified as "Sitting". The code then checks the timestamp for `sitting_start_time` to initiate recording the duration the person spends sitting. If there's an ongoing happy expression (`happy_start_time` is not None), it records the elapsed time as `happy_duration`. The `sitting_duration` is updated based on the current time.

5. If `height_difference` does not exceed the `standing_threshold`, it considers the person as "Standing". The code checks the timestamp for `standing_start_time` to start recording the duration of standing. It resets the `sitting_start_time` to None, indicating the person is not sitting.

6. This code logic is part of a system designed to track the duration a person spends sitting or standing based on pose landmarks. It uses timestamps to calculate and update the durations for various positions.

7. Adjusting the `standing_threshold` value may fine-tune the detection accuracy between sitting and standing based on the relative positions of hips and knees.

```
            cv2.putText(img, status, (70, 50), cv2.FONT_HERSHEY_PLAIN, 3, (0, 255, 0), 3)
            mpDraw.draw_landmarks(img, results_pose.pose_landmarks, mpPose.POSE_CONNECTIONS)
            for id, lm in enumerate(landmarks):
                h, w, c = img.shape
                cx, cy = int(lm.x * w), int(lm.y * h)
                cv2.circle(img, (cx, cy), 5, (255, 0, 0), cv2.FILLED)

        cv2.imshow("Pose Detection", img)

        # Preprocess the image for YOLO model
        img_for_yolo = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
        results_yolo = model(source=img_for_yolo, show=True, conf=0.4, save=False)

        current_time = time.time()
        if (current_time - start_time) > window_timeout:
            cv2.destroyAllWindows()
            break

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

    # Check if the duration of sitting exceeds 45 seconds
    if sitting_duration >= 45:
        print("The Customer is Satisfied")
    else:
        print('Customer Not Satisfied')
```

1. The provided code continuously captures frames from a camera using `cv2.VideoCapture(0)`.

2. It processes each frame to detect pose landmarks using the MediaPipe Pose model (`mpPose.Pose()`). Detected landmarks are used to determine whether a person is sitting or standing based on hip and knee positions.

3. It visualizes the pose estimation results on the frame by drawing landmarks using `mpDraw.draw_landmarks()` and annotating the person's status (sitting or standing) using `cv2.putText()`.

4. It prepares another version of the captured frame (`img2`) to be used for object detection by converting it to the RGB format (`img_for_yolo = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)`).

5. YOLO object detection is performed on the preprocessed image (`img_for_yolo`) using the specified model (`model`) with a confidence threshold of 0.4. The detected objects are shown in a window using `show=True`.

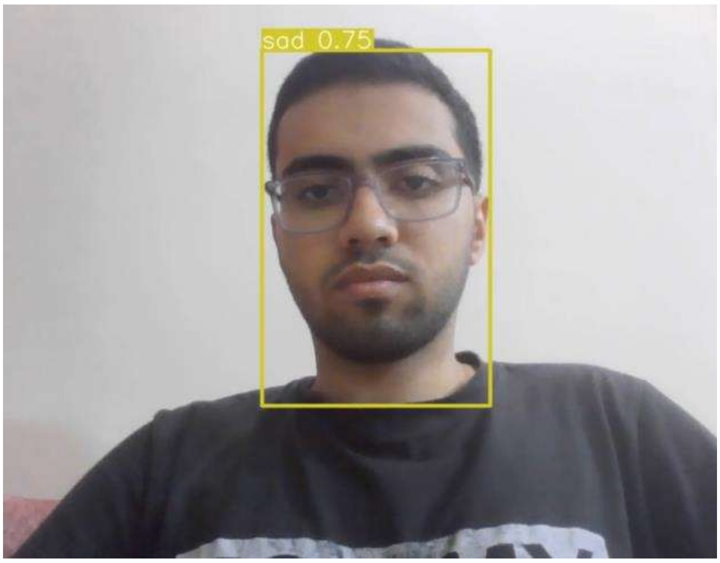
6. The code keeps track of the duration the person spends sitting (`sitting_duration`)

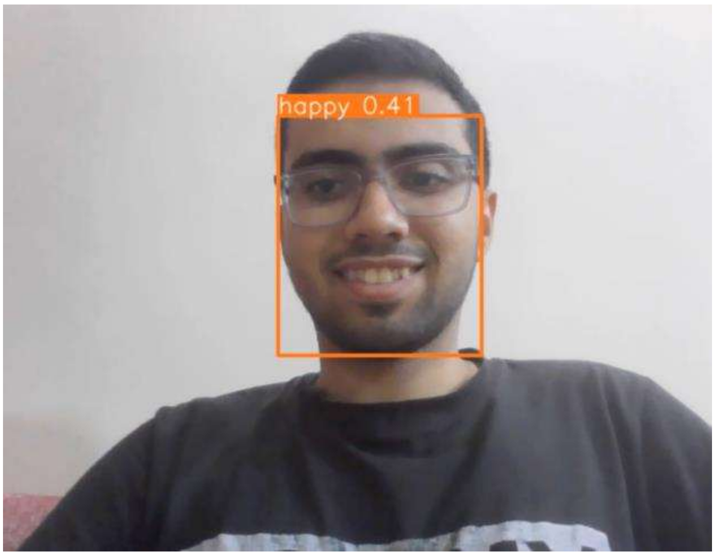
7. The video capture loop continues until the time exceeds a predefined `window_timeout` or the user presses 'q' to quit. Upon exiting the loop, it displays whether the sitting duration exceeds 45 seconds, determining customer satisfaction.

3. OUTPUT RESULT

recognition of sad faces

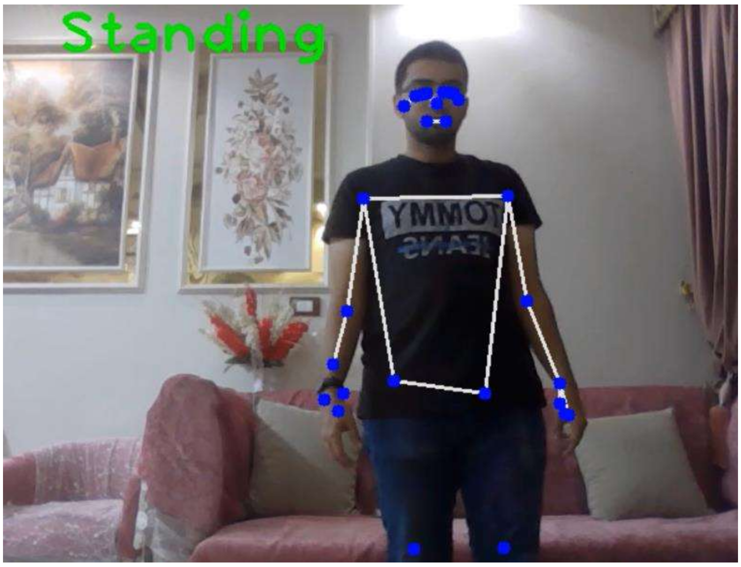
recognition of happy faces

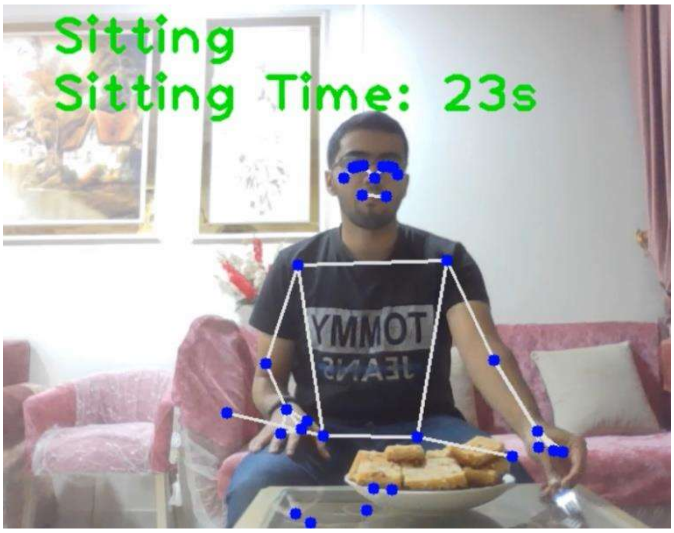


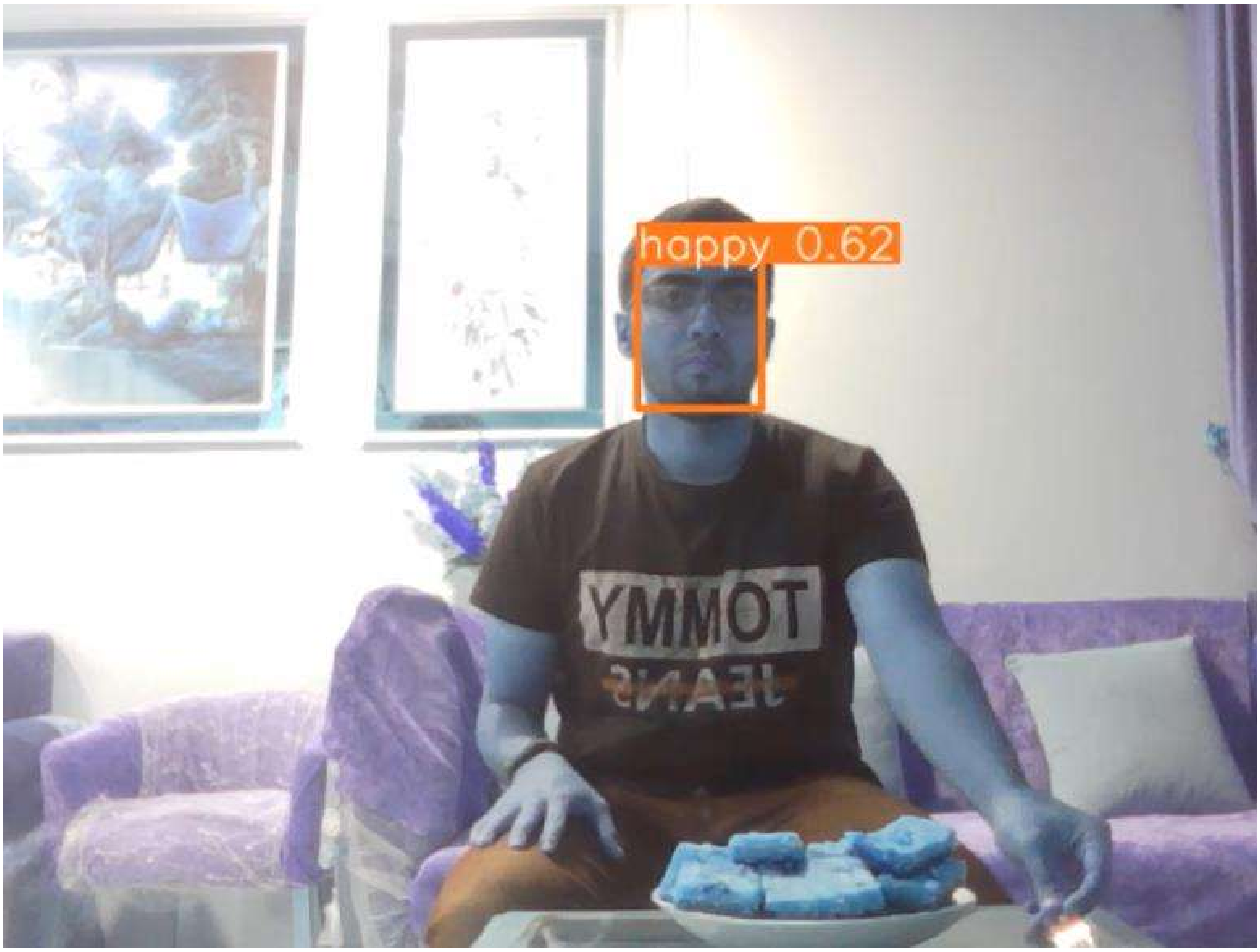


defined standing people.

defined standing people.







Recognition mood while eating

4. REFRENCESS

1. Zhang, L., & Tjondronegoro, D. (2011). Facial expression recognition using facial movement features. *IEEE Transactions on Affective Computing*, *2*(4), 219–229. https://doi.org/10.1109/t-affc.2011.13

2. Role of facial expressions in social interactions | philosophical ... (2015, December 12). https://royalsocietypublishing.org/doi/10.1098/rstb.2009.0142

3. B., E. (2023, September 11). *9 challenges restaurant face and how to solve them*. Orders.co. https://orders.co/blog/challenges-restaurant-face-and-how-to-solve-them/

4. The behaviour of consumer society in consuming food at ... - core. (2005, August 19). https://core.ac.uk/download/pdf/82203463.pdf

5. Farnsworth, B., Nguyen, N., Bortone, A., Seernani, D., Krosschell, K., Bülow, P., & Martin, M. (2023, January 6). *Facial expression analysis: The Complete Pocket Guide*. iMotions. https://imotions.com/blog/learning/best-practice/facial-expression-analysis/

6. Dye, T. (2022, August 25). *Review analysis: How to analyze customer & product reviews*. Thematic. https://getthematic.com/insights/review-analysis/#:~:text=What%20is%20review%20analysis%3F,for%20sentiment%20on%20desired%20features

7. team, Aic. (2023, November 5). The role of AI in content sentiment analysis for customer feedback. https://aicontentfy.com/en/blog/role-of-ai-in-content-sentiment-analysis-for-customer-feedback

8. Troyer, J. (2023, November 28). *Launch your own reputation management platform*. Reviewgrower. https://reviewgrower.com/get-customer-feedback-restaurant/#h-focus-groups