Assignment #2: Basic Detection

Nikolay Vasilev
IBB 22/23, FRI, UL
nv7834@student.uni-lj.si

I. Introduction

For the second assignment we will set up two popular detection methods. The first one is the Viola-Jones/Haar cascade detection method, which we will use to predict and optimize detection on ear images. The second method is the YOLO (v5) detection method, which we will also use to run predictions on the same ear images. We will compare the predictions of the images with both methods and calculate Intersection over union (IoU) and precision-recall (PR) scores over all thresholds with step of 0.01. Analyzing these results, will help us figure out how to optimize the VJ/Haar cascade detector, so it could work as good as the YOLO (v5) detector for ear detection.

II. Related Work

Before starting to work on the assignment, we have to find out more about Viola-Jones/Haar cascade detection method by reading a few articles about rapid object detection [1], about VJ/Haar cascade detection in general [2], [3] and about its implementation in Python [4]. With the help of these information we will be able to create a Python program that loads up XMLs and images and predicts VJ/Haar cascade detection for different parameters on ear images. With the help of some articles about the YOLO detection method [5]–[7], we will be able to set up also the YOLO v5 in Python [8]. With all this data we will compute accuracy about each of the methods. We will also calculate the IoUs [9] of and precision-recall scores [10] over all thresholds for these predictions.

III. Methodology

To do the experiment, we have to develop a program, which will read ear images and use VJ/Haar and YOLO detection on them. To be able to analyse the results we will add functionalities for calculating the accuracy of these predictions with the help of the ground-truths for each image (stored in .txt files). Because a simple accuracy is too exact, we will have to implement a way to analyse what percentage of the predicted data is in the ground-truths. That is why we will also set up functionalities for calculating the intersection over union for each photo's predicted data depending on its ground-truth for all thresholds from 0.01 to 1.00 for step of 0.01). This will help us calculate the average IoU for all 100 steps, but also to calculate the precision-recall scores for each threshold.

IV. Experiments

To do the experiment we will use VJ/Haar with different parameters and YOLOv5 detection over 500 ear images. We will compare each image's predictions with its ground-truth to calculate the accuracy, IoU and precision-recall scores over all thresholds. The developed Python program will give us all these results for a chosen parameters of VJ/Haar such as scale factor (SF) and minimum neighbors(N). These results used in another Python program will allow us to draw plots, that help us easier evaluate the results.

V. Results and Discussion

We have done VJ/Haar cascade detection for different parameters (SF=1.01/1.03/1.05; N=1/3/5) to achieve as high results as possible. The usage of YOLOv5 help us see ,how good our cascade detector works in comparison with YOLOv5. The used ear images are different and have different sizes, that is why the VJ detector doesn't work good on each of them. The evaluation helps us also find out, which images give better results, which worse and why.

A. Results

First we will start with the predicted values of VJ/Haar cascade detector and YOLOv5 algorithm. Figure 1 shows us the average accuracy, based on the supplied ground-truths and the predictions over all 500 ear images for VJ detection for all the parameters and the YOLO detection.

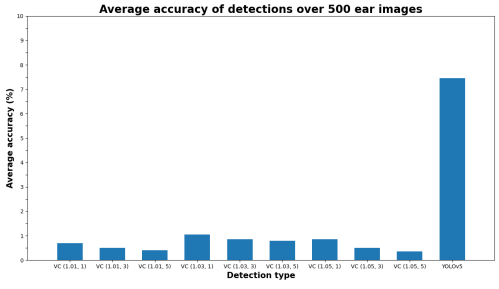

Fig. 1. Average accuracy of VJ and YOLOv5 detections over 500 ear images

On the Y-axis is the average accuracy in percent only from 0% to 10%. On the X-axis are all types of predictions we have made. The values are not very good - around 0.5 % and 1.5% for VJ and 7.5% for YOLOv5. From these results

we can see, that the best prediction for VJ detection is made with parameters SF=1.03 and N=3 with around 1.5%. But we cannot believe these results, since we are checking if the predicted values are exactly the same as the ground-truths values. In the case of detection, this is not a good way to evaluate, how good the detection works.

That is why we will calculate the average IoU for thresholds from 0.01 to 1.00 with step 0.01. It will show us the percentage of intersection over union between the predicted boxes and the ground-truth box, which will be valid if it is bigger or equal to the threshold. This will give us better information about the accuracy of the VJ and YOLOv5 detections, because we don't always need exactly the same values as the ground-truth to have a valid detection. On figure 2 we can see the average IoUs of VJ for all parameters and of YOLOv5 for each threshold. On the Y-axis is the value of the average IoU in percent, and on the X-axis are the thresholds. Now we can see that we get better results - between 18% and 30% for VJ and around 83% for YOLOv5. We can see that the VJ detection gives a lot lower result then the YOLOv5, which is normal. The important think here is to figure out, which parameters give best Average IoU for the VJ/Haar detection. As we can see the worst accuracy in general is for the parameters SF=1.05 and N=5 and the best in general is for the SF=1.01 N=1/3/5. If we follow, how the average IoU changes for these different thresholds, we can figure out that the VJ with parameters SF=1.01 and N=5 gives the best results for thresholds from 0.01 until around 0.70, from the threshold 0.70 until 0.85 the best results are for the parameters SF=1.05 N=1, and the parameters SF=1.03 and N=1 give the best values for the threshold from 0.85 until 1.00. Something else, we can see from the graph is that all VJ predictions have pretty much constant IoU until the threshold 0.60. After that the accuracy is going down and after threshold 0.90 they all have average IoU around 0. In comparison to these results, the YOLOv5s average IoU is constant almost until the threshold 0.80 and it hits the value 0 around the threshold 1.00, which means that the predictions are more accurate in general.

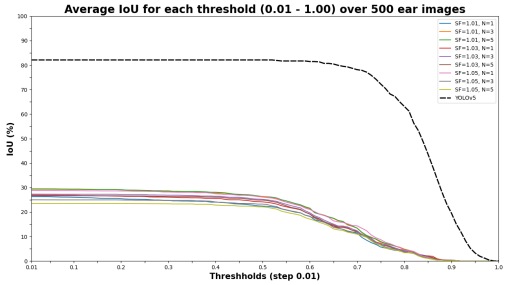

Fig. 2. Average IoU of VJ and YOLOv5 for each threshold (0.01-1.00, step 0.01) over 500 ear images

To figure out how good is a detection, we don't need to know only how good the accuracy is, we need to know also the precision and the recall of the detection. IoU told us how many times the detection was correct overall. Precision will give us an information about how good the detection is at predicting a specific category and the recall will tell us how many times were we able to detect a specific category. Figure 3 shows us PR curves for all the VJ and YOLOv5 predictions for all thresholds.

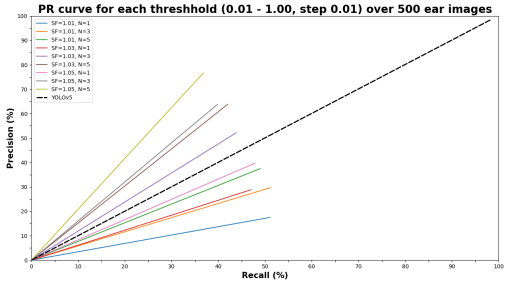

Fig. 3. VJ and YOLOv5 PR curve for each threshold (0.01-1.00, step 0.01) over 500 ear images

From the graph we can see that the YOLOv5 detection has equally good recall and precision from 0% to almost 100% for all thresholds, but the VJ detections not so much. For the parameters SF=1.05 and N=5, we get very good precision score (around maximum 80%), but not so good recall score (around maximum 40%). On the other side for the parameters SF=1.01 and N=1/3, we get good recall score (around maximum 55%), but very bad precision score - around maximum 15% for N=1 and around maximum 30% for N=3. That means that, if we want to use the VJ detection with good recall most probably we will choose the parameters SF=1.01 and N=3, but if we want a good precision we will choose the parameters SF=1.05 and N=5. What detection we choose is up to the things we want to detect with it. Mostly good recall score is needed for live feed detection and good precision score is more compatible with images detection. But knowing the information from the precious figure, that the best average IoU value in general over all thresholds for VJ detection is for SF=1.01 and N=1/3/5, we can say that the parameters that give the best results for VJ/Haar detection are SF=1.01 and N=3. Now when we have chosen the best parameters for the VJ cascade detection, we will find a set of 10 failed and 5 best VJ predictions and will compare them with the YOLOv5 predictions.

TABLE I
SET OF 5 REPRESENTATIVE BEST VJ (SF=1.01, N=3) PREDICTIONS COMPARED TO YOLOV5 PREDICTIONS OVER THE 500 EAR IMAGES

Image Name	**VJ** $(TP\|ALL\|IoU)$	**YOLOv5** $(TP\|ALL\|IoU)$
0614.png	1|3|90.98%	1|1|64.82%
0612.png	1|4|89.64%	1|1|90.15%
0510.png	1|2|87.62%	1|1|86.49%
2046.png	1|1|86.67%	1|1|92.14%
0577.png	2|2|86.64%	1|1|86.60%

On table I we can see the best 5 VJ (SF=1.01, N=3) predictions compared with their YOLOv5 values. Each image is presented

with its VJ's and YOLOv5's true positive predictions (TP), all predictions (ALL) and IoU value (IoU).

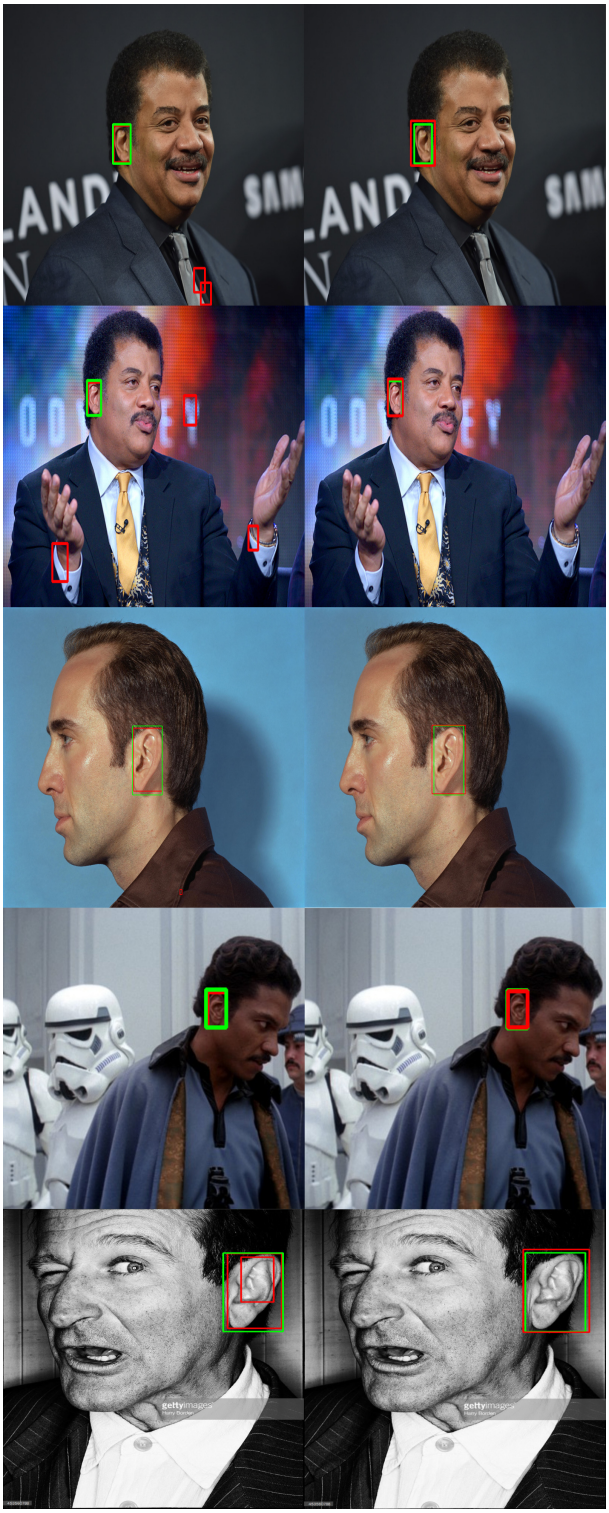

Fig. 4. Example of the best 5 representative VJ (SF=1.01, N=3) predictions compared to YOLOv5 predictions

We can see that all of them have IoU for VJ prediction bigger than 86% and some of them are even bigger than the IoU of the

Fig. 5. Example of the 10 failed representative VJ (SF=1.01, N=3) predictions

YOLOv5. For example the first image (0614.png) has IoU of 90.98% for VJ and IoU of 64.82% for YOLOv5, which makes it with 30% more accurate. There are also other images that have bigger IoU for VC than YOLOv5 (0510.png, 0577.png), but even these, that have smaller value (0510.png, 2046.png)

in comparison with the YOLOv5, don't have so much smaller. Example of these photos with drawn prediction and ground-truth boxes can be seen on image 4, where the images are ordered in the same order as they are in the table. On the left side of each row is an example of the VJ detection and on the right side is example of the YOLOv5 detection. For each image the green box represents the ground-truth and the red boxes represent the predictions.

Of course we need to mention also some failed predictions, which will help us figure out how can we make the VJ detection work better. On table II we can see 10 failed VJ predictions, which have most all invalid detections.

TABLE II
SET OF 10 REPRESENTATIVE FAILED VJ (SF=1.01, N=3) PREDICTIONS WITH MOST DETECTIONS OVER THE 500 EAR IMAGES

Image Name	**VJ predictions**
2181.png	41
0537.png	10
2121.png	7
0573.png	6
2119.png	6
0583.png	5
0656.png	5
1967.png	5
2014.png	5
2030.png	5

As we can see only the first image has very high number of detections (41), while the other have 10, 7, 6 and most of them 5 detections. Example of these detections can be seen on image 5, where the images are ordered in the same way as they are in the table. For each image the green box represents the ground-truth and the red boxes represent the predictions.

B. Discussion

Now it's time to evaluate the results and find out the reason, why the scale factor 1.01 and the minimum neighbors 3 are the best parameters for the Viola-Jones/Haar cascade detector. The scale factor specifies how much the image size is reduced at each image scale. By rescaling the input image, we can resize a larger ear to a smaller one, making it detectable by the algorithm. Using 1.01 we increase the chance of matching, which means that using this parameter, we reduce the number of images that will have no predictions at all. The second parameter specifies how many neighbors each candidate rectangle should have to retain it, which impacts also the quality of detected faces. Higher value will give us fewer detections but with higher quality that is why the value in between - gives us enough predictions with good enough quality.

Some other evaluation we can make is this time from the results of the best VJ predictions and the failed VJ predictions. From the photos it becomes clear that the detection mostly fails when the ear is in significant angle (not whole ear visible, inner ear not visible), there is occlusion in front of the ear (hair), the lightning is different where the ear is (shadow, brightness), the ear has specific structure (size, more rounded, flattened) and others. By focusing on increasing the IoU for the failed images, we can rapidly increase the results of the VJ detection in general.

VI. CONCLUSION

In this assignment, we analysed Viola-Jones/Haar cascade detection method compared to the YOLOv5 detection using a Python program, which calculated simple accuracies, IoUs and precision-recall scores for each threshold from 0.01 to 1.00 with step size of 0.01. With the experiment, we found out that the VJ detections gives us best results, when we use a scale factor 1.01 and minimum neighbors 3, which give us average IoU of around 30% for threshold from 0.01 until almost 0.55, maximum recall of more than 50% and maximum precision of more than 30%. The final results help us figure out, that the VJ detections works good for the images, where a right detection is found (some even better than YOLOv5). Using the analysis from the best 5 predictions and the 10 failed predictions, we now know that if we focus on fixing these problems, we can achieve very high results with the Viola-Jones/Haar cascade detector.

REFERENCES

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–I.

[2] C. Rahmad, R. A. Asmara, D. Putra, I. Dharma, H. Darmono, and I. Muhiqqin, "Comparison of viola-jones haar cascade classifier and histogram of oriented gradients (hog) for face detection," in *IOP conference series: materials science and engineering*, vol. 732, no. 1. IOP Publishing, 2020, p. 012038.

[3] D. Peleshko and K. Soroka, "Research of usage of haar-like features and adaboost algorithm in viola-jones method of object detection," in *2013 12th International Conference on the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, 2013, pp. 284–286.

[4] (2021, 07) Viola Jones Algorithm and Haar Cascade Classifier. [Online]. Available: https://towardsdatascience.com/viola-jones-algorithm-and-haar-cascade-classifier-ee3bfb19f7d8

[5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[6] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022, the 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 2021): Developing Global Digital Economy after COVID-19. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922001363

[7] Y. Zhao, Y. Shi, and Z. Wang, "The improved yolov5 algorithm and its application in small target detection," in *International Conference on Intelligent Robotics and Applications*. Springer, 2022, pp. 679–688.

[8] (2022, 10) Object Detection Inference in Python with YOLOv5 and PyTorch. [Online]. Available: https://stackabuse.com/object-detection-inference-in-python-with-yolov5-and-pytorch/

[9] (2022, 04) Intersection over Union (IoU) for object detection. [Online]. Available: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

[10] Precision and Recall in Python. [Online]. Available: https://www.askpython.com/python/examples/precision-and-recall-in-python