1. [10 pts] Currently standard object detection evaluation penalizes approaches in the same way for confusing an object category with a (1) semantically related or (2) semantically unrelated category. How would you modify the mAP computation to account for semantic similarity? Can you imagine other scenarios where you might want to optimize a variant of mAP that doesn't exactly look like the mAP you computed above? What would this scenario be, and how might you modify the mAP computation to better reflect your desired outcome?

I assume semantically means when 2 categories are related. like bat and ball vs bat nd truck. I believe these replation would be captured in the last layer before the softmax. If we can get the 2nd last layer outputs and do a similarity match between categories and example which identifies original category, we can capture this relation. Same thing can be achieved by taking cosine similarity of word2vec of object and ground truth. Now this similarity could be used instead of TP. If it’s a TP, the word2vec similarity will be 1. If it is a FP, similarity can be between -1 to 1. This we can threshold it to 0 to 1. Similarly FP can be 1-similarity thresholded to range 0 to 1.

After every example

If IoU>0.5

sumSimilarity+=Similarity(detection, ground truth) sumDissimilarity+=1-Similarity(detection, ground truth)

Final formula:

(sumSimilarity)/(sumSimilarity +sumDissimilarity)

So when it’s a TP:similarity=1, dissimilarity=0

And when it is FP: similarity=a, dissimilarity=1-a where a is between 0 to 1.

One thing that IoU does not consider is classes which have been predicted right but have IoU less than 0.5.

To solve this I can say

It;s a FP when class doesn’t match ground truth when area don’t overlap

It’s a TP\* IoU if class match ground truth and they have overlapping area.

So

sumTIoU+=IoU(class match ground truth and they have overlapping area)

sumFIoU+=1( when class and ground truth area don’t overlap or classes doesn’t match but overlap)

Final formula:

(sumTIoU)/(sumTIoU +sumFIoU)

This will give some weight to at least correctly classify and give indication that this model is better than one which completely doesn’t overlap.

1. [10 pts] There are a variety of approaches to how we can model video with neural networks. Describe two approaches that you can construct based on the content presented in class. For each method, describe some possible applications that this method would be appropriate/advantageous for, and some applications where this method might achieve poor results.

**Method 1: LSTMS**

Pros: captures temporal data. Can be used for any size sequences( ong/short videos)

Cons: It is slow.

Applications: When the data is coming for variable length and length is generally very long it is excellent choice like 10 mins or more video. For data which is just few seconds(5 secs) it is not that good choice.

**Method2 : 3D Convnet**

Pros: Captures relation well.

Cons: Time complexity increases exponentially with any increase in the dimension.

Applications: When the data is coming from short sized videos like 20 secs or less with low resolution, it is a good choice. For variable length data, it will perform poor.

1. [10 pts] The computer vision community has primarily approached object detection using static/image data. In what ways might learning about objects from video make the object modeling task easier? In what ways does video make object modeling *harder*? In what ways might embodied learning, where an agent can approach objects and manipulate them (e.g. turn them or shake them) offer additional advantages beyond video? What are the challenges of learning in this embodied fashion?

From video, we can get multiple angles of an object. This will help object model to generalize better and more robust. We may need to store multiple configuration of same object which can make it harder. Als video processing is more costly in terms of computation as compared to image processing. Annotating videos are also expensive. Embodied learning helps the network learns the semantics better, 3D geometry and depth perception better and understanding of text better. The training will be slow as the data in generated online. Also there are risks to put agent directly in the environment. For example it is risky that a self driving car is learning on the roads to learn.

1. [10 pts] Unsupervised learning, where we have plentiful data but it is not semantically labeled, is beneficial in that it allows us to leverage massive "free" data, without incurring a high human labeling cost. What are some ways in which we can use unlabeled data to learn about the visual world? Do you think unsupervised learning will always be upper-bounded in terms of performance by supervised learning? Why or why not?

Examples:

Album creation where images of our pictures can be sorted according to event like party/outing.

Can help business group customers with similar behaviour to serve them better.

Unsupervised learning can find patterns which are unintended and spurious. It’s hard to find right kernel that will suit well the data. Supervised learning is limited unless the data is huge and is unbiased. Even then, the best it can perform is to the level of supervised learning. This is because if data from label1 and label2 are very close at some areas and sparse on other areas. Supervised learning will be able to make a separating line through them and unsupervised will make a cluster of both the group.