**torch and torchvision**:

* **Definition**: PyTorch is an open-source machine learning framework that facilitates building and training deep learning models. Torchvision is a PyTorch library that provides computer vision utilities and datasets.
* **Purpose**: Used for defining and training deep neural networks such as InceptionResnetV1, handling tensors, and performing GPU-accelerated computations when available.

**facenet\_pytorch (MTCNN):**

* **Definition**: A PyTorch-based implementation of the Multi-task Cascaded Convolutional Networks (MTCNN) for face detection and alignment.
* **Purpose**: Enables accurate detection of faces within images and video frames, providing pre-trained models and utilities for facial feature extraction.

**numpy**:

* **Definition**: Fundamental package for numerical computing in Python, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions.
* **Purpose**: Used extensively for data manipulation, array operations, and converting tensors to numpy arrays for visualization and preprocessing tasks.

**PIL (Pillow)**:

* **Definition**: Python Imaging Library (PIL) fork known as Pillow, which adds image processing capabilities to Python.
* **Purpose**: Handles image loading, resizing, and conversion between different image formats, essential for preprocessing input images and generating visual outputs.

**cv2 (OpenCV)**:

* **Definition**: Open Source Computer Vision Library (OpenCV) is a library of programming functions mainly aimed at real-time computer vision.
* **Purpose**: Used for video capturing, frame extraction, and image processing tasks such as image filtering, transformations, and blending, crucial for handling video inputs in the detection pipeline.

**pytorch\_grad\_cam**:

* **Definition**: A PyTorch library for generating class activation maps (CAM) and gradient-weighted class activation maps (Grad-CAM) for explaining deep learning model decisions.
* **Purpose**: Utilized to visualize where the model focuses its attention when making predictions, enhancing interpretability and understanding of model behavior.

**gradio**:

* **Definition**: An easy-to-use Python library for creating customizable UI components around machine learning models.

**Purpose**: Facilitates model deployment with user-friendly interfaces for uploading