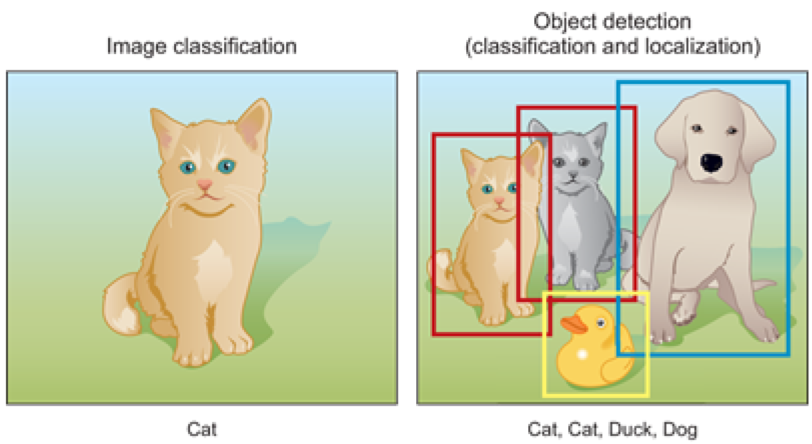
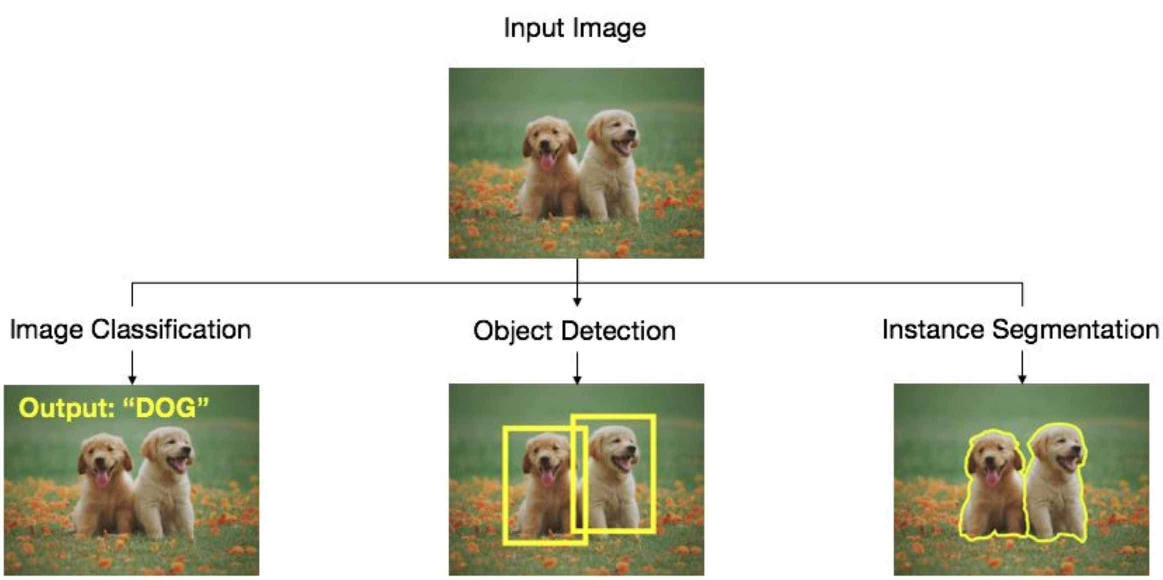
Viren Velacheri
EID: vv6898
Slip Days: 0

Assignment 5

Part 1

1. Image classification is where based on a given image you classify it as belonging to a certain class. Let's say you are given an image that contains a dog and you hence classify it as belonging to the dog class. In object detection though, you not only take into account the class that the image belongs to but also try and predict/find the location of the object in the image as well. In the example with an image that contains a dog, not only would the class be identified but the location of the dog object itself would be found as well. Both object detection and image classification depend on image segmentation or the idea of seeing the different components that the image consists of. Overall image classification helps classify what is in an image, while object detection actually specifies the location of multiple objects in an image. Object detection is basically image classification with localization. Couple of examples below:

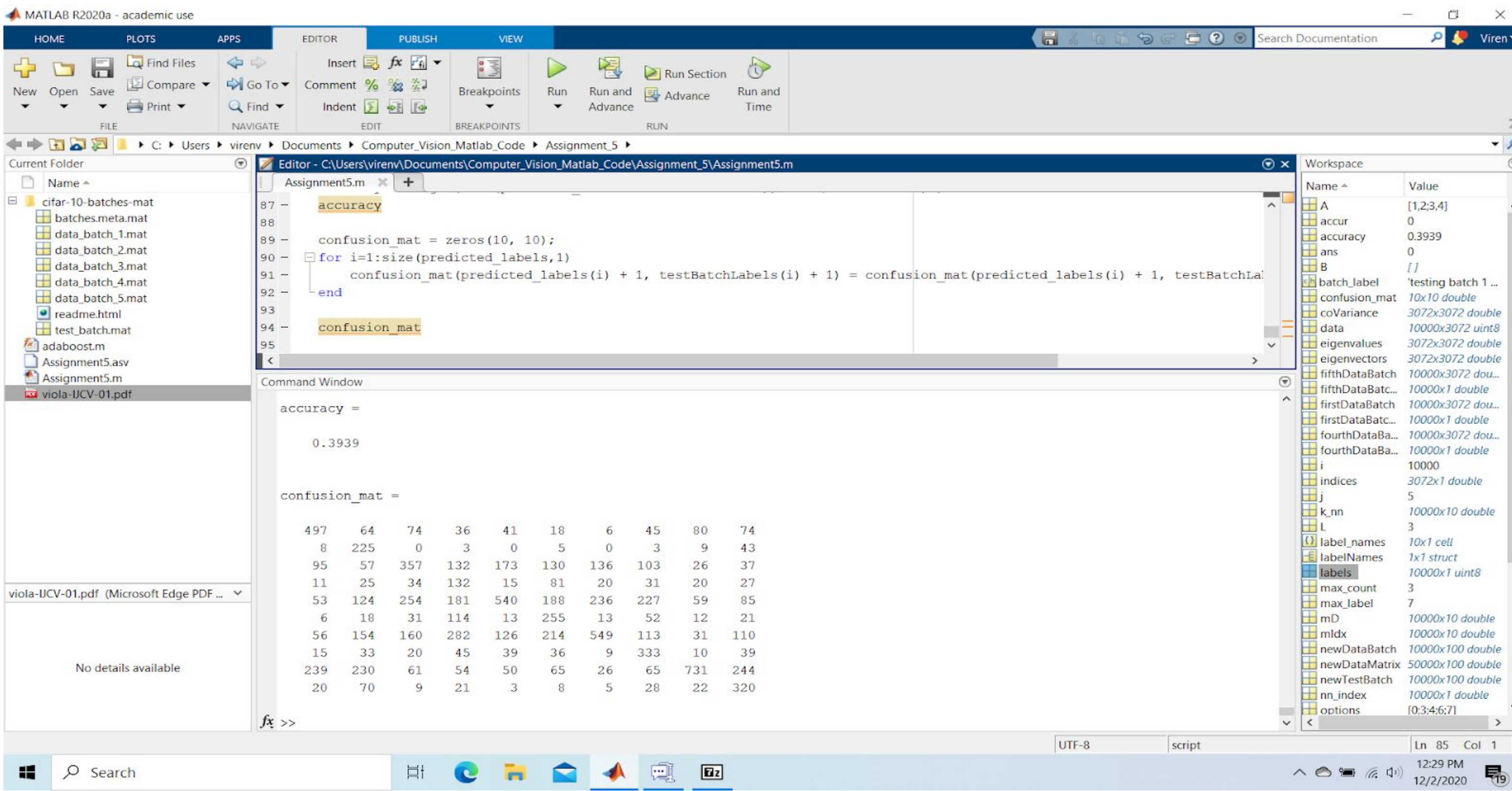




2. Semantic Segmentation is basically the process of assigning a label to every pixel in an image. This obviously is the opposite of image classification where the overall image is given a label. Semantic Segmentation also treats multiple objects of the same class as a single entity. On the other hand, instance segmentation treats multiple objects of the same class as distinct individual objects/instances. Due to this, instance segmentation is usually harder than semantic segmentation. Object detection is used for identifying the location of one or more objects in an image and having bounding boxes drawn around them as well as classifying these respective objects. Basically, all three (semantic segmentation, instance segmentation, object detection) all utilize image classification. The difference between them though is the way they treat the objects in an image. From least in depth to most in depth, it goes in the order of object detection, semantic segmentation, and instance segmentation.
3. One neural network design for the task of semantic segmentation could be to do R-CNN or basically region based semantic segmentation. To do this, you would extract the free-form regions from the input image and describe them and then do region-based classification. From there, at test time, these region based predictions would then be transformed into pixel predictions by labeling a pixel according to the highest scoring region that contained it. I guess to be more specific, in R-CNN the semantic segmentation would be done based on the object detection results. A selective search would have to be used to get a large number of object proposals and then compute the CNN features for each of them. The regions would then be classified based on class-specific linear SVMs. Another possible neural network design would be to use a Fully Convolutional Network that is based on semantic segmentation. The main thing would be to make the classic CNN take as input these various sized images. The restriction of these CNNs to accept and produce the labels for only the specific sized inputs comes from the fully connected layers which are fixed. Here though FCNs only have convolutional and pooling layers which enable them to make predictions on any sized input.
4. As far as from a machine learning perspective, they share many techniques for visual recognition. They often appear as sub modules for networks used in visual recognition. They also provided training data for supervised neural networks. Basically the applications of generative models in solving computer vision tasks/problems is that they generate new data, particularly when there is not much training data available initially. Some applications of generative models include image to image translations like converting day photos into night ones, rendering images, etc.
5. Convolution and Deconvolution neural networks use different layers. Convolution neural networks use convolution layers where each layer basically consists of independent filters that are used to convolve independently and produce output for the next layer. You are basically extracting the useful features from input at each layer. In deconvolution neural networks that layers are basically transposed convolution and perform the inverse operation of convolution. Here, an attempt is made to add useful features to an image to upscale it. It has these learnable features which are learnt through the use of backpropagation.

<u>Part 2</u>

1. K-nearest Neighbor Classifier Results:



As seen in displayed results above, accuracy was 39.39% and the confusion matrix is displayed above as well.

Note: Didn’t implement adaboosting or any of the following algorithms after K Nearest Neighbor Classifier as I had no more slip days.