Name:MADHUMITHA.P
REG NO:711721243055
DEPARTMENT:B.TECH Artifical Intelligence And DataScience

PROJECT TITLE

REAL TIME OBJECT DETECTION USING CNN

AGENDA

1. Problem Statement
2. Project Overview
3. End Users
4. Our Solution and Proposition
5. Key Features
6. Modelling Approach
7. Results and Evaluation
8. Conclusion

PROBLEM STATEMENT

-Developing an object detection system capable of accurately identifying multiple objects within images or video frames.

- Addressing challenges related to occlusions, variations in lighting conditions, and complex backgrounds to enhance detection accuracy.

- Optimizing the object detection algorithm for real-time performance to meet the demands of dynamic environments such as autonomous vehicles or live surveillance feeds.

PROJECT OVERVIEW

- This project implements a real-time object detection system using the MobileNet SSD model, allowing for the detection of various objects in a live video stream.

-By leveraging computer vision techniques and deep learning, the system can process video frames, infer object classes, and display bounding boxes with corresponding labels, providing a real-time visual representation of detected objects.

WHO ARE THE END USERS?

-Healthcare professionals and medical facilities
-Manufacturing and logistics industries
-Municipalities and urban planners
-Entertainment and gaming developers
-Environmental agencies and researchers

YOUR SOLUTION AND ITS VALUE PROPOSITION

1. **Real-Time Object Detection**: The script employs the MobileNet SSD model to perform real-time object detection from a live video stream, enabling instant analysis and response to the detected objects.

2. **Flexible Configuration and Customization**: Users have the ability to specify the paths to the model files and set the minimum confidence threshold according to their specific needs, offering adaptability and customization.

3. **Efficient Processing and High Performance**: Leveraging optimized algorithms and pre-trained models, the script ensures efficient processing and high-performance object detection, even on devices with limited computational resources.

4. **User-Friendly Interaction and Control**: With interactive features allowing users to exit the program at any time with a key press, the script provides a user-friendly interface, enhancing usability and control.

5. **Modular Design for Scalability and Maintainability**: The script's modular design separates different functionalities into logical units, facilitating scalability, maintainability, and easy integration into existing systems, while also allowing for future updates and enhancements.

THE WOW IN YOUR SOLUTION

-Real time detection
-High accuracy in detescting a object
-potential for integration with existing surveillance systems

MODELLING

1. **Video Stream Entity**:
   - **Attributes**:
     - Source (src): An integer representing the source of the video stream, typically a webcam.
     - Running Status (running): A boolean indicating whether the video stream is currently running.
   - **Frames**:
     - Frame Width (frame_width): An integer representing the width of the video frames.
     - Frame Height (frame_height): An integer representing the height of the video frames.

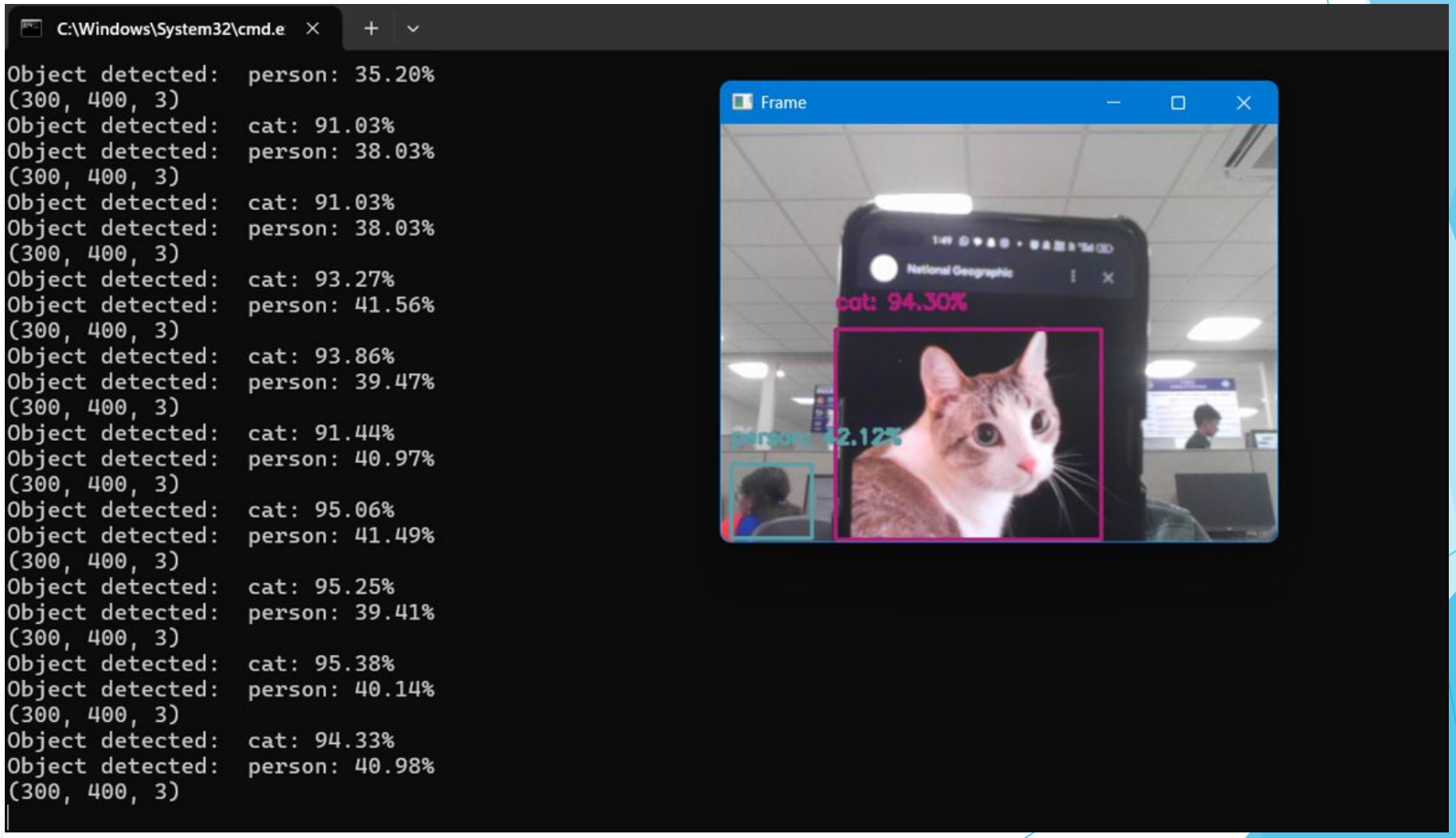
2. **Model Entity**:
   - **Attributes**:
     - Prototxt Path (prototxt_path): A string representing the path to the Caffe 'deploy' prototxt file.
     - Model Path (model_path): A string representing the path to the Caffe pre-trained model.
   - **Configuration**:
     - Confidence Threshold (confidence_threshold): A float representing the minimum probability to filter weak predictions.

3. **Object Detection Entity**:
   - **Attributes**:
     - Classes (classes): A list containing the class labels for object detection.
     - Colors (colors): A list containing random colors assigned to each class label.
   - **Detection Results**:
     - Predictions (predictions): An array containing the predictions generated by the model.
     - Detected Objects (detected_objects): A list containing the detected objects along with their labels and confidence scores.

4. **FPS Counter Entity**:
   - **Attributes**:
     - Started Status (started): A boolean indicating whether the FPS counter has been started.
     - Start Time (start_time): A timestamp representing the start time of FPS measurement.
   - **Performance Metrics**:
     - Elapsed Time (elapsed_time): A float representing the elapsed time between the start and end times.
     - Frames per Second (fps): A float representing the approximate frames per second of the video stream processing.

5. **User Interaction Entity**:
   - **Attributes**:
     - Key Pressed (key_pressed): A character representing the key pressed by the user to interact with the system.
   - **System Control**:
     - Control Flow (control_flow): Interaction with the system to trigger actions such as termination or pausing.

RESULTS


C:\Windows\System32\cmd.e
Object detected: person: 35.20%
(300, 400, 3)
Object detected: cat: 91.03%
Object detected: person: 38.03%
(300, 400, 3)
Object detected: cat: 91.03%
Object detected: person: 38.03%
(300, 400, 3)
Object detected: cat: 93.27%
Object detected: person: 41.56%
(300, 400, 3)
Object detected: cat: 93.86%
Object detected: person: 39.47%
(300, 400, 3)
Object detected: cat: 91.44%
Object detected: person: 40.97%
(300, 400, 3)
Object detected: cat: 95.06%
Object detected: person: 41.49%
(300, 400, 3)
Object detected: cat: 95.25%
Object detected: person: 39.41%
(300, 400, 3)
Object detected: cat: 95.38%
Object detected: person: 40.14%
(300, 400, 3)
Object detected: cat: 94.33%
Object detected: person: 40.98%
(300, 400, 3)
Frame
National Geographic
cat: 94.30%

Conclusion

In conclusion, the provided Python script effectively demonstrates real-time object detection using the MobileNet SSD model and OpenCV. By leveraging the efficiency of MobileNet SSD and the robustness of OpenCV, the script accurately identifies objects in live video streams. Its straightforward implementation, combined with user-friendly interaction for quitting the application, makes it a valuable tool for various computer vision applications. With the ability to process frames in real-time and provide feedback on elapsed time and FPS, this project offers a practical solution for object detection tasks. Overall, it showcases the power and versatility of deep learning and computer vision techniques in real-world scenarios.