CS82 Final Project “Blur the Baby”
Team: Harold Kim and Mary Zhuo Ke

Ensemble Method to Detect, Deidentify, and Blur Faces in Videos Automatically

Improving facial detection for infants and children in videos and image data helps de-identify data to protect privacy of minors. This is especially important when infants and children are not making these decisions for themselves during a period where, once data is shared, there’s no telling where it could end up. Lacking a way to de-identify faces not only has privacy risks in the future when these minors become adults, but lack of sharing these data within the research community stifles innovation because of privacy concerns of sharing identifiable data. Failure to abide by privacy laws can lead to high penalties, so there is a vested interest in improving our ability to de-identify data. A way to better detect child faces at scale (i.e., from videos) with greater accuracy without too much computational demand would help overcome the obstacle with more certainty. We hypothesize that there will be face detection models that better detect some features characteristic of infants and children and can leverage those to optimize object classification for this age group. This may mean we need more sophisticated ways to be able to detect their faces in photos and videos.

We will use an open source dataset of videos of infants and children. To start as a baseline assessment of the models, we use the WIDER FACE validation dataset. WIDER FACE has been used as an evaluation dataset due to its richness in variety of expressions, occlusions, scales, orientations, and image quality and lighting[1] . Using this validation dataset, we extracted photos of families that included a mixture of adult and infant, child faces in order to test the performance of the face detection on different age groups. After we optimized and fine-tuned the best model from testing performance on the WIDER FACE validation dataset, which we used because it has the ground truth data on how many faces are in each image, we used the best model on a real video dataset of infants and children. We selected a video that had many challenging cases that would stress test the model we selected, including animations, noise, varying orientations. Thus, our goal is to detect 100% of all frames with a baby’s face in a video recording, because in data privacy, it’s not effective unless it’s 100%.

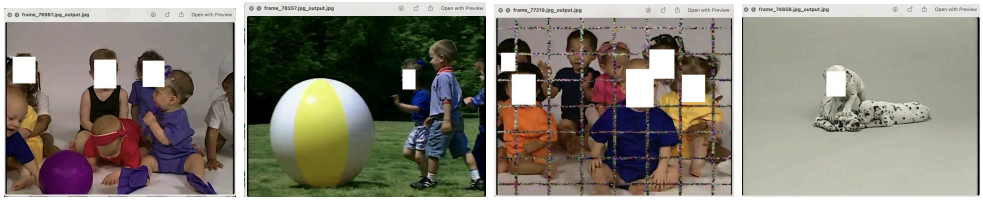
We tested existing pre-trained models that used the WIDER FACE dataset: YuNet[2] and RetinaFace[3]. YuNet uses far fewer parameters than other models and can perform better on smaller edge devices, while still preserving depthwise and pointwise convolutions that help improve performance on the most challenging to detect faces (e.g., small faces) using a tiny feature pyramid network (TFPN)[2]. YuNet has advantages of being fast due to its simplicity. RetinaFace, on the other hand, uses a ResNet and Feature Pyramid Network (FPN). It handles different scales of faces well, is much more computationally intensive, and uses techniques that are comprehensive enough to potentially estimate 3D pose and face attributes that improve accuracy of detecting side face views and other views of the face[3].

After we ran YuNet and RetinaFace on WIDER FACE[1], we assess accuracy compared to the ground truth number of faces, and then combine YuNet and RetinaFace using an ensemble method. We hypothesized that RetinaFace would perform better on all cases, given it is slower and supposedly more accurate with the benefit of higher compute devices. We aim to consider combinations that are not only accurate, but also computationally efficient. Then, to evaluate the model on infant videos, we prepare several “challenge” cases where the model struggles to recognize some of the angles of baby, infant faces. We extract frames of infants from each video by determining the timestamps where we see babies in the Brainy Babies video[4].

When testing YuNet, we adjusted some of the built-in settings, like tolerance threshold, in order to understand how YuNet’s face detection changed depending on this feature. Here are samples of missed images at threshold = 0.9. We notice it misses several challenging cases

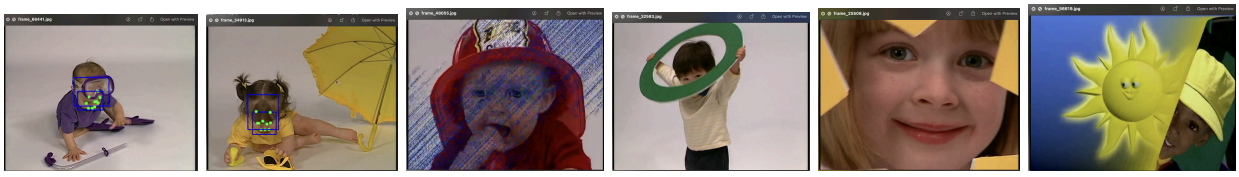
where a face is behind another face, or there is some “animation” noise of an overlay of frames. We also miss sideways and downward looking faces.

We tried again with a lower threshold confidence of 0.8 in order to see if it would pass some of the faces it was less confident about in the previous test.



However, even with a lower confidence threshold, YuNet wasn’t able to detect some of these faces with enough confidence. Unfortunately the lower confidence also makes it detect faces of dogs, which is incorrect. YuNet still misses obscured faces. We initially tried to use the output of YuNet with tens of thousands of annotated baby faces to re-train YuNet on these images, but ran into several complications with PyTorch erroring out when trying to apply transfer learning with additional data.

Despite the setback, we still needed to detect the remaining faces that posed a challenge, so we used RetinaFace, which is more accurate than YuNet, but slower. When testing only on RetinaFace, it missed cases or identified 2 faces if the child was wearing something that looked like a face. It also had a different time detecting half-covered faces or really large faces where the head isn’t completely visible. However, it was much more accurate with very small, blurry faces, and detecting many faces that YuNet missed, as expected.



What if there were advantages to both models, where we would have even higher accuracy with the combination of the strengths of both, with a faster computational speed. Combining YuNet and RetinaFace’s face detection may help us achieve the 100% for the challenge cases we had in the beginning. Thus, we try the ensemble of the two, by running RetinaFace on only the frames that YuNet missed or felt less confident about, therefore maximizing its speed based on only what is necessary. We also try another ensemble method where we run both RetinaFace and YuNet, and weight the one that has the most accurate detection through an intersection and union of both models.

Metric	YuNet	RetinaFace	BTB w/ Intersection and Union	BTB w/ YuNet as first pass and RetinaFace as second pass
Number of Frames (with and without human faces)	14,434	14,434	14,434	14,434
Number of Faces (Human or Not) Detected	11,066	15,456	15,479	~15,400
Time to Blur	5.95 minutes	2.72 hours	2.75 hours	1 hour

Based on the performance of these models (assessed by number of confident faces detected and also a manual view of the videos post-blurring to see if there are any undetected faces that would violate the privacy), we see that the multi-scale approach of these models significantly improves the ability to detect infant to child faces. However, because there are flaws in both models, we hope to make a happy medium that takes advantage of each model’s strengths where appropriate. We see that the ensemble performed better in the number of faces detected of high confidence (0.9 at least) and even detected additional faces beyond what RetinaFace alone could detect, contrary to what we originally thought. However, it takes too long, so the model that only feeds data that YuNet is not confident about to RetinaFace achieves the same number of faces but takes less than half the time!

With this, we demonstrated the benefit of seeing a middle ground to some models in speed and accuracy, so that we can help scale these sorts of solutions to more problems in the future. The hope is that this project also inspires handling transitions/animations more robustly as videos increase the variations of orientations/scales especially if it were 60 fps.

References:

1. Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). WIDER FACE: A face detection benchmark. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5525–5533. http://shuoyang1213.me/WIDERFACE/

2. Wu, W., Peng, H. & Yu, S. YuNet: A Tiny Millisecond-level Face Detector. Mach. Intell. Res. 20, 656–665 (2023). https://doi.org/10.1007/s11633-023-1423-y
    a. https://github.com/opencv/opencv_zoo/blob/main/models/face_detection_yunet/face_detection_yunet_2023mar.onnx (need to download this model)
    b. Support from chatGPT to help understand YuNet.

3. J. Deng, J. Guo, E. Ververas, I. Kotsia and S. Zafeiriou, "RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 5202-5211, doi: 10.1109/CVPR42600.2020.00525.
    a. Support from chatGPT to help understand RetinaFace.

4. Baby video dataset: https://archive.org/details/BrainyBabyShapesandColors