Benchmarking Face Detectors

Ethan Wilson
University of Florida

Jenny Skytta
Seattle Children's Hospital and Research Institute

Frederick Shic
Seattle Children's Research Institute

Eakta Jain
University of Florida

1 Introduction

For algorithmic tasks revolving around facial manipulation, a prerequisite is to have bounding boxes demarcating all faces within an image set. There are a large number of existing automated facial detection algorithms. Among the current state of the art there are trade-offs between speed and accuracy. We investigate two algorithms that are well-suited to our use case: automatic detection of child faces in video recordings of behavioral observation sessions. We propose a second pass over one of the methods to reduce the time taken to process each video while preserving a high level of accuracy.

2 Method

We benchmark two approaches for automatic face detection: MTCNN and S3FD. Both are state of the art deep learning approaches and are meant to perform in real time on modern hardware.

2.1 MTCNN

The Multi-task Cascaded Convolutional Network (MTCNN) [2] is a framework to analyze multiple resolutions of a given image with a cascading network design. The network operates by feeding scaled copies of the image through a lightweight fully convolutional network (CNN) to obtain bounding boxes that may contain a face, called "candidate windows". The candidate windows are then fed through a refinement CNN that deletes false candidate windows and calibrates the windows that survive the process using bounding box regression. This calibration more tightly layers the bounding boxes to represent a more distinct area.

Finally, the windows are passed through a third CNN layer that computes a single bounding box along with five key facial landmarks (position of eyes,

nose, and corners of the mouth). Between each aforementioned phase, a non-maximum suppression (NMS) step is performed to merge overlapping candidate windows within a threshold. Our implementation of this algorithm came from a python package build off of Tensorflow and Keras [1].

2.2 S3FD

The Single Shot Scale-invariant Face Detector (S3FD) [3] was created to specifically address the problem of small faces. The architecture revolves around a deep network consisting of anchor associated layers. As information is fed through these layers, the stride size continually doubles from 4 to 128 pixels. The information from these layers map to anchors of set resolutions, which helps to associate faces with anchors at a large variety of resolutions. We use the S3FD implementation available as the *de facto* face detector used in DeepFaceLab (DFL) [1], a popular deep faking application. A not well publicized implementation detail of DeepFaceLab's S3FD detector is that they down-sample their images before feeding them into the model, sacrificing model accuracy for a significant increase in speed and computational efficiency.

2.3 Our method: Modified S3FD

After comparing the two methods, we created a modification to the S3FD original method that allowed us to feed images at full resolution without slowing down the run time. Instead of running face detection on every frame of our 60fps video clips, we run the S3FD face detector on every 1/X frames [2]. This modification utilizes the S3FD detector's specialization in finding small faces on the full resolution frames of our video clips, yielding a high detection rate. After detection, our system looks $X + 1$ frames forward from each face to determine if there is a next detected face. If there is a next face, the median frames are filled in with bounding boxes derived from interpolating the two real faces.

2.4 Datasets

To compare these face detectors, we selected a few video samples relevant to our specific face detection task. The videos selected are as follows:

- **ADOS Clip, Room Layout A:** Clinical session where the camera has been positioned such that the patient's seated position is in view and the patient's face is available at a moderately high resolution. The patient and clinician are situated side-by-side and are both visible for the vast majority of the session.
- **ADOS Clip, Room Layout B:** Clinical session with a nonideal camera angle. Camera is positioned in the corner of the room to where the patient's face is consistently low resolution. The clinician is seating facing

[1] https://pypi.org/project/mtcnn/
[2] Thus far, we've used the value X=10. This can be adjusted if need be.

away from the camera, and a parent can be seen sitting in the back of the room.

We additionally perform a comparison between two distinct groups that we identified from our total sample of 50 recorded ADOS sessions. These are room layouts A and B, which are represented above by individual samples. We report the detection rates produced by our modified S3FD detector among all ADOS sessions when considering the two room layouts as distinct categories in the next session.

Figure 1: The room layouts of the sessions used for benchmarking. Layout A on the left, layout B on the right.

3 Results

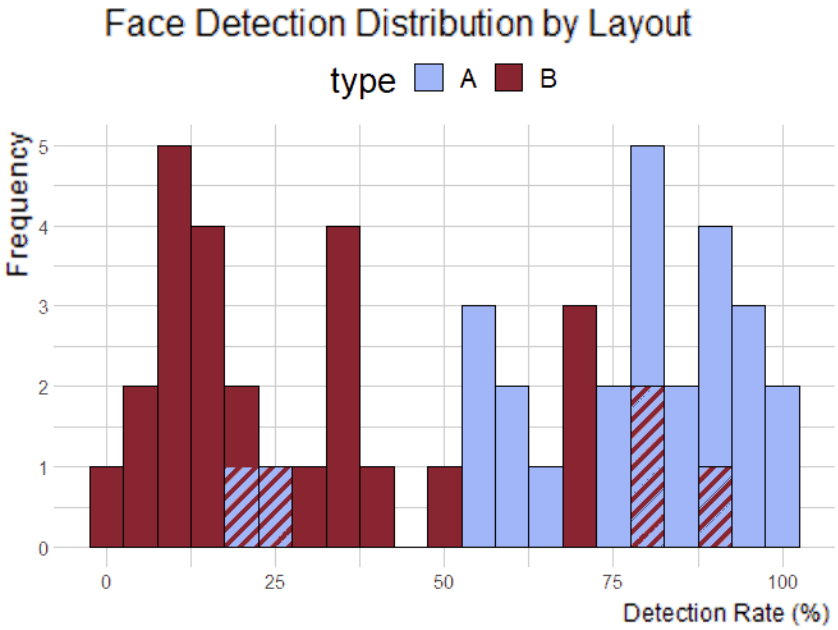

Figure 2: Distribution of single face Face Detection results organized by room layout.

Our results yielded some significant trade-offs between the two main methods. The MTCNN was more accurate in the more difficult case of layout B. However, the MTCNN's run-time was an order of magnitude higher than that of the S3FD. This was not a worthwhile compromise for our use case, as the clinical sessions need to be processed in a reasonable amount of time. Our modified S3FD was significantly quicker than the original S3FD. In the layout A case, it outperformed the original S3FD, likely due to the modification in which we consider frames at full resolution instead of half resolution, an implementation detail of DFL as mentioned previously.

We additionally report the distribution for detection rates organized by room layout, as shown in 2. These detection rates were computed using the Modified S3FD. This figure shows a distinct split between layouts A and B. Despite a few outliers on each side, it is clear that layout A is vastly preferable when performing automatic face detection.

Session (Layout)	Detector	Time (Minutes)	Detection Rate (Single Face)	Detection Rate (Multiple Faces)
A	MTCNN	1148.7	87.76%	33.93%
A	S3FD	123.8	91.88%	62.57%
A	ModS3FD	19.8	95.28%	86.47%
B	MTCNN	1127.1	29.58%	6.64%
B	S3FD	228.5	22.06%	3.35%
B	ModS3FD	44.3	15.70%	2.89%

Table 1: Results of automatic face detection and runtimes for the three detectors being considered. Results are shown for both for session A (with high detection rates across the board) and session B (a much harder case).

4 Discussion

We selected the S3FD face detector because of its speed. The MTCNN, despite having a high level of accuracy, was not time efficient enough to be feasible for our project. We hypothesize the MTCNN's drastically lower time efficiency to be an implementation detail. The open-source python package that we sourced was obviously not designed for processing video. After making our selection, we further investigated potential optimizations and arrived the Modified S3FD detection system which optimized speed while maintaining performance.

Our experiments suggest a preferred room and camera layout for video taping behavioral observation sessions for downstream processing. If a recorded session is to be processed, it is imperative to position the camera such that participants' faces are visible and largely present at a workable resolution.

Our findings additionally indicate that a future line of work might include developing methods for reliable automatic face detection in videos where the subject changes head orientation rapidly or moves around a lot. In the near term, it is necessary to use manual annotation to catch the missed faces and ensure total coverage (because our purpose is privacy protection, we cannot rely solely on automatic face detectors as a single missed frame can compromise an entire session).

References

[1] Perov, I., Gao, D., Chervoniy, N., Liu, K., Marangonda, S., Umé, C., Dpfks, M., Facenheim, C. S., RP, L., Jiang, J., Zhang, S., Wu, P., Zhou, B., and Zhang, W. Deepfacelab: A simple, flexible and extensible face swapping framework, 2020.

[2] Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal*

Processing Letters 23, 10 (Oct 2016), 1499–1503.

[3] Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., and Li, S. Z. S^3fd: Single shot scale-invariant face detector, 2017.