Project Title: Utilizing Phone Camera for Computer Vision in Python

Problem Statement:

Many devices come equipped with high-resolution cameras, including smartphones. However, accessing and utilizing these cameras for computer vision tasks, such as object detection, pose estimation, or facial recognition, from a Python environment presents challenges. This project aims to bridge this gap by developing a system that enables users to leverage the camera on their smartphones for computer vision tasks seamlessly.

Proposed System/Solution:

The proposed solution involves creating a Python-based application that communicates with a mobile app installed on the user's smartphone. The mobile app streams live video from the phone's camera to the Python application running on a computer. The Python application then performs computer vision tasks on the streamed video using libraries like OpenCV and TensorFlow.

System Development Approach:

Requirement Analysis: Understand the functionalities required from both the mobile app and Python application.

Mobile App Development: Develop a mobile app compatible with Android and iOS platforms for streaming live video to the Python application.

Python Application Development: Create a Python application using OpenCV and TensorFlow for performing computer vision tasks on the streamed video.

Integration: Integrate the mobile app with the Python application to establish a seamless communication channel.

Testing: Conduct rigorous testing to ensure the reliability and accuracy of computer vision tasks performed using the phone camera.

Deployment: Deploy the system on both the mobile and computer platforms for end-user access.

Algorithm:

Initialize the mobile app to stream live video.

Establish a connection between the mobile app and the Python application.

Continuously capture frames from the streamed video.

Apply computer vision algorithms such as object detection or facial recognition on the captured frames.

Display the processed frames with detected objects or recognized faces to the user.

Deployment:

The system can be deployed by distributing the mobile app through respective app stores (Google Play Store for Android and Apple App Store for iOS). The Python application can be packaged as an executable or distributed as a Python package for installation on the user's computer.

SOURCE CODE :

import cv2

import numpy as np

url = "Your IP Address/video"

cp = cv2.VideoCapture(url)

while(True):

camera, frame = cap.read()

if frame is not None:

cv2.imshow("Frame", frame)

q = cv2.waitKey(1)

if q==ord("q"):

break

cv2.destroyAllWindows()

Result:

The deployed system allows users to leverage the camera on their smartphones for various computer vision tasks, providing real-time feedback on object detection, facial recognition, or other desired functionalities.

Conclusion:

Utilizing phone cameras for computer vision tasks in Python opens up a wide range of possibilities for developers and users. This project demonstrates the feasibility and practicality of such a system, offering flexibility and convenience in capturing and processing visual data.

References:

OpenCV documentation: https://docs.opencv.org/

TensorFlow documentation: https://www.tensorflow.org/api\_docs

Mobile app development resources for Android and iOS platforms.