FACE DETECTION IN LARGE DISTANCE(FADILD)

Jiechang Guo
Department of Computer Science
University of Houston
UHID: 2084258
jguo34@cougarnet.uh.edu

Meng-chen Lee
Department of Computer Science
University of Houston
UHID: 1985750
mlee45@uh.edu

May 10, 2023

ABSTRACT

The primary objective of this project is to tackle the challenging task of detecting faces at long distances. To accomplish this, we employed YOLOv8, an advanced object detection algorithm, and utilized the WiderFace dataset, which is widely recognized as the benchmark dataset for face detection. The yolov8 model was initially fine-tuned on the original data of the WiderFace dataset. We hypothesized that by using Super Resolution techniques to enhance the resolution of the blur faces in long distances, the model's performance could be improved. Therefore, we preprocessed the images labeled as heavy blur with Super Resolution. The enhanced data was then used to fine-tune the yolov8 model as well. Moreover, we added a transformer layer to the yolov8 model architecture and trained the model on both the original and enhanced data. Overall, we trained four models, and our results surpassed the baseline with a smaller model size as well as less training time.

1 Introduction

Face detection is the first step for many face-related applications, such as face recognition, face tracking, facial expression recognition, facial landmarks detection, and so on. Those technologies can achieve an overall better performance by faster and more accurate face detectors. Other than that, face detection itself is also important, especially at long distances. For example, face detection over long distances is needed for surveillance and security applications. With the ability to detect faces at long distances, cameras and other surveillance systems can cover larger areas, leading to more effective monitoring and tracking of individuals. Face detection at long distances can also be used in other applications, such as tracking suspects in a high-speed chase or persons of interest in search and rescue operations. With the increasing use of surveillance cameras and other systems, the need for accurate and efficient face detection at long distances will likely continue to grow.

The challenges associated with detecting faces at large distances stem from the nature of the task itself. As stated in the literature [1], detecting faces at long ranges refers to distances of 10-20 meters indoors and 50-300 meters outdoors. One of the major challenges is that at larger distances, the size of the face in the image becomes smaller, leading to lower resolution. This, in turn, can make it difficult to detect and recognize facial features. Additionally, faces at large distances may be subjected to different lighting conditions, such as shadows, glare, or reflections, which can affect the quality of the face image and make detection more challenging. Another issue is that faces at larger distances may appear at different poses and orientations, making it difficult to detect them. Furthermore, faces at significant distances may be partially or completely occluded by other objects, such as hats, glasses, or other people, which can further complicate the detection process. Lastly, background noise in the image can also interfere with the face detection process, particularly in crowded environments where multiple objects are present in the image. Overall, detecting faces at large distances requires robust algorithms and techniques that can overcome these challenges.

The report will begin by presenting the background of face detection, including core techniques and the current state-of-the-art work, in section 2. In section 3, we will introduce the dataset that we used to train and evaluate the model for the face detection task specifically for faces in long distances. Section 4 will propose the methods that we employed to perform the face detection task at large distances. We will then provide detailed information on the

experiments conducted and the results obtained in section 5. Finally, we will present the conclusion and future work of the project in sections 6 and 7, respectively.

2 Background

In this section, we will introduce the core techniques involved in face detection, the state-of-art face detection and object detection models, and also the baseline of this project.

2.1 Core Technique of Face Detection

Sliding window The sliding window technique involves dividing an image into a grid of fixed-size windows and applying a classifier or neural network to each window. This approach helps determine if the window contains the object of interest, usually by providing a score or probability. However, it can be computationally expensive when dealing with numerous windows at different scales and aspect ratios. Techniques like image pyramid and scale-space representation are employed to preprocess the image and reduce the number of windows to be evaluated.

Anchor box Anchor boxes are predefined bounding boxes with various scales and aspect ratios that are positioned across the input image. During training, the algorithm learns to adjust these anchor boxes to better fit the ground truth bounding boxes of the faces in the image.

Non-maximum Suppression (NMS) : NMS is a technique used to eliminate redundant bounding boxes and retain only the most accurate and representative detections. It involves sorting predicted bounding boxes by their confidence scores and selecting the one with the highest score as the initial detection. Subsequently, any other predicted bounding boxes with an IoU overlap greater than a certain threshold (e.g., 0.5) are removed. This process continues until no more bounding boxes remain.

Object clustering Object clustering is utilized to group bounding boxes that belong to the same object or instance. Clustering algorithms are applied to group bounding boxes based on their location and appearance features. This technique is beneficial when dealing with closely located or partially occluded objects.

Anchor-free In contrast to anchor boxes, anchor-free methods directly predict the object's bounding box coordinates and class probabilities without relying on predefined anchor boxes. These methods aim to improve the flexibility and accuracy of object detection.

State-of-the-art architecture The state-of-the-art architecture of an object detector comprises three main components: the backbone, neck, and head. The backbone is responsible for extracting features from the input images, often employing a pre-trained model. The neck, situated between the backbone and head, collects feature maps from different stages, facilitating interaction between low-level and high-level features. The head is the final component, which predicts bounding boxes and object classes. Its role is to process the collected features and produce the final detections.

2.2 Related Work

2.2.1 RetinaFace

RetinaFace [2] is an advanced deep learning-based facial detector for Python that provides facial landmarks and follows a backbone, neck, and head structure. The backbone of RetinaFace uses a modified ResNet-based feature extractor that comprises several convolutional layers and residual blocks to extract features from the input image. The neck of RetinaFace includes a Feature Pyramid Network (FPN), which utilizes lateral connections to merge features from different layers of the backbone network and produces a set of feature maps at various scales. These feature maps are then utilized to generate proposals for the face locations in the input image. The head of RetinaFace includes a set of subnetworks for classification and regression, which predict the presence of faces and refine the location and size of the face bounding boxes. Additionally, the head includes a landmark detection subnetwork that predicts the coordinates of facial landmarks such as the eyes, nose, and mouth. However, since RetinaFace is designed for multitasking, including predicting face scores, face boxes, face landmarks, and 3D vertices, we believe that the landmark and 3D vertices would be unnecessary for our task.

2.2.2 YOLO

The YOLO (You Only Look Once) framework has gained recognition for its exceptional combination of speed and accuracy, which allows for quick and reliable object detection in images. Over time, the YOLO family has undergone several iterations, with each new version improving upon the previous ones to overcome limitations and boost performance.

The YOLO family of object detection models has consistently aimed to balance speed and accuracy, striving to deliver real-time performance without compromising detection quality. As the YOLO framework has progressed through its various iterations, this tradeoff has been a recurring theme, with each version optimizing these competing objectives differently. The original YOLO model [3] prioritized high-speed object detection, using a single convolutional neural network (CNN) to predict object locations and classes directly from the input image, achieving real-time processing. However, this focus on speed came at the expense of accuracy, particularly with small or overlapping objects.

Subsequent YOLO versions introduced refinements and enhancements to address these limitations while maintaining real-time capabilities. YOLOv2 [4] introduced anchor boxes and passthrough layers to improve object localization, resulting in higher accuracy. YOLOv3 [5] enhanced performance by using a multi-scale feature extraction architecture, allowing for better detection across various scales.

The tradeoff between speed and accuracy became more nuanced as the YOLO framework evolved. Models like YOLOv4 [6] and YOLOv5 [7] introduced innovations such as new network backbones, improved data augmentation techniques, and optimized training strategies. These developments led to significant accuracy improvements without compromising real-time performance.

The baseline in our project is YOLOv7 Face[8], which is a modified version of YOLOv7 [9] specifically designed for face detection. The model was trained using the WiderFace and yolov7-face-label datasets, and we present the results of the easy, medium, and hard tasks in Table 3.

In contrast, YOLOv8 [10], developed by Ultralytics, is a state-of-the-art model that builds on the success of previous YOLO versions while introducing new features and improvements to enhance performance and flexibility. YOLOv8 is optimized for speed, accuracy, and ease of use, making it an excellent choice for various tasks such as object detection, image segmentation, and image classification.

Table 1: The labels of each image.

Labels	Description
Bounding box	x1,y1,w,h
Blur	clean(0), normal blur(1), heavy blur(2)
Expression	typical expression(0), exaggerate expression(1)
Illumination	extreme(0), occlusion(1)
Invalid	false(0), true(1)
Occlusion	no occlusion(0), partial occlusion(1), heavy occlusion(2)
Pose	typical pose(0), atypical pose(1)

3 Dataset

The dataset we used in this project is the WIDER FACE dataset which is the benchmark dataset for the task of face detection[11]. The dataset includes 32,203 images and labels 393,703 faces in 61 different events covering a large number of scenes in the real world with a high degree of variability in scale, pose, occlusion, etc. The label for each image includes bounding boxes that indicate the position of the face and other attributes like blur, expression, illumination, validation, occlusion, and pose as listed in the table 1. The dataset is randomly splits into 40% of training dataset, 10% of validation dataset, and 50% of testing data. For each dataset, the images are randomly chosen from different events and based on the detection rate, the events are divided into easy, medium, and hard, and we will evaluate based on these levels. WIDER FACE does not release bounding box ground truth for the test images, thus, in this project, the evaluation of baseline and the model we trained is based on the validation dataset.

4 Methodology

In this project, we utilized the state-of-the-art object detection model YOLOv8 to perform the face detection in large distance task. We experimentaly explored three methods as followed and will explain them in detail in the following subsections:

- Fine-tune YOLOv8 for face detection
- Fine-tune YOLOv8 with Transformer for face detection
- Train and evaluate the model on the enhanced images after Super Resolution.

4.1 Fine-tune YOLOv8 for face detection

Yolov8 places the power of AI in everyone's hands. The origin YOLOv8 model does not support the prediction of human face, however, it is easy for us to train the model on our own dataset like Wider Face to perform face detection tasks if we prepare the dataset according to the requirements.

Prepare YOLO style WiderFace. In order to train or fine-tune YOLOv8, we need to prepare the dataset to yolo style. A dataset.yaml file is needed to let YOLO know where our images are, and how many classes the model needs to predict, they will find the label of image automatically if we organize the directories in YOLO style. The bounding box label of Wider Face is different from YOLO's label, thus we need to convert them. For Wider Face the format is (x1,y1,w,h) where (x1,y1) is the top-left corner in pixel coordinate of the bounding box and (w,h) is the width and height of the image, while for YOLO, the format is (x,y,w,h) where (x,y) is the center of the bounding box and the values are normalized to (0,1).

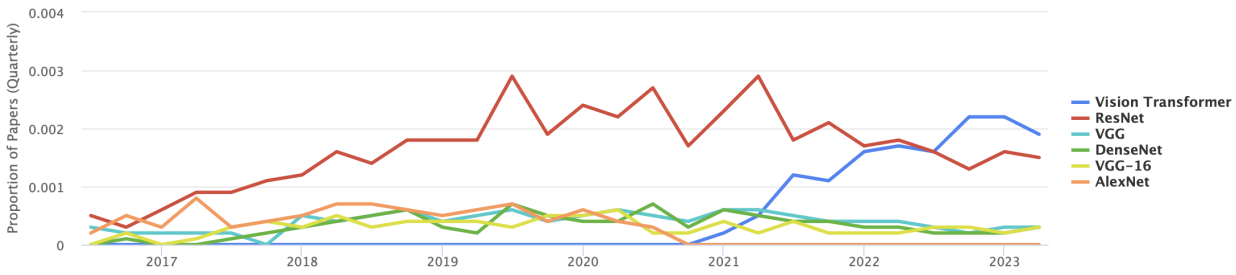

Figure 1: The use of techniques for Computer Vision Papers

4.2 Fine-tune YOLOv8 with Transformer for face detection

While the Transformer architecture has become the standard for natural language processing tasks, its application to computer vision has also shown great promise, achieving excellent results compared to state-of-the-art convolutional networks[12]. Transformers also allow for more efficient computation during training and inference due to their ability to parallelize operations across tokens. This makes them particularly well-suited for large-scale object detection tasks. More and more people are using ViT (Vision Transformer) for Computer Vision tasks (see Figure 1), so we also considered adding a transformer to our task to see if the result improved.

For our project, we made modifications to the original backbone in Yolov8 by adding a transformer layer directly after the feature pyramid, as shown in Figure 2. We believe this approach will yield improved results as it leverages the self-attention mechanism of the transformer layer to identify the salient regions of the input image, which can then be better incorporated into the neck and head structures of the model for more accurate predictions.

4.3 Enhance the resolution of image using Super Resolution

As discussed in section 2, one of the challenges faced in face detection at large distances is the low resolution of faces, making it difficult for the model to identify them as faces. Within the training dataset, approximately 35% (4496 images) were labeled as heavy blur, while in the validation dataset, approximately 36% (1164 images) were labeled as heavy blur. To address this, we conducted Super Resolution processing on the training and validation datasets,

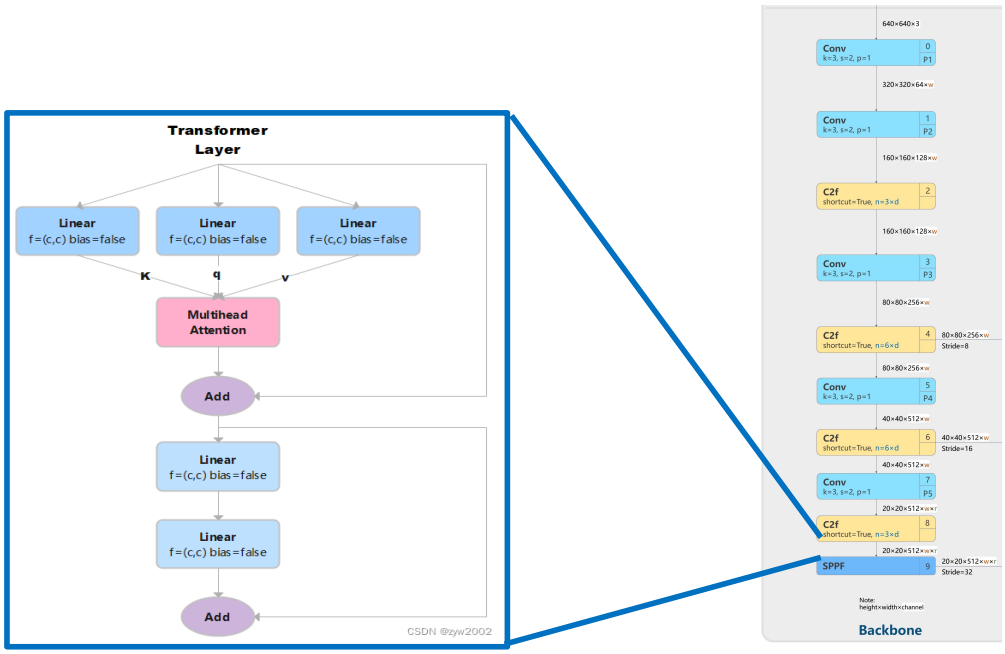

Figure 2: The architecture in our approach. The right part is the backbone of Yolov8, and the left part is the additional transformer layer.

specifically targeting the images labeled as heavy blur. We used the method proposed in the paper[13]. This process was time-consuming, taking over 15 hours to complete.

Our hypothesis was that by utilizing Super Resolution techniques to enhance the resolution of each pixel in the image, the faces at long distances would become clearer, thereby increasing the detection rate. However, during the course of our experiments, we discovered that this approach might not be as helpful as anticipated, given the nature of how Super Resolution works.

In our experiment, the Super Resolution technique involved first detecting the faces in the original blurred image using RetinaFace, and then enhancing the details of the detected faces as shown in figure 3. The remaining portions of the image were considered as background and reconstructed as objects with similar colors and shapes as the shown in the figure 4.

Figure 3: Super Resolution results on failed detected face

The result revealed that the faces that were not initially detected were instead reconstructed as other objects. Our initial expectation was that Super Resolution would enhance the details of heavily blurred faces, making them detectable. However, we now understand that the effectiveness of Super Resolution is dependent on the accuracy of the underlying face detection model, which may not always align with our expectations.

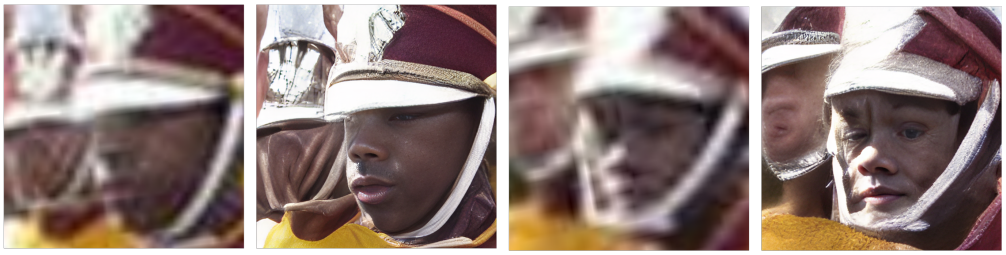

Figure 4: Super Resolution results on failed detected face

4.4 Evaluation

The typical evaluation metric for the task of face detection is Intersection over Union (IoU). The IoU is the metric to evaluate the accuracy of the detection, Given a ground-truth bounding box and a detected bounding box, we compute the IoU as the ratio of the overlap and union areas as shown in the figure 5. The IoU can have any value between 0 and 1. If two boxes do not intersect, the IoU is 0. On the other hand, if they completely overlap, the IoU is 1. Therefore, the higher the IoU, the better the prediction of a face detection system. The threshold of IoU is usually set to 0.5. If the IoU score is above the threshold, the detection is considered a true positive. If the score is below the threshold, the detection is considered a false positive.

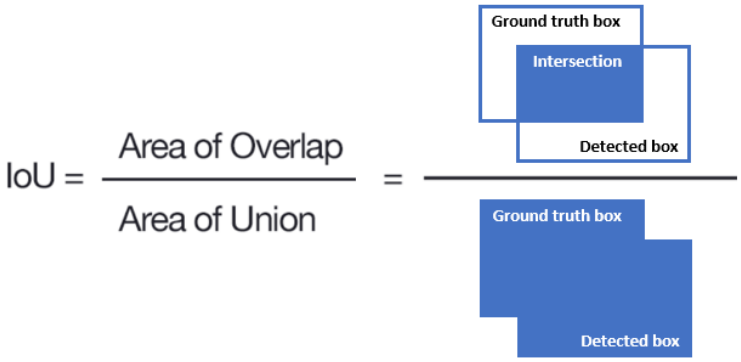

Figure 5: Intersection over Union[14]

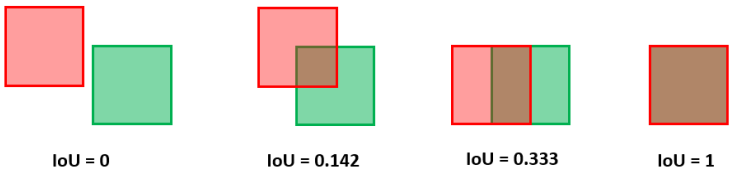

Figure 6: Example of IoU values[14]

5 Experiment

In our experiment, we initially evaluated the performance of the baseline YOLOv7-Face model on the validation dataset using the five pre-trained models provided by the author. Subsequently, we proceeded to train six additional models using the methods described in section 4. The rest of this section will present a comprehensive overview of the results obtained from these experiments.

Baseline evaluation We conducted a baseline evaluation on the validation dataset of Wider Face, which consisted of 3226 images. The evaluation was performed on the five models provided by the authors. Although the authors did not

provide specific details about the training process, based on their code, it can be inferred that the models were trained for 300 epochs using a default input image size of 640x640. The commonly used IoU threshold of 0.5 was employed for evaluation. The results of this evaluation are presented in Table 2. The yolov7 model achieved relatively high results, however, it exhibited longer test times compared to the lite model. Additionally, the yolov7 model had a significantly larger size compared to the lite model.

Table 2: The performance of baseline.

Models	Size	layers	Parameters	Easy	Medium	Hard	Time(ms)
yolov7-lite-t	747KB	236	251795	88.67	85.19	71.44	96.96
yolov7-lite-s	2.4MB	276	1073667	92.67	89.88	78.53	187.12
yolov7-tiny	12.4MB	224	6048051	94.73	92.57	82.09	328.74
yolov7s	8.9MB	297	4269124	94.81	93.13	85.18	383.4
yolov7	73.6MB	315	36562547	96.93	95.56	88.05	1670.07

Fine-tuning YOLOv8 We conducted an experimental fine-tuning of YOLOv8 on the original Wider Face training data for 30 epochs using default training hyperparameters. Stochastic Gradient Descent (SGD) was employed as the optimization algorithm, with a learning rate set to 0.01 and a batch size of 16. The training process took approximately 1.395 hours on an advanced GPU (NVIDIA A100-SXM4-40GB). The resulting model had a size of 6.2MB. The performance of this model is presented in Table 3 as "yolov8(30)". We further fine-tuned YOLOv8 on the enhanced Wider Face training data using the same training methodology as with the original data. The performance of this model is presented in Table 3 as "yolov8-sr(30)".

Fine-tuning YOLOv8 with Transformer layers Additionally, we explored the performance of the YOLOv8 model with Transformer layers. Two models were trained for the modified architecture—one using the original data and another using the enhanced data. The training details remained consistent with 30 epochs and a batch size of 16. The performance of the model with Transformer layers on the original data is presented in Table 3 as "yolov8-tr(30)". Similarly, "yolov8-tr-sr(30)" represents the model trained on the enhanced data.

Table 3 shows the results obtained from the models trained for only 30 epochs, which demonstrate competitive performance compared to our baseline. These models achieved satisfactory results while maintaining an acceptable model size and inference time. Specifically, the "yolov8-sr(30)" and "yolov8-tr-sr(30)" models were evaluated on the enhanced validation dataset. However, the results did not align with our expectations. As discussed in section 4, the Super Resolution technique itself relies on a face detection model. In cases where faces were not initially detected, the technique enhanced them to random objects that were still labeled as faces. This inconsistency may have had a negative impact on the model's performance, as it was unable to learn the correct facial features. Consequently, our hypothesis did not hold true.

Table 3: The performance of models trained for 30 epochs.

Models	Size	layers	Parameters	Easy	Medium	Hard	Time(ms)
yolov8 (30)	6.2MB	225	3011043	93.29	90.83	75.67	69.57
yolov8-sr (30)	6.2MB	225	3011043	93.25	90.92	75.46	67.99
yolov8-tr (30)	11.7MB	246	6070243	92.12	89.92	74.25	306.25
yolov8-tr-sr (30)	11.7MB	246	6070243	92.12	89.71	73.99	304.93

Fine-tuning YOLOv8 with Transformer layers for more epochs The performance of the YOLOv8 model with Transformer layer training by 30 epochs was not as impressive as the model without Transformer layers. One possible reason for this is that the introduction of Transformer layers significantly increased the number of parameters in the model, more than three times that of the original YOLOv8. Additionally, the pre-trained weights used in the original model did not encompass the new Transformer layers. We believe that if we were to train the model for a longer duration, we may achieve improved results. Consequently, we proceeded to train the YOLOv8 model with Transformer layers for 100 and 300 epochs, respectively. To mitigate the risk of overfitting and optimize training time, we utilized early stopping techniques with patience of 50 epochs. The training process took approximately 20 hours, and the model was early stopped at the 276th epoch when no further improvement was observed. The outcomes of these extended training sessions are presented in Table 4.

Table 4: The performance of all models.

Models	Size	layers	Parameters	Easy	Medium	Hard	Time(ms)
yolov7	73.6MB	315	36562547	96.93	95.56	88.05	1670.07
yolov8 (30)	6.2MB	225	3011043	93.29	90.83	75.67	69.57
yolov8-tr (30)	11.7MB	246	6070243	92.12	89.92	74.25	306.25
yolov8-tr (100)	11.7MB	246	6070243	93.24	90.93	75.67	351.28
yolov8-tr (300)	11.7MB	246	6070243	**93.74**	**91.63**	**76.21**	392.60

6 Conclusion

In this project, we aimed to enhance the YOLOv8 architecture by incorporating a transformer layer to improve long-range face detection. Additionally, we explored the effects of applying super resolution to input images on the performance of our model. Interestingly, we discovered that super resolution did not contribute to better results for our specific task. Nevertheless, our proposed model showed a slight improvement over the fine-tuned YOLOv8 in face detection.

7 Future Work

To ensure a fair comparison, we will train the fine-tuned YOLOv8 for 300 epochs before comparing the results. In addition, we plan to augment our model by incorporating more advanced transformer layers after each level of feature pyramid. This will enable us to further improve our model's long-range face detection capabilities and enhance its overall performance.

[1]

References

[1] Yi Yao, Besma Abidi, Nathan D. Kalka, Natalia Schmid, and Mongi Abidi. High magnification and long distance face recognition: Database acquisition, evaluation, and enhancement. In *2006 Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference*, pages 1–6, 2006.

[2] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild, 2019.

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[4] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger, 2016.

[5] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[7] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, (Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, November 2022.

[8] Derron. YOLOv7-Face. 2023.

[9] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022.

[10] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, January 2023.

[11] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[1]The report is revised using ChatGPT. Thank you ChatGPT.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[13] Shangchen Zhou, Kelvin C. K. Chan, Chongyi Li, and Chen Change Loy. Towards robust blind face restoration with codebook lookup transformer, 2022.

[14] Baeldung. Intersection Over Union for Object Detection. May 2023.