KARACHI INSTITUTE OF ECONOMICS & TECHNOLOGY

College Of Engineering

Software Engineering

Complex Engineering Activity

Artificial Intelligence & Neural Networks

Student Name:_____ **Student ID:** _____

CEP Statement	Design and implement a complex problem using artificial Intelligence or Machine Learning techniques to solve a real-life issue. .

PLO's	**PLO2 – Problem Analysis** **PLO3 – Design and Development** **PLO9 – Team Work or Individual**	Bloom's Taxonomy	**C4 – Analyze** **C3 – Apply** **A3 – Assume Responsibility**

COMPLEX ENGINEERING ACTIVITY

CPA	CLO's	Aspects of Assessments	Excellent (75-100%)	Average (50-75%)	Poor (<50 %)	Marks	
CEA-2 CEA-5	CLO2 PLO2	**Problem Analysis** Problem identification, analysis /literature review, resulting in meaningful conclusions	Completely identifies the problem in question through efficient analysis/produces near-to-exact results	Partially identifies the problem in question and with academic support produces the required results.	Lack of identification of the problem, needing more than par support to analyze the problem and production of results.	**Project Idea (15%)**	
						Proposal (15%)	
CEA-1 CEA-2 CEA-5	CLO3 PLO3	**Design & Development** Design/Develop solutions for CEA using AI and machine learning algorithms with Python programming language.	A complete solution / Explain necessary theories according to task description with great use of time and resource material.	The solution was complete but needed minor modifications/the student could have followed the specifications more closely.	The solution was complete but did not work, needed several modifications / did not make correct use of resource material or instructions.	**Project Progress (40%)**	
CEA-2 CEA-5	CLO7 PLO9	**Teamwork** Completion of project tasks with proper teamwork and contribution.	Proactively work with other team members to complete assigned tasks.	Worked well with a team but did not offer much positive feedback.	Very little, if any, contributions to the group and less contribution in the completion of overall lab tasks.	**Demo (15%)**	
						Report (15%)	
					Total Marks: 30		

Graded by Engineer: _____ **Date:** _____

Remarks: _____

Complex Engineering Activity

The semester project is designed in a way to able students to solve a complex engineering Activity. The following characteristics of Complex Engineering Activity are targeted in this semester's project of **Artificial Intelligence**:

CEA-1	Range of resources	**Diverse resources** (people, money, equipment, materials, information, and technologies
CEA-2	Level of interaction	Require resolution of significant problems arising from interactions between **wide-ranging** or **conflicting** technical, engineering, or other issues.
CEA-5	Familiarity	Can extend **beyond previous** experiences by applying **principles-based** approaches

Course	CPA Attributes	CLO	PLO (WA)	Bloom's Taxonomy
Artificial Intelligence & Neural Networks (Lab)	CEA-2 CEA-5	CLO2	PLO2	C4 **(Analyze)**
	CEA-1 CEA-2 CEA-5	CLO3	PLO3	C2 **(Apply)**
	CEA-2 CEA-5	CLO7	PLO9	A3 **(Assume Responsibility)**

➢ CEP Statement:

Design and implement a complex problem using Artificial Intelligence or Machine Learning techniques to solve a real-life issue.

➢ Design Constraints:

Select your project idea from the below topics

- **Neural Network**
  1. Artificial Neural Network **(ANN)**
  2. Convolutional Neural Network **(CNN)**
- **Machine Learning**
  1. Linear Regression
  2. K nearest Neighbor
  3. Decision Tree
- **YOLO v5 Model**

You can use built-in Libraries in your project which we studied in the lab i.e. **Tensor flow**, **Keras**, **Deep Face**, etc.

➢ Project Phases

The project is carried out in three phases

1. Problem Identification/Idea/Initial Study:

In the first phase, students are asked to bring the problems they intend to work on. Counseling is given to students in the lab and contact hours for finalizing their ideas and preparing a proposal. Students have to explore the problems/issues around them, which they can solve using programming constructs. If the problems brought by the students are irrelevant to the course or not feasible at this level, they are asked to bring up some other problem. Once ideas are finalized, constant counseling must be provided by the Course Instructor/Lab Engr.

2. Project Proposal:

In this phase, students have to explore the literature or existing solutions for their selected project idea. In this phase, students are also encouraged to have a detailed analysis of the problem to solve it in a better way. Each student's project is unique and may have many possible solutions as well as may be explored and developed in a different way. After discussion, students are asked to submit a proposal on one idea approved by the Instructor/Lab Engr. A sample of the Project Proposal is also attached to the **Google Classroom**

3. Project Demonstration

Every project is checked by running and observing the output. Students had the option of using any programming tool of Python to develop a project. Students have trio apply in-depth engineering knowledge to complete each project **(CEA1, CEA5**). During the initial study and formulation of the proposed solution, they focused on the detailed requirements **(CEA-2)**, and real-time constraints **(CEA-2)** and performed an in-depth analysis.

A complete assessment of each student is presented in the result. Sample project reports are also attached in the **Google Classroom**

Projects were evaluated on the following criteria:

- **Idea/Initial Study** (CEA-2) **15%**
- **Project Proposal** (CEA-2, CEA-5) 1**5%**
- **Project Progress** (CEA-1, CEA-2, CEA-5) **40%**
- **Demo/Report** (CEA-2, CEA-5) **30%**

➢ Summary:

The following are salient outcomes of the semester project in terms of **Complex Engineering Activity**:

- Brainstorming exercises forced them to explore the surrounding environment to sort out the problems to be solved using programming constraints.
- Problem formulation enhances their ability to gather real-time requirements and address conflicts/constraints.
- Design/Implementation gave them a chance to go through the in-depth engineering knowledge to solve the problem and analyze it in an effective way

GROUP MEMBERS:

1. Ali Siraj 14156
2. Mohammad Faqhi 12756
3. Hashir Ahmed 13948

Face Movement Tracking

1. Abstract:

Facial movement tracking has emerged as a crucial technology in enhancing human-computer interaction (HCI) by enabling more intuitive and natural interfaces. This project explores the development and application of facial movement tracking algorithms for real-time interaction with computing systems. Through computer vision techniques and hardware integration, the project aims to improve user experience and accessibility in various domains such as gaming, healthcare, and market research. The implementation of facial movement tracking offers promising opportunities for advancing HCI and fostering innovative ways of interaction between humans and machines.

2. Introduction:

In recent years, the field of human-computer interaction (HCI) has witnessed significant advancements driven by emerging technologies such as facial movement tracking. Traditional input devices like keyboards and mice have limitations in capturing the richness of human expression and communication. Facial movement tracking technology offers a compelling alternative by enabling computers to interpret and respond to subtle facial gestures and expressions in real-time. This project endeavors to explore the potential of facial movement tracking in revolutionizing HCI, with a focus on its applications across diverse sectors. By leveraging computer vision algorithms and sensor technologies, the project seeks to develop a robust framework for integrating facial movement tracking into interactive computing systems. The ultimate goal is to enhance user experience, accessibility, and engagement in various human-computer interaction scenarios.

3. Implementation:

Implementing a project on facial movement tracking for human-computer interaction involves several steps, including hardware setup, software development, data collection, training models, and testing. Here's a high-level overview of how you could approach the implementation:

3.1 Software Development:

- Choose a programming language and development environment suitable for computer vision and machine learning tasks (e.g., Python with libraries like OpenCV, TensorFlow, or PyTorch).

- Develop algorithms for facial feature detection, tracking, and gesture recognition using computer vision techniques such as Haar cascades, facial landmark detection, and deep learning.
- Implement real-time processing and analysis of facial movement data captured by the camera.

3.2 Data Collection:

- Gather a diverse dataset of facial movements and expressions to train and validate the facial gesture recognition model.
- Ensure the dataset includes a variety of facial expressions, gestures, and demographic characteristics to improve the model's robustness and generalization.

3.3 Training Models:

- Train machine learning or deep learning models to recognize and classify facial gestures based on the collected dataset.
- Fine-tune the models to improve accuracy and performance using techniques like data augmentation, transfer learning, and hyper parameter optimization.

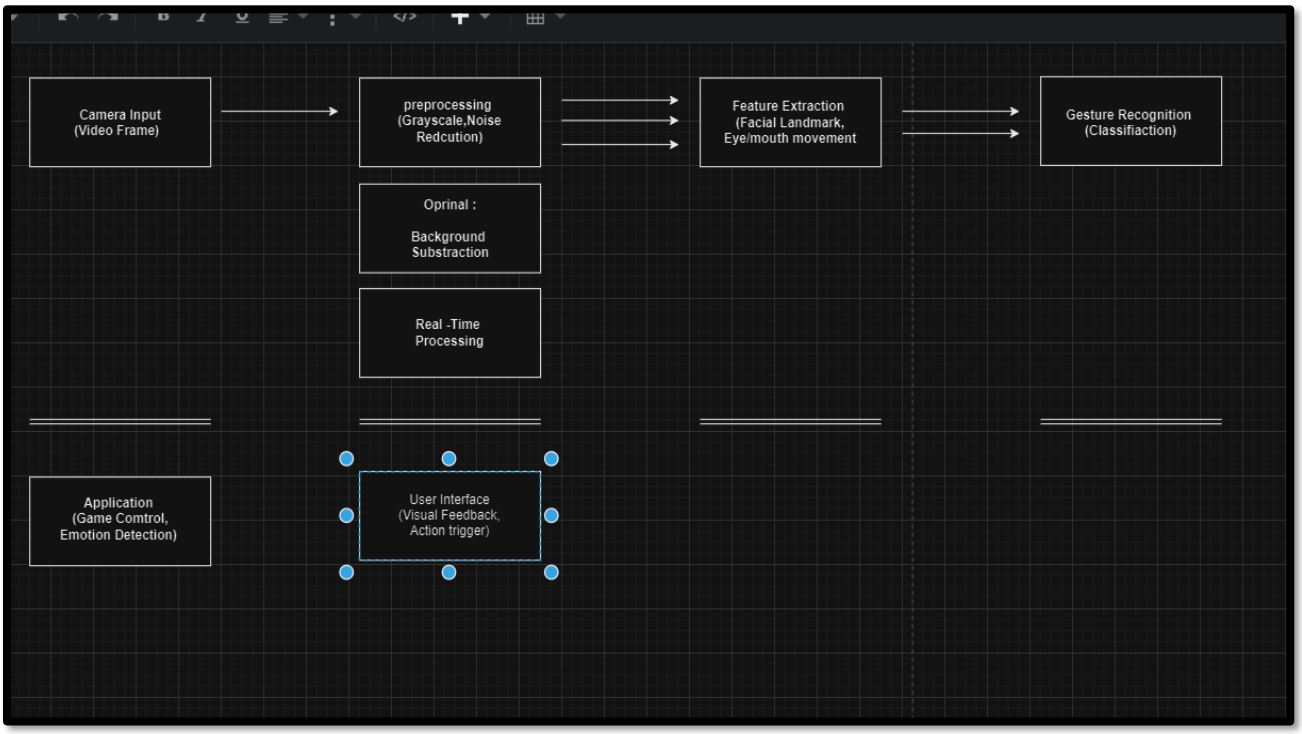
3.4 Testing and Evaluation:

- Conduct thorough testing and evaluation of the implemented system to assess its accuracy, responsiveness, and usability.
- Gather feedback from users through user studies, surveys, or usability testing sessions to identify strengths, weaknesses, and areas for improvement.
- Iterate on the design and implementation based on user feedback and performance metrics to refine the system further.

3.5 Deployment and Maintenance:

- Deploy the facial movement tracking system in real-world settings or applications where it can be used by end-users.
- Monitor the system's performance and reliability in production environments and address any issues or bugs that arise through regular maintenance and updates.

4 BLOCK DIAGRAM:



5 SOURCE CODE:

```
1
2  #GROUP NAME: Hashir Ahmed 13948
3  #            Ali Siraj 14156
4  #            M. Faqhi 12756
5  #BE-SOFTWARE ENGINEER BATCH 3
6  #AI LAB PROJECT:
7  #                        "Real -Time Face Movement Tracking"
8  #
9
10 import cv2
11 import keyboard
12 import time
13
14 color = {"blue":(255,0,0), "red":(0,0,255), "green":(0,255,0), "white":(255,255,255)}
15
16 # Method to detect nose
17 def detect_nose(img, faceCascade):
18
19      # convert image to gray-scale
20     gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
21
22     # detecting features in gray-scale image, returns coordinates, width and height of features
23     features = faceCascade.detectMultiScale(gray_img, 1.1, 8)
24     nose_cords = []
25     # drawing rectangle around the feature and labeling it
26     for (x, y, w, h) in features:
27 #         cv2.rectangle(img, (x,y), (x+w, y+h),  color['green'], 2)  #uncomment if you want to see face boundary
28         cv2.circle(img, ((2*x+w)//2,(2*y+h)//2), 10, color['green'], 2)
29         nose_cords = ((2*x+w)//2,(2*y+h)//2)
30     return img, nose_cords
31
32 def draw_controller(img, cords):
33     size = 40
34     x1 = cords[0] - size
35     y1 = cords[1] - size
```

```
    size = 40
    x1 = cords[0] - size
    y1 = cords[1] - size
    x2 = cords[0] + size
    y2 = cords[1] + size
    cv2.circle(img, cords,  size, color['blue'], 2)
    return [(x1,y1), (x2,y2)]

# tracking sudden changes
def get_movement(curr, prev, last_time_update, cmd):
    if not (len(curr) > 0 and len(prev) > 0):
        return last_time_update, cmd
    xc, yc = nose_cords
    tc = time.time()
    ox,oy, to = prev_cords
    diffx = xc - ox
    diffy = yc - oy
    thres_diff = 50

    thres_diff_t = 1
    if last_time_update + 0.4 > tc:
        return last_time_update, cmd

    if (abs(diffx)>thres_diff or abs(diffy) > thres_diff) and abs(tc-to)<thres_diff_t:
        if abs(diffx) > abs(diffy):
            if diffx > 0:
                cmd = "right"
            else:
                cmd = "left"
        else:
            if diffy > 0:
                cmd = "down"
            else:
                cmd = "up"
        print("Movement detected: ", cmd, "\n")
```

```
                cmd = "down"
            else:
                cmd = "up"
        print("Movement detected: ", cmd, "\n")
        keyboard.press_and_release(cmd)
        last_time_update = time.time()
    return last_time_update, cmd

# Loading classifiers
faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Capturing real time video stream.
video_capture = cv2.VideoCapture(0)

# get vcap property
width  = video_capture.get(3) # float
height = video_capture.get(4) # float
cmd = ""
prev_cords = (0,0,  time.time())
last_time_update = time.time()


while True:
    # Reading image from video stream
    _, img = video_capture.read()
    img = cv2.flip( img, 1 )

    # detect nose and draw
    img, nose_cords = detect_nose(img, faceCascade)
    cv2.putText(img, cmd, (10,50), cv2.FONT_HERSHEY_SIMPLEX, 1, color['red'], 1, cv2.LINE_AA)

    last_time_update, cmd = get_movement(nose_cords, prev_cords, last_time_update, cmd)

    x, y = (0,0) if len(nose_cords) == 0 else nose_cords
    prev_cords = (x,y,time.time())
```

```
    img = cv2.flip( img, 1 )

    # detect nose and draw
    img, nose_cords = detect_nose(img, faceCascade)
    cv2.putText(img, cmd, (10,50), cv2.FONT_HERSHEY_SIMPLEX, 1, color['red'], 1, cv2.LINE_AA)

    last_time_update, cmd = get_movement(nose_cords, prev_cords, last_time_update, cmd)

    x, y = (0,0) if len(nose_cords) == 0 else nose_cords
    prev_cords = (x,y,time.time())

    # Writing processed image in a new window
    cv2.imshow("face detection", img)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# releasing web-cam
video_capture.release()
# Destroying output window
cv2.destroyAllWindows()
```

6 Conclusion:

In conclusion, this project has demonstrated the significant potential of facial movement tracking technology in transforming human-computer interaction across multiple domains. By leveraging advancements in computer vision and sensor technologies, the project has developed a robust framework for real-time facial gesture recognition and interpretation. Through empirical testing and evaluation, the efficacy of the proposed system has been validated in enhancing user experience and accessibility in various interactive computing applications. However, challenges such as privacy concerns, hardware limitations, and algorithmic complexities remain pertinent and warrant further research and development. Despite these challenges, the promising outcomes of this project underscore the importance of facial movement tracking in shaping the future of HCI and fostering more natural and intuitive interactions between humans and machines.

7 References:

https://www.reddit.com/r/programming/comments/f0rc9e/i_made_a_face_tracking_nerf_turret_to_help/?rdt=55099

https://codeeverywhere.medium.com/face-detection-and-marking-with-python-and-opencv-beginners-b99551c4181