CS376: Computer Vision: Assignment 5

1 Face Detection (100 points)

The first programming assignment is to implement a face detector in natural images. We will follow the standard pipeline of sliding window + classification. For classification, please implement either a boosting framework or a SVM framework. For the boosting framework, please follow the Viola-Jones algorithm:

- Rapid object detection using a boosted cascade of simple features. CVPR 2001. `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.6807`. Note that there are many other resource to read, such as `https://en.wikipedia.org/wiki/Viola--Jones_object_detection_framework`.

For both programming assignments, please stick to the programming environment that you used in assignment 4.
We will use the Face Detection Data Set and Benchmark Home `http://vis-www.cs.umass.edu/fddb/`. Regarding training/testing, we will use the first 9 folders for training and the last 2 folders for testing. Details can be find in the readme file of FDDB`http://vis-www.cs.umass.edu/fddb/README.txt`:

- Training: FDDB-fold-00, FDDB-fold-01, $\cdots$, FDDB-fold-08;
- Testing: FDDB-fold-09, FDDB-fold-10;

Note that this dataset annotates an face using an ellipse. Please convert an ellipse into a rectangular box. One approach can be found at `https://stackoverflow.com/questions/87734/how-do-you-calculate-the-axis-aligned-bo`

In terms of evaluation, please report the precision-recall curve with an IOU threshold of 0.5.
Please submit the following:

- **Face classification (60 points):** Please implement an approach that takes an image patch as input, and output whether it contains a face in it or not. Note that how to generate the training data is important. For negative instances, please include non-face patches and faces that overlap with human objects but are too big or too small. We will look into how you train the classifier and how you generate the training data. Please report the classification accuracy of your approach.
- **Sliding window + Human classification (40 points):** Implement a sliding window approach to perform face object detection. Please include numbers and pictures of classification result. Please report the precision-recall curve with an IOU threshold of 0.5 on 'sec06'.

2 Extra Credits (50 points)

We are going to offer 50 extra credits for excellent algorithms and implementations that boost the performance of human detections in natural images. You are not restricted a particular approach. You can also use additional training data to train your object detector. However, the evaluation is always on testing set described above.

We are looking for creative solutions. Below are some ideas that you can implement:

- Train a neural network to classify each sliding window.
- Train a dense prediction network to regress the corners of the bounding boxes of the detected objects.

- Think about how to utilize the context, which is very important for detecting partially occluded human objects.

Please keep in mind that it is recommended to have unique ideas. **You are allowed to use any toolkit and packages for this part.**

We will grade this programming assignment from the following four perspectives:

- **Object detection (25 points):** We will run a competition here. Please report the precision-recall curve with an IOU threshold of 0.5. Please plot the precision-recall curve for our validation. You will get 20 points as long as you provide valid code, result number, and the precision-recall curve. Another 5 points will be based on the your accuracy, $> 80\%$ 5pt, $60 - 80\%$ 4pt, $40 - 60\%$ 3pt, etc.
- **Running time (15 points):** We again run a competition here. You will get 10 points as long as you provide a screenshot of running your code with time calculated using the built-in tool. Another 5 points will be based on the inference time for 100 test images: $< 1s$ 5pt, $< 10s$ 4pt, $< 100s$ 3pt, etc.
- **Creative ideas (10 points):** We encourage creative ideas for performing object detection. You will get 5 points with an approach that is not discussed in class. You will get 10 points if your approach differs from what has been proposed in the literature. Please clearly state your method in your writeup to claim the score for this part.

Below are some useful papers to look at for human object detection using neural networks:

- Is faster r-cnn doing well for pedestrian detection? `https://arxiv.org/pdf/1607.07032.pdf`
- A unified multi-scale deep convolutional neural network for fast object detection. `https://arxiv.org/pdf/1607.07155.pdf`
- Occluded pedestrian detection through guided attention in CNNs. `http://openaccess.thecvf.com/content_cvpr_2018/papers/Zhang_Occluded_Pedestrian_Detection_CVPR_2018_paper.pdf`
- Cornernet: Detecting objects as paired keypoints. `http://openaccess.thecvf.com/content_ECCV_2018/papers/Hei_Law_CornerNet_Detecting_Objects_ECCV_2018_paper.pdf`

Submission instructions:

Create a single zip file so submit on Canvas that includes

- Your well-commented code, including the files and functions named as specified above.
- A pdf writeup of your results with embedded figures where relevant.

Please do not include any saved matrices or images etc. within your zip file.