Object Presence Notification System

By:-
Souvik Mandal(160002056)
T. Venkat Nikhil(160001058)

Introduction:

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance. In our problem we use object detection and track the object to notify about suspicious.

Problem Formulation:

Problem Statement: To develop an object presence notification (time-based) algorithm detect certain objects on real-time video stream and generate a notification if they are found present for more than specified time.

Model: R-CNN(regional convolutional neural network)

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually refer to fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer.

In R-CNN the CNN is forced to focus on a single region at a time because that way interference is minimized because it is expected that only a single object of interest will dominate in a given region. The regions in the R-CNN are detected by selective search algorithm followed by resizing so that the regions are of equal size before they are fed to a CNN for classification and bounding box regression.

Tensorflow_used model: ssd_mobilenet_v1_coco

Dataset Used: Common Objects in Context dataset.

Link: http://cocodataset.org/#home

Training: 80% of the data images.

Testing: Remaining 20% of the images

Loss Function: Cross-entropy

We would seek a set of model weights that minimize the difference between the model's predicted probability distribution given the dataset and the distribution of probabilities in the training dataset. This is called the cross-entropy. In this problem we have used weighted sigmoid function as a probability distribution function.

Installation Guide: (For Windows, for python 3.7)

pip install tensorflow
pip install opencv-python
pip install pillow
pip install lxml
pip install jupyter
pip install matplotlib
git clone https://github.com/tensorflow/models
From https://github.com/protocolbuffers/protobuf/releases download protobuf 3.4.0
On models/research run following command
protoc object_detection/protos/*.proto --python_out=.
If there is a error saying not recognized command add protoc.exe to system variables
Location of protoc.exe "protoc-3.4.0-win32\bin"
Now copy every files in files1 given in zip file to "models> research> object_detection" folder
Replace models/research/object_detection/utils/visualization_utils.py with visualization_utils.py given in zip file

Execution Guide:

On models/research/object_detection folder run python test2.py

Files Description:

models> research> object_detection > test2.py (Main File, Object Detection and classifying into suspicious objects)

models> research> object_detection > match.py (Object Matching)

models> research> object_detection > tmp (Saving detected objects)

models> research> object_detection > tmp1 (Saving suspicious images)

models> research> object_detection > tmp1 > objects (Saving suspicious objects)

models> research> object_detection > tmp1 > persons (Saving suspicious people)

models> research> object_detection > samples > configs> ssd_mobilenet_v1_coco.config (Configuration file of used model ssd_mobilenet_v1_coco)

Object_detection> models> research> object_detection > retrain.py : Used to add additional class in pre-trained model.
For more details : https://www.tensorflow.org/hub/tutorials/image_retraining

Results:

Object Detection:

Example: Bag.

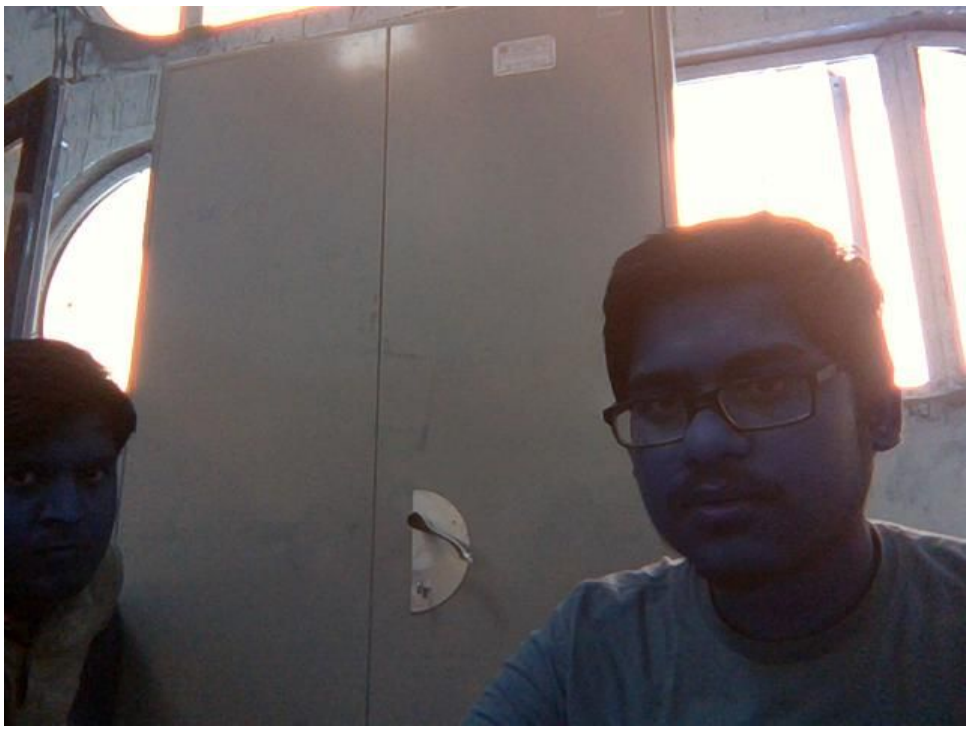
Original Image:(Taken from Video Input)

Detected Object:(When present for specified amount of time)

Person Detection:

Example:Persons

Original Image:(From Video Input)



Detected People:(When present for specified amount of time)

Conclusion:

Whenever an object is present for at least 60 frames the object image is captured and saved into their respective folders. The model notifies the presence of a suspicious object/person in real time. A new a class can added to the existing model according to the problem. So, this model can be used for efficient crime detection, crowd control, etc.

References:

https://www.tensorflow.org/hub/tutorials/image_retraining
https://www.tensorflow.org/hub/installation
https://github.com/tensorflow/models
https://github.com/protocolbuffers/protobuf
https://docs.opencv.org/4.1.0/