Assignment #2: Basic Detection

Luka Boljević
IBB 22/23, FRI, UL
lb7093@student.uni-lj.si

I. Introduction

This assignment is about setting up two popular detection methods, Haar cascades and YOLOv5, for an ear detection task. The goal is to evaluate (and in the case of Haar cascades, optimize) their performance, calculating mean IoU, precision/recall, and average precision/recall scores for both models.

II. Related Work

The Haar cascades model, also called the Viola-Jones model, was presented in the paper by Viola and Jones in 2001 [1]. The original YOLO paper, published in 2015 by Redmon et al., presented a brand new way for object detection [2]. It has since seen multiple improvements across different version, with the latest version being YOLOv7 at the time of writing this report [3]. For this assignment, we will be using YOLOv5 [4] (there is no paper for YOLOv5).

III. Methodology

The two models are pretrained, and provided alongside the ear dataset, so we only need to load them. To begin with, we calculate the mean IoU, for YOLOv5 and different parameter settings for the Viola-Jones (VJ) model. Then, for some IoU thresholds (threshold which determines whether a detected bounding box is TP or FP) we will report precision and recall values, while also reporting the average precision and recall across all IoU thresholds from 0 to 1, with a step size of 0.01. **Note:** this average precision (recall) is not the same as the usual AP/mAP (AR) metric used in literature. To calculate (m)AP, we need confidence scores from the model, which we don't get from OpenCV's implementation of VJ (which is what are using in this assignment). In our instance, average precision (recall) refers to the actual average of precision (recall) values, across the 100 IoU thresholds.

IV. Experiments

The dataset consists of 500 images of celebrities' ears, with each image containing exactly 1 ear. Each image is accompanied with a file describing the position and size of the ground truth box of the ear. As stated, we will first report the mIoU for the default models. To calculate mIoU, for each image, we do the following: (1) run the model to get the bounding boxes, (2) calculate IoU for each detected bounding box against the ground truth box, (3) save the calculated IoUs in a vector. After going through all the images, sum up the vector with all IoU values, and divide by the length of the vector (number of detected bounding boxes) to get the mean IoU.

After that, we will try to optimize VJ by adjusting its parameters, to try to get as high a mean IoU as possible. The parameters we can vary are `scaleFactor`, `minNeighbors`, and `minSize`. We will see how changing them affects mIoU. We will also report (average) precision and recall values for some VJ models, for IoU thresholds of 0.2, 0.5, 0.7, 0.8 and 0.9. We picked those because they seemed to produce the most representative results during testing. Also, we note that there can be at most one TP on each image, as there is only one ground truth box (ear) in it.

We will see how the precision and recall values change as the thresholds increase, while also trying to establish if mIoU of the particular VJ model has any connection with the precision and recall values. Following that up, we will show the (average) precision and recall values of the YOLOv5 model (for the same thresholds), so we can compare the two models.

V. Results and Discussion

Let us now show the results/performance of our models. We will follow each presented result with a slight explanation or clarification, with the discussion coming after presenting all the results.

A. Results

As stated, let us begin with Table I, where we present mIoU values for the **default** VJ model, and for YOLOv5.

TABLE I
Mean IoU for the default VJ model and for YOLOv5. Based on the mIoU values alone, we see that YOLOv5 is capable of more accurate detection.

Default VJ	YOLOv5
0.536	0.821

Now, let's see what happens when we play around with different VJ parameters.

Table II shows us how different values for parameters of VJ affects mIoU. Looking at the Table row wise, we see that increasing the `minSize` parameter results in a higher

TABLE II
MIOU VALUES FOR DIFFERENT PARAMETER SETTINGS OF THE VJ MODEL.

scaleFactor, minNeighbors / minSize	30x30	70x70	100x100
1.05, 3	0.454	0.585	0.587
1.05, 6	0.593	0.675	0.675
1.10, 3	0.579	0.685	0.675
1.10, 6	0.625	0.678	0.674
1.20, 3	0.645	0.704	0.711
1.20, 6	0.646	0.674	0.700

mIoU. Of course, since this parameter denotes the minimum possible object size, increasing it can only go so far, since at some point we will just stop detecting ears in general. On the other hand, looking at the Table column wise, we see that increasing `scaleFactor` and/or `minNeighbors` does not necessarily increase mIoU. For example, in the case of `scaleFactor=1.2`, increasing `minNeighbors` decreased mIoU. Of course, a higher mIoU may not directly correlate with more and better quality detections. We are only able to see that with precision and recall values of the models - which is exactly what we'll look at next.

Let's have a look at (average) precision and recall values (at various IoU threshold values) of some representative VJ models, so we can draw a conclusion more easily.

TABLE III
PRECISION AND RECALL VALUES FOR DIFFERENT VJ MODELS, AT DIFFERENT IOU THRESHOLDS, AS WELL AS THE AVERAGE PRECISION AND RECALL VALUES ACROSS ALL THRESHOLDS FROM 0 TO 1, WITH A STEP SIZE OF 0.01. PRECISION@0.2 MEANS THAT THIS IS THE PRECISION VALUE FOR THIS MODEL AT IOU THRESHOLD 0.2. "1.05-3-30x30" MEANS THAT THIS IS A VJ MODEL WITH PARAMETERS `SCALEFACTOR=1.05`, `MINNEIGHBORS=3`, AND `MINSIZE=(30, 30)`.

. / VJ model	default	1.05-3-30x30	1.1-6-30x30	1.2-3-100x100
Precision@0.2	0.819	0.680	0.939	1.000
Recall@0.2	0.372	0.386	0.248	0.066
Precision@0.5	0.744	0.613	0.856	0.939
Recall@0.5	0.338	0.348	0.226	0.062
Precision@0.7	0.330	0.306	0.424	0.545
Recall@0.7	0.150	0.174	0.112	0.036
Precision@0.8	0.093	0.109	0.129	0.303
Recall@0.8	0.042	0.062	0.034	0.020
Precision@0.9	0.004	0.004	0.008	0.000
Recall@0.9	0.002	0.002	0.002	0.000
Avg precision	0.541	0.458	0.631	0.716
Avg recall	0.246	0.260	0.167	0.047

Table III indeed shows us that a higher mIoU does not necessarily correlate to a "better" model. The model with parameters "1.2-3-100x100" achieved the highest mIoU, as seen in Table II, but in a way it has the worst performance - with a recall of only 0.062 at IoU threshold 0.5. Out of the ones shown, we would say that actually, the default model *looks* the best, even though it had an mIoU of 0.536 (Table I). Generally, if we compare precision and recall values of models from Table III with corresponding mIoU values from Table II, we see that a higher mIoU generally means higher (average) precision and smaller (average) recall values. This behaviour was noticed throughout our testing, and it means that, at higher mIoU, our VJ model becomes "picky", so it finds fewer ears in total, but it's confident about those detections.

Finally, let's show the results of the YOLOv5 model.

TABLE IV
PRECISION AND RECALL VALUES FOR YOLOV5, AT DIFFERENT IOU THRESHOLDS, AS WELL AS THE AVERAGE PRECISION AND RECALL VALUES ACROSS ALL THRESHOLDS FROM 0 TO 1, WITH A STEP SIZE OF 0.01.

. / IoU threshold	0.2	0.5	0.7	0.8	0.9	Average
Precision	0.984	0.984	0.920	0.719	0.194	0.825
Recall	0.984	0.984	0.920	0.719	0.194	0.824

B. Discussion

We can argue that the "pickiness" of VJ models we mentioned is not exactly the behaviour we want from a detector - we would much rather have a high level of precision, while also having higher recall. Unsurprisingly, this is exactly what we get (and better) with YOLOv5. We can quite easily see from Table IV that YOLOv5 *completely* outperforms any VJ model that we tested (and shown here). Namely, even at 0.7 IoU threshold it still manages to achieve incredibly good performance, while VJ models already start "failing" at that point. It was hard to *truly* optimize VJ in this experiment, as for higher recall values, the model has very low recall, and vice versa, and even the "best" VJ models do not even come close to that of YOLOv5.

VI. CONCLUSION

In this assignment, we evaluated the performance of two popular object detection models - Viola-Jones and YOLOv5. We have calculated different metrics to evaluate their performance. From the presented results, we can conclusively say that YOLOv5 is the better model. We also saw, in the case of VJ models, that a higher mIoU, in the way it is calculated, does not necessary mean a "better" model (based on precision and recall values). A better indicator of performance would definitely be mAP, but unfortunately OpenCV's implementation of VJ does not return any confidence scores - meaning we cannot calculate mAP.

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] Redmon, Joseph and Divvala, Santosh and Girshick, Ross and Farhadi, Ali, "You Only Look Once: Unified, Real-Time Object Detection," 2015.

[3] Wang, Chien-Yao and Bochkovskiy, Alexey and Liao, Hong-Yuan Mark, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022.

[4] Ultralytics, "YOLOv5 GitHub repo," Link to repo.