**IMAGE RECOGNITION WITH IBM CLOUD VISUAL RECOGNITION:**

**Project explanation:**

Given IBM Watson Visual Recognition is discontinued. Existing instances are supported until 1 December 2021, but as of 7 January 2021, you can't create instances. Any instance that exists on 1 December 2021 will be deleted.so we use the Openvino, OpenCV and TensorFlow Application using Python language. Few information about these three application are,

**OPENVINO:**

**1.Optimized Inference Engine:**

OpenVINO provides an optimized inference engine that allows you to run deep learning models efficiently on a variety of hardware platforms, including CPUs, GPUs, FPGAs, and VPUs (Vision Processing Units). It uses hardware acceleration and model quantization techniques to speed up the inference process, making it suitable for real-time and edge device applications.

**2.Cross-Platform Compatibility:**

OpenVINO is designed to be a cross-platform and cross-framework toolkit. It supports various deep learning frameworks like TensorFlow, Caffe, and ONNX. This flexibility allows you to convert and optimize models trained in different frameworks and deploy them on a wide range of target devices without major modifications. OpenVINO also provides APIs for various programming languages, including Python, C++, and Java.

**3.Pre-Trained Models and Model Zoo:**

OpenVINO includes a Model Zoo with a collection of pre-trained models for various computer vision and image recognition tasks. These models cover a wide range of use cases, including object detection, image classification, facial recognition, and more. Using pre-trained models from the Model Zoo can save development time and resources, making it easier to get started with image recognition projects.

**OPENCV:**

**A) Robust and Versatile Library:**

OpenCV is a comprehensive and widely used computer vision library that provides a rich set of tools and functions for image recognition. It supports a wide range of image processing tasks, such as feature extraction, object detection, object tracking, and more. OpenCV's versatility makes it suitable for various image recognition applications, including facial recognition, object detection, OCR (Optical Character Recognition), and image segmentation.

**B) Integration with Machine Learning and Deep Learning:**

OpenCV integrates well with machine learning and deep learning frameworks, enabling developers to use pre-trained models or train their own models for image recognition. OpenCV's DNN (Deep Neural Network) module supports popular deep learning frameworks like TensorFlow, Caffe, and PyTorch, allowing you to employ state-of-the-art neural networks for tasks like object detection and image classification. This integration enables advanced image recognition techniques that leverage deep learning models.

**TENSORFLOW:**

1. **Flexibility and Customization:**

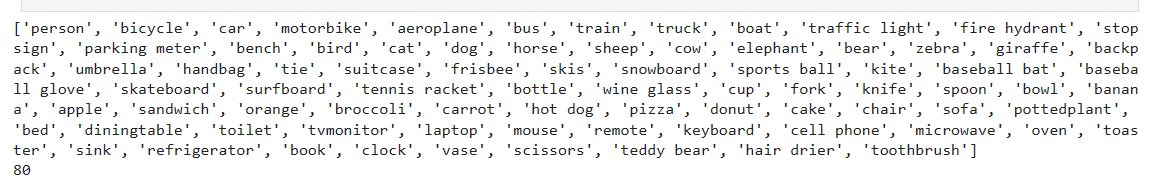
TensorFlow offers a high degree of flexibility and customization for image recognition tasks. You can design and train custom deep learning models for image classification, object detection, and more using TensorFlow's extensive set of tools and APIs. This flexibility allows you to adapt models to specific requirements, such as recognizing unique objects or handling particular types of images.

1. **Pre-trained Models:**

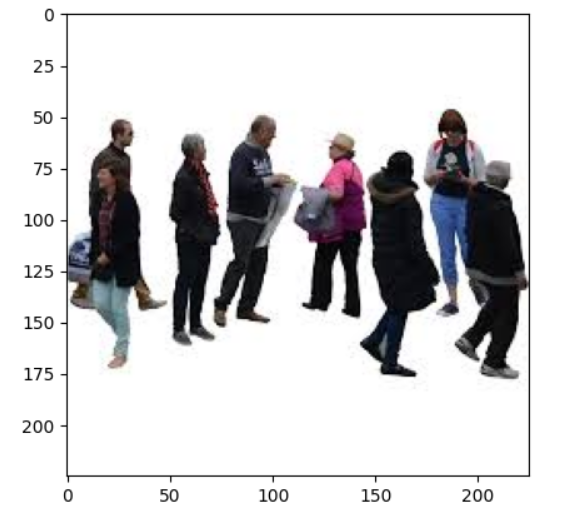
TensorFlow provides a collection of pre-trained models for image recognition tasks. These models are trained on large datasets and can be fine-tuned for specific applications. Pre-trained models like Inception, ResNet, and MobileNet are readily available and serve as a valuable starting point for image recognition projects. They can significantly reduce the effort and data required to build effective recognition systems.

**OUTPUT:**

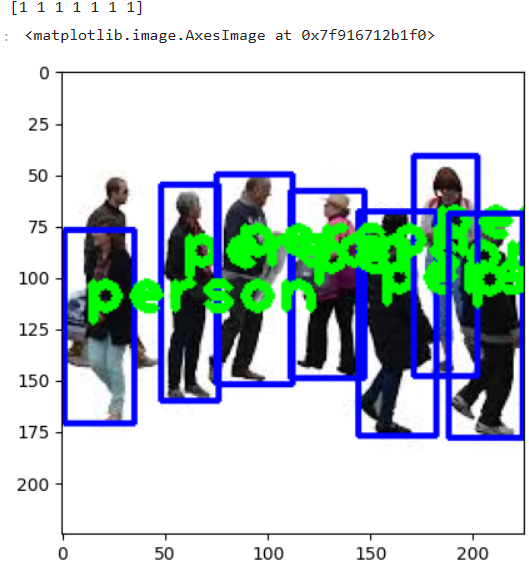
There some output using openvino and opencv and tensorFlow used for the image recognition in the jupyter notebook ,some screenshots are given below, And also use the detect function to identify the given image but these output will display the number of the image or object present in the give image.

This picture will assign the number for each object. For example, person is 1, bicycle is 2 ………………toothbrush is 80.

**INPUT 1:**



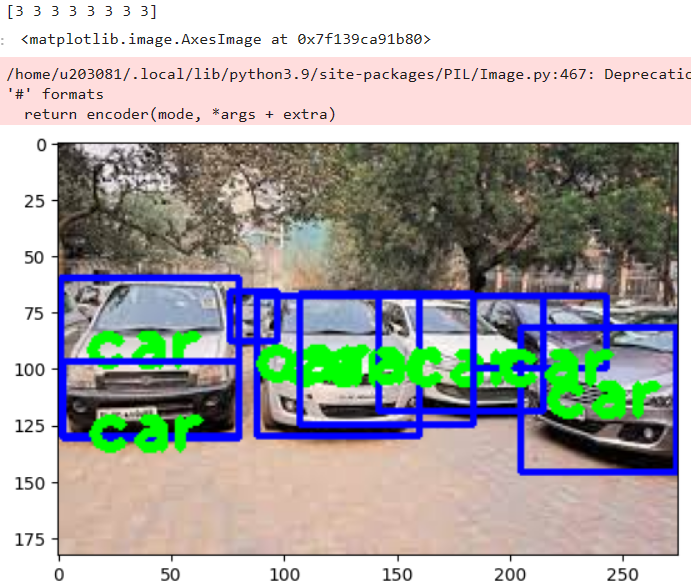
**OUTPUT 1:**

****

**INPUT 2:**

****

**OUTPUT 2:**

****