Chapter 12: AI-Based Activities (like Emoji Generator, Face Detection, etc.)

Introduction

Artificial Intelligence (AI) is not just theory — it is applied in exciting, real-world activities. In this chapter, we explore **hands-on AI-based projects** and applications that help students understand how AI works in practice. These activities are often implemented using **pre-trained AI models and tools** such as Teachable Machine, Google's AI Experiments, Python libraries, and more.

By working with examples like **Emoji Generators, Face Detection Systems, and Pose Estimation**, students learn key AI concepts such as **classification, object detection, image recognition**, and **data training**. These fun and interactive exercises form a bridge between classroom learning and real-world application.

12.1 Emoji Generator

What is it?

An **Emoji Generator** is an AI application that maps human facial expressions to corresponding emojis using a **trained image classification model**.

Concepts Involved

- **Image Classification**: Using AI to classify different images of facial expressions (e.g., happy, sad, angry).
- **Data Collection**: Capturing a dataset of facial expressions via webcam or image upload.
- **Model Training**: Using platforms like **Teachable Machine** to train the model.
- **Real-Time Prediction**: After training, the model predicts emotion and displays matching emoji.

Steps to Build

1. Open Teachable Machine (https://teachablemachine.withgoogle.com/).
2. Choose the **Image Project**.
3. Create different **classes** (e.g., Happy, Sad, Surprised).
4. Record samples for each class using webcam.
5. Train the model with the collected data.
6. Export or test the model directly.
7. Integrate the model with HTML/JS or Python to display the corresponding emoji.

Educational Outcomes

- Understanding training data and bias.
- Exploring model accuracy and retraining.
- Realizing limitations of AI in real-world conditions.

12.2 Face Detection

What is it?

Face Detection is an AI task that identifies and locates human faces in digital images or video streams. Unlike face **recognition**, it does not identify the person, just the presence of a face.

Concepts Involved

- **Object Detection**: Recognizing specific objects (faces) in an image.
- **OpenCV Library**: Popular library used for face detection in Python.
- **Haar Cascade Classifier**: Pre-trained model for detecting faces.

Steps to Build (Python-based)

1. Install OpenCV: `pip install opencv-python`
2. Import required modules:

   ```
   import cv2
   ```

3. Load the Haar Cascade Classifier:

   ```
   face_cascade =
   cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
   ```

4. Read from webcam and detect faces:

   ```
   cap = cv2.VideoCapture(0)
   while True:
       ret, frame = cap.read()
       gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
       faces = face_cascade.detectMultiScale(gray, 1.1, 4)
       for (x, y, w, h) in faces:
           cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
       cv2.imshow('Face Detection', frame)
       if cv2.waitKey(1) == ord('q'):
           break
   cap.release()
   cv2.destroyAllWindows()
   ```

Educational Outcomes

- Understanding difference between detection and recognition.

- Learning real-time processing using Python and OpenCV.
- Exploring ethical aspects (privacy, surveillance).

12.3 Pose Estimation

What is it?

Pose estimation is the technique of detecting human posture and key body points from images or video using AI.

Concepts Involved

- **Keypoint Detection**: Identifying parts like head, shoulders, arms, knees.
- **Pre-trained Models**: Like PoseNet, BlazePose.
- **Computer Vision**: AI's ability to extract human body posture from visuals.

Tools

- **TensorFlow.js** + PoseNet in the browser.
- **MediaPipe** (by Google) for Python-based solutions.

Steps (PoseNet JS Example)

1. Load PoseNet via TensorFlow.js in an HTML file.
2. Capture webcam input.
3. Run PoseNet on frames.
4. Display keypoints and connect them visually.

Applications

- Fitness apps (form correction).
- Dance and gesture-based games.
- Health monitoring.

12.4 Teachable Machine Experiments

What is Teachable Machine?

A browser-based tool by Google that allows students to **train custom machine learning models** in image, sound, and pose without coding.

Experiments to Try

- **Image Classification** (similar to emoji generator).
- **Audio Classification** (e.g., recognize claps, whistles).
- **Pose Classification** (e.g., yoga poses).

Why Use It?

- Beginner-friendly.
- Fast training with high interactivity.
- Can export models to TensorFlow for advanced use.

12.5 AI with Scratch and Blockly

Why Use Visual Tools?

For beginners, drag-and-drop platforms like **Scratch + ML extensions** or **Blockly** simplify AI logic building.

Example Activities

- Train a model to recognize hand gestures and use them to control a Scratch sprite.
- Make a robot that reacts differently to different sounds.

12.6 Ethical Considerations

Even while performing fun activities, students should reflect on:

- **Bias in models**: Models might perform poorly on underrepresented data.
- **Data privacy**: Images and personal data must be handled carefully.
- **Overfitting**: Small training data can lead to poor generalization.

Chapter Summary

Key Concept	Description
Emoji Generator	Maps facial expressions to emojis using classification models.
Face Detection	Identifies face regions using pre-trained models like Haar Cascades.
Pose Estimation	Detects body keypoints for applications in fitness, games, etc.
Teachable Machine	A no-code platform for training image, audio, and pose models.
Visual Programming	Simplifies AI logic building for beginners using Scratch or Blockly.
Ethical Aspects	Understanding bias, privacy, and responsible use of AI tools.

✓ Key Takeaways

- You don't need to be a coder to build and understand AI applications.
- Activities make abstract AI concepts real and tangible.
- With simple tools, students can build AI projects that reflect the **power and responsibility** of modern technology.