Universitat
de les Illes Balears

MASTER'S THESIS

A COMPARATIVE STUDY OF YOLOV5 AND YOLOV7 MODIFICATIONS FOR FACE DETECTION ON A CUSTOM DATASET

Amarachi Chetachi, Anyim

Master's Degree in Intelligent Systems (MUSI)

Specialisation: Computer Vision

Centre for Postgraduate Studies

Academic Year 2022-2023

A COMPARATIVE STUDY OF YOLOV5 AND YOLOV7 MODIFICATIONS FOR FACE DETECTION ON A CUSTOM DATASET

Amarachi Chetachi, Anyim

Master's Thesis

Centre for Postgraduate Studies

University of the Balearic Islands

Academic Year 2022-2023

Key words:

YOLOv5face, YOLOv7face, Keypoint Detection, Face Detection

Thesis Supervisor's Name: Ortiz Rodríguez, Alberto

A Comparative Study of YOLOv5 and YOLOv7 Modifications for Face Detection on a Custom Dataset

Amarachi Chetachi, Anyim
Tutor: Ortiz Rodríguez, Alberto
Trabajo de fin de Máster Universitario en Sistemas Inteligentes (MUSI)
Universitat de les Illes Balears
07122 Palma, Illes Balears, Spain
amarachi.anyim00@gmail.com

ABSTRACT

Face detection is a fundamental task in computer vision with applications spanning facial recognition, pose estimation, and human-robot interaction. This thesis presents a comprehensive comparative study of two modified versions of the YOLO (You Only Look Once) algorithm, YOLOv5face and YOLOv7face, tailored for landmark detection on a custom dataset of human faces. The study evaluates these models on various aspects, including architecture, accuracy, speed, generalization capability, and specific features.

YOLOv5face strikes a balance between accuracy and speed, rendering it suitable for real-time or near-real-time applications. Equipped with a landmark regression head, it excels in tasks requiring precise facial landmark detection. YOLOv7face, on the other hand, outperforms YOLOv5face in accuracy, even in challenging conditions like occlusion and varying lighting. Its robustness positions it as a reliable choice for real-world applications.

The comparative analysis underscores the importance of selecting the right model based on specific requirements. YOLOv5face offers efficiency and versatility, while YOLOv7face prioritizes accuracy and robustness. Future research directions include diversifying datasets, fine-tuning, real-world testing, efficiency improvements, and applications in human-robot interaction.

This study contributes to the advancement of facial keypoint detection algorithms and guides researchers and practitioners in choosing appropriate models for various computer vision tasks.

***Index Terms*—YOLOv5face, YOLOv7face, Keypoint Detection, Face Detection**

I. INTRODUCTION

A. Background and motivation

Object detection is an important computer vision task that has seen remarkable progress in recent years. YOLO (You Only Look Once) is a widely used object detection algorithm that has achieved state-of-the-art results in many tasks. YOLOv5 and YOLOv7 are relatively recent versions of the YOLO algorithm that have introduced significant improvements in performance and speed. In this thesis, we propose to compare YOLOv5face and YOLOv7face, which are YOLOv5 and YOLOv7 modifications for landmark/keypoint detection on a custom dataset, e.g. comprising human faces, mostly because of the availability of labelled datasets.

Keypoint detection is a specific task in object detection that involves identifying key points on an object, such as corners, edges, and other distinctive features. Keypoint detection is crucial in applications related to face recognition, and also for pose estimation, and even in robotics when we refer to tasks involving Human-Robot Interaction (HRI) [41].

In this thesis, under the motivation of comparing the performance of YOLOv5face and YOLOv7face for keypoint detection on a custom dataset, we will explore the modifications to the backbone network, the loss function, and the training strategies with regard to, respectively, YOLOv5 and YOLOv7. We will also compare in terms of accuracy the adapted versions of YOLOv5 and YOLOv7 with other existing state-of-the-art keypoint detection algorithms on the same custom dataset.

B. Research objectives

The main goal of this work is to study comparatively several keypoint detection algorithms. To be more precise, the goals are:

1) To select a dataset of labelled faces, so that we can obtain quantitative performance data..
2) To find implementations of the YOLOv5face and YOLOv7face architectures and training strategies, so that we can deploy them in a suitable way for face detection evaluation on the chosen dataset.
3) To compare the performance on a specific dataset of YOLOv5face and YOLOv7face with existing state-of-the-art face detection algorithms on the chosen dataset.
4) To evaluate the impact of the different modifications performed over YOLOv5 and YOLOv7 with respect to their performance in face detection tasks.

C. Significance of the study

The significance of this study lies in its potential to contribute to the advancement in keypoint detection algorithms, specifically in the context of facial landmark detection. By comparing the performance of YOLOv5face and YOLOv7face with other existing state-of-the-art algorithms, we aim to provide insights into the strengths and weaknesses of these modified YOLO versions. This knowledge can guide researchers and practitioners in choosing appropriate algorithms for keypoint detection tasks, especially in scenarios involving human faces.

Additionally, comparing YOLOv5face and YOLOv7face to other algorithms might help identify new techniques for improving keypoint detection accuracy and speed. These findings could have broader implications for object detection tasks beyond facial landmarks. Moreover, advancements in keypoint detection can potentially benefit various applications, such as pose estimation and object tracking.

D. Scope and limitations

This study focuses specifically on comparing the performance of YOLOv5face and YOLOv7face for face detection on a custom dataset of human faces. The evaluation will include metrics related to accuracy and speed. However, it is important to note that the scope of this study is limited to facial detection and does not cover all possible object detection scenarios.

The limitations of this study include potential biases in the custom dataset, variations in lighting conditions, facial expressions, and poses that might affect the accuracy of the algorithms. Additionally, the modifications made to YOLOv5 and YOLOv7 are specific to the task of face detection and may not be directly applicable to other object detection issues.

In the following sections, we will delve into a comprehensive analysis of the literature related to face keypoint detection algorithms, including the evolution of the YOLO algorithm, the architectural details of YOLOv5face and YOLOv7face, the methodologies employed in data collection and model training, the comparative analysis of different algorithms, and the presentation and discussion of results. The insights gained from this study have the potential to inform future developments in face keypoint detection algorithms and their application in computer vision tasks.

II. Literature Review

A. Overview of keypoint detection algorithms

Keypoints detectors, also commonly referred to as interest points detectors, are a class of algorithms that identify points in an image that are likely to be distinctive and useful for a given task. These points are then used to extract features that can be used for further processing, such as object detection or image registration [1, 19, 30].

The Harris corner detector is a simple and efficient keypoint detector that was introduced by Chris Harris and Mike Stephens in 1988. The Harris corner detector works by calculating the local intensity gradients in an image and then identifying points where the gradients are both large and correspond to a certain degree of curvature [20].

One of the most well-known interest point detectors is available through the SIFT (Scale-Invariant Feature Transform) algorithm, introduced by David Lowe in 1999. SIFT is a scale-invariant algorithm, which means that it can identify interest points that are at different scales in the image. It is also rotation-invariant, which means that it can identify interest points at different orientations in the image [13, 17].

Another popular interest point detector is given by the SURF (Speeded Up Robust Features) algorithm, introduced by Herbert Bay et al. in 2004. SURF is a faster version of SIFT that is also scale-invariant and rotation-invariant [3, 13].

The FAST corner detector is a very fast and efficient keypoint detector that was introduced by Edward Rosten and Tom Drummond in 2006. The FAST corner detector works by comparing the intensity of a pixel to the intensities of its eight neighbors. If the pixel is significantly brighter or darker than its neighbors, then it is considered to be a corner [38].

The ORB detector is a combination of the FAST corner detector and the BRIEF descriptor. The ORB detector is fast and efficient, and it is also fairly robust to noise and other distortions [13].

In recent years, there has been a growing interest in using deep learning techniques for keypoint detection. Deep learning-based keypoint detectors are able to learn features that are more robust to noise and other distortions than traditional interest point detectors [2, 6, 7].

Within the specific domain of facial keypoint detection, [42] describes a Multi-Task cascaded Convolutional Neural Network (MTCNN) that can be used to detect facial keypoints. It is a popular solution that is known for its accuracy and robustness to noise and changes in illumination. MTCNN works by first detecting candidate facial regions in the image. It then classifies each candidate region as a face or not-a-face. Finally, it refines the location of the facial keypoints in the face regions that have been classified as faces [42].

Face Alignment with Expanded Local Binary Patterns (ELBP-FA) is a deep learning-based facial keypoint detection algorithm that uses LBP features [10]. LBP features are local binary patterns, which are a type of image descriptor that is used to represent the texture of an image. ELBP-FA is robust to noise and changes in illumination, and it has been shown to be effective in detecting facial keypoints in a variety of conditions.

The evolution of facial keypoint detection algorithms has been driven by the need for more accurate, robust, and efficient algorithms [11, 12]. The algorithms that are used today are capable of detecting facial keypoints under a variety of conditions, including noise, changes in illumination, and pose variations. They are also fast enough to be used in real-time applications.

Dlib is a popular open-source library that can be used for facial keypoint detection. Dlib uses a variety of machine learning techniques, including deep learning methodologies. Dlib is known for its accuracy and speed, so that it becomes a suitable choice for facial keypoint detection in a variety of applications [15, 28].

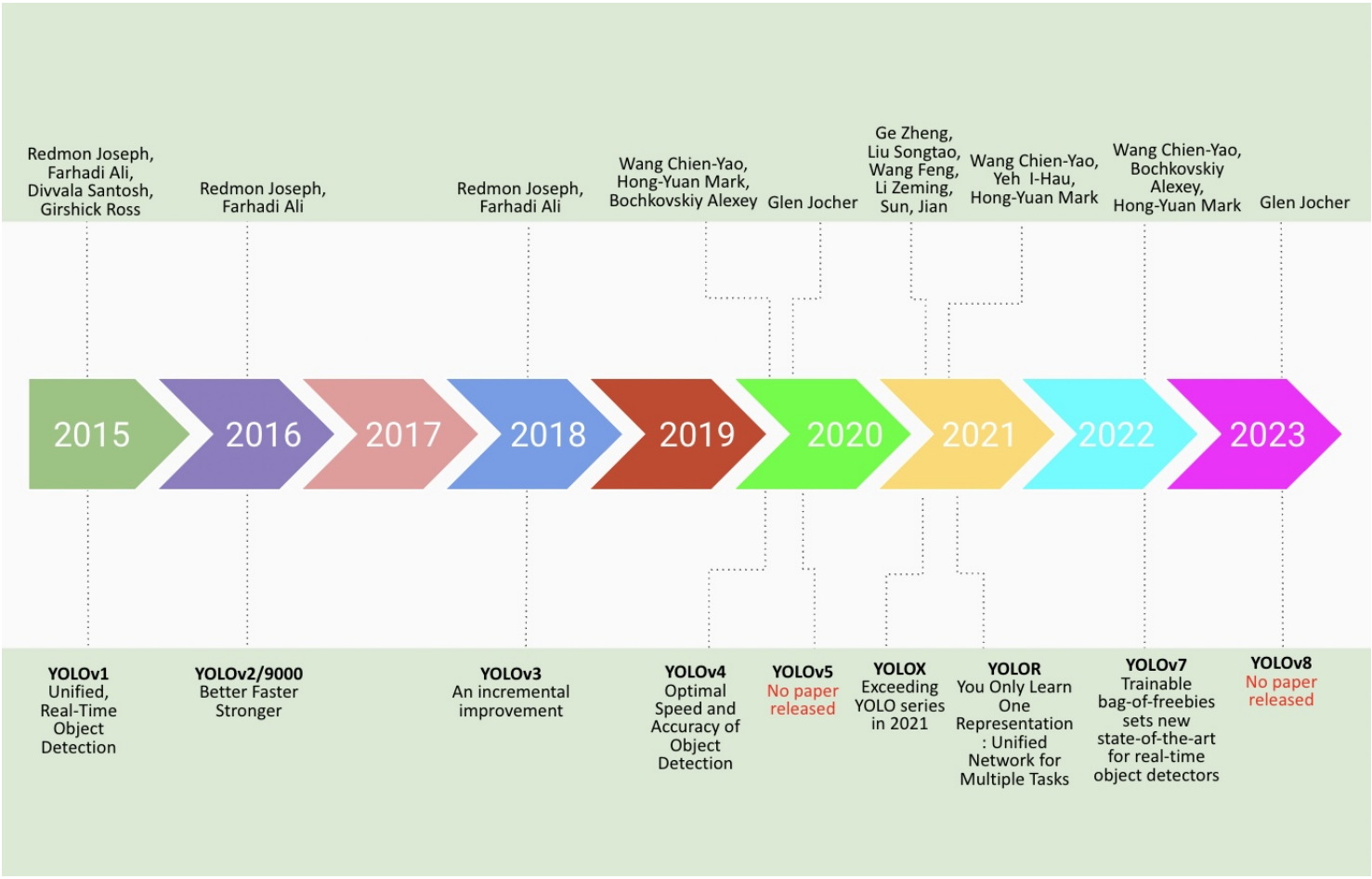

Figure 1. YOLO Timeline from 2015 to 2023 (updated version of the YOLO timeline by Zoumana Keita, https://www.datacamp.com/blog/yolo-object-detection-explained)

B. Evolution of YOLO

Yolo is a generic object detector, as R-CNN and Faster R-CNN. The R-CNN (Region-based Convolutional Neural Network) algorithm was introduced by Ross Girshick et al. in 2014. It is a two-stage algorithm that first proposes regions of interest in the image and then classifies each region as containing an object of interest or not [9]. The Faster R-CNN algorithm, introduced by Shaoqing Ren et al. in 2015, is a faster version of R-CNN that uses a region proposal network to generate the regions of interest [27].

On the other side, Darknet, proposed by Joseph Redmon, is a neural network framework that is used for training and implementing deep neural networks [23, 24]. It is written in C and CUDA, which makes it highly efficient for running on both CPU and GPU hardware. It includes pre-trained models for object detection and classification. Darknet provides the backbone for implementing YOLO [18, 31, 36].

YOLO is the first of a series YOLO algorithms. It uses a single neural network to predict the class and location of objects in an image, making it faster and more efficient than other object detection algorithms (this is why they are called one-shot detectors). It has 24 convolutional layers followed by 2 fully connected layers [24].

YOLOv2 was introduced in 2016. It improved on the original YOLO algorithm with the use of anchor boxes to improve the accuracy of bounding box predictions. The idea behind anchor boxes is to pre-define a set of bounding boxes with different sizes and aspect ratios, to be placed at various positions across the image, which are then used as a reference during the object detection process. Another improvement was a feature extraction network based on residual blocks [25].

YOLOv3 improved on YOLOv2 with the use of multi-scale feature maps to improve the detection of small objects, a more complex backbone network based on Darknet-53, and improved training techniques such as data augmentation and batch normalization [26]. Redmond quit computer vision research after this paper due to some ethical issues.

Created by Alexey Bochkovskiy et al, the improvement in YOLOv4 included features such as the use of spatial pyramid pooling (SPP) to capture features at multiple scales, improved anchor box design, and advanced training techniques such as CutMix and the Mish activation function [4].

YOLOv5 was created by Glenn Jocher, Founder and CEO at Ultralytics. The improvements in this version include: a dynamic architecture (makes it easier to customize the model for different use cases), an efficient backbone (a modified EfficientNet backbone), improved training techniques (AutoML), and better performance on small objects [33]. There

is currently a YOLOv8 from the same authors.

YOLOX is based on YOLOv3. It was introduced by Megvii Technology and also includes some improvements: a decoupled backbone and head (meaning that feature extraction and detection are separated), a cross-stage partial network, ability to adjust the network size to achieve different trade-offs between speed and accuracy, and better performance in general [8]. The use of the YOLOX-Nano backbone, the Feature Pyramid Network (FPN) neck, the head, and the anchor-free detection mechanism are some of the key features that make it popular for object detection.

YOLOR paper proposes a unified network for multiple tasks that integrates implicit and explicit knowledge to generate a general representation that can be used for various tasks. The network combines compressive sensing and deep learning and is based on previous work that uses sparse coding to reconstruct feature maps of a CNN. The proposed network is shown to improve model performance with a very small amount of additional cost. The paper also discusses three different ways for modeling implicit knowledge. [35]

YOLOv7 backbones do not use ImageNet pre-trained backbones. Rather, the models are trained using the COCO dataset entirely. Nevertheless, some similarities with previous versions can be expected because YOLOv7 is written by the same authors as Scaled YOLOv4, which is an extension of YOLOv4. E-ELAN (Extended Efficient Layer Aggregation Network) is the computational block in the YOLOv7 backbone [34]. It also used Scaling for Concatenation-based Models to increase the depth, resolution of an image, and the width of the model.

Figure 1 presents a chronological overview of the evolutionary progression of YOLO algorithms, spanning the years from 2015 to 2023 [14].

C. Overview of YOLOv5face and YOLOv7face

1) YOLOv5face: The YOLOv5face paper presents a novel approach to face detection using the YOLOv5 object detector. The authors of the paper designed two super light-weight models based on ShuffleNetV2, which are optimized for embedded or mobile devices. The YOLOv5face models are capable of achieving state-of-the-art performance on the Wider Face validation dataset, including the Easy, Medium, and Hard subsets [22].

The YOLOv5face paper presents several modifications made to the YOLOv5 object detector to optimize it for face detection. These modifications include changes to the network architecture, the introduction of a landmark regression head, and modifications to the loss function:

1) Network Architecture: The YOLOv5face detector uses the YOLOv5 object detector as its baseline and optimizes it for face detection. The network architecture of YOLOv5face consists of backbone, neck, and head. The backbone is based on the CSPNet design used in YOLOv5, while the neck uses an SPP and a PAN to aggregate features. The head includes both regression and classification layers.
2) Stem Block: The authors of the paper experimented with the use of a *stem* block versus a *focus layer* in the network architecture. They found that using a stem block improved the mAP by 0.57%, 0.33%, and 0.23% on the easy, medium, and hard subsets, respectively.
3) SPP with Smaller Size Kernels: The authors also experimented with the use of SPP with smaller size kernels. They found that using smaller size kernels did not affect performance significantly.
4) Landmark Regression Head: One of the key modifications made to YOLOv5 to create YOLOv5face is the addition of a five-point landmark regression head. This head allows for the detection of facial landmarks, which can improve the accuracy of face detection and alignment. The landmark outputs can be used to align face images before they are sent to the face recognition network.
5) Loss Function: The authors of the paper modified the loss function used in YOLOv5 to optimize it for face detection. They introduced a new loss term for landmark regression, which penalizes the distance between predicted and ground-truth landmarks. They also modified the classification loss term to include a focal loss term, which helps to address class imbalance in the dataset.

The architecture of the YOLOv5face face detection network is illustrated in Figure 2, comprising three main components: the backbone, the neck, and the head [22]. In YOLOv5, a newly designed backbone called CSPNet is employed. Within the neck section, the authors utilize an SPP (Spatial Pyramid Pooling) and a PAN (Path Aggregation Network) to aggregate features effectively. The head of the network employs both regression and classification components.

Figure 2(a) provides an overview of the entire network architecture. Figure 2(b) introduces a critical building block known as CBS (Convolutional Block Structure), consisting of a Convolutional layer, Batch Normalization layer, and the SILU activation function. This CBS block finds application in various other blocks throughout the network.

In Figure 2(c), the authors present the output label format for the head, encompassing bounding box (bbox) information, confidence scores (conf), classification labels (cls), and five-point facial landmarks. The inclusion of landmarks is a unique feature that transforms YOLOv5 into a face detector with landmark prediction. If landmarks are excluded, the last dimension, initially 16, should be reduced to 6. It is important to note that the specified output dimensions e.g. $80 \times 80 \times 16$ in P3 or $40 \times 40 \times 16$ in P4, apply to each anchor, and the actual dimensions need to be multiplied by the number of anchors.

Figure 2(d) illustrates the Stem structure, which replaces the original Focus layer in YOLOv5. The integration of the Stem block into YOLOv5 for face detection represents one of the innovative contributions.

Figure 2(e) showcases the CSP block (also known as C3), which takes inspiration from DenseNet. However, instead of directly summing the full input with the output after some CNN layers, the input is split into two halves. One half passes through a CBS block, a series of Bottleneck blocks depicted in Figure 2(f), followed by another Convolutional layer. The other half undergoes a Convolutional layer, and then both halves are concatenated, followed by another CBS block.

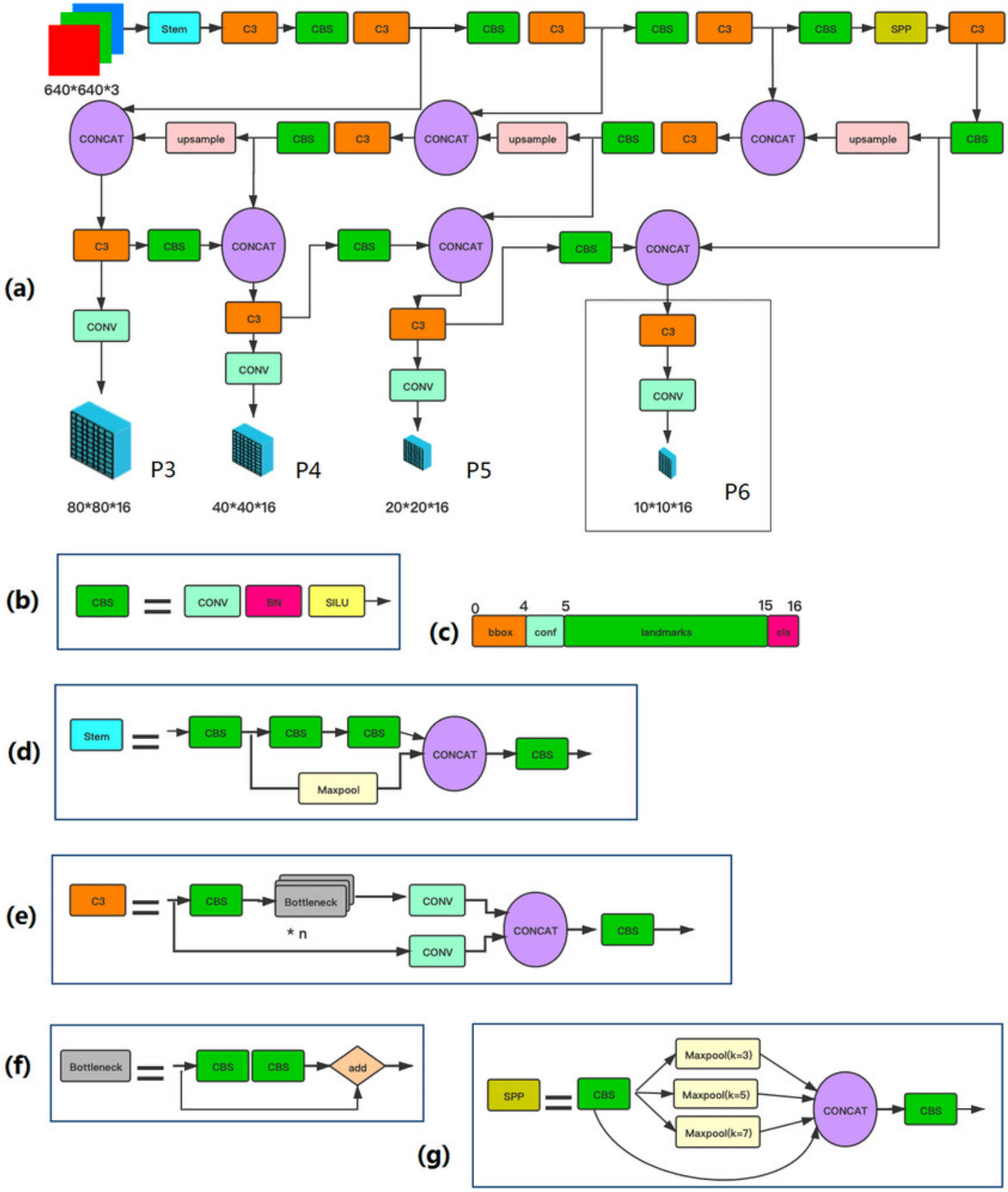

Figure 2. YOLOv5face detector architecture: a) overall view, b) CBS block, c) format of P3-P6 output, d) STEM block, e) CSP (C3) block, f) bottleneck block, g) SPP block.

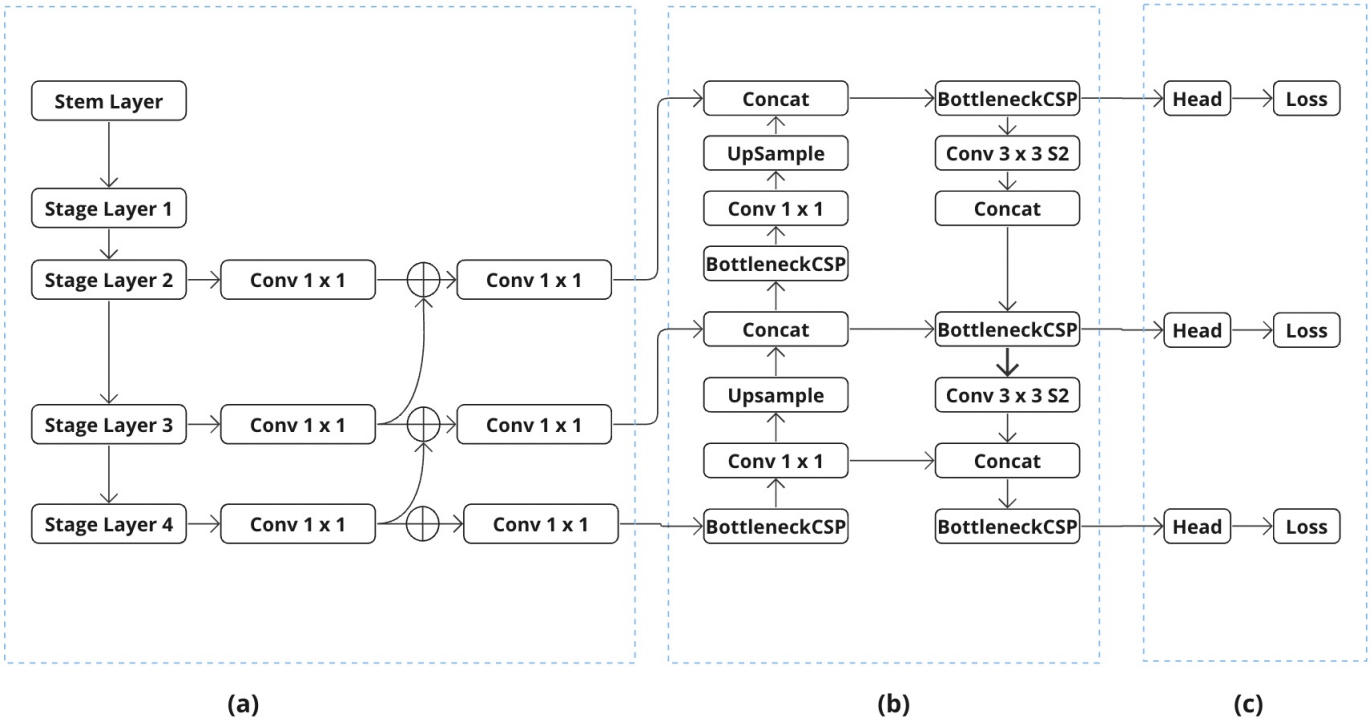

Figure 3. YOLOv7face detector architecture: a) ResNet-v1d Backbone, b) PANet Neck, d) Head.

Figure 2(g) introduces the SPP block, where the kernel sizes 13×13, 9×9 and 5×5 of YOLOv5 respectively become 7×7, 5×5 and 3×3 in YOLOv5face. This adjustment has been identified as one of the innovations contributing to improved face detection performance.

Overall, the modifications made to YOLOv5 to create YOLOv5Face are designed to optimize the network for face detection. The addition of a landmark regression head and modifications to the loss function help to improve the accuracy of face detection and alignment, while changes to the network architecture help to optimize the network for real-time face detection applications.

2) YOLOv7face: YOLOv7face is a modified version of the YOLOv7 object detection algorithm that is specifically designed for face detection. It was released in September 2022 [21]. YOLOv7face makes a number of changes to the YOLOv7 algorithm to improve its performance for face detection, including:

1) Incorporation of a ResNet-v1d backbone network
2) Replacement of the neck network with a PANet neck
3) Addition of a new head to the network that predicts the keypoints for each face that is detected. The keypoint head is a lightweight network that is trained on a dataset of facial keypoint annotations. The keypoint head takes the output of the detection head as input and predicts the keypoints for each face that is detected.
4) Use of a new loss function for the keypoint head that is specifically designed to optimize the accuracy of the keypoint predictions. The new loss function for the keypoint head is a combination of two loss functions: a cross-entropy loss and a smooth L1 loss. The cross-entropy loss is used to optimize the classification of the keypoints, while the smooth L1 loss is used to optimize the regression of the keypoints.
5) Incorporation of a new post-processing step to refine the keypoint predictions. The new post-processing step for the keypoint predictions uses a Gaussian filter to smooth the keypoint predictions. This helps to reduce the noise in the keypoint predictions and improve their accuracy.

Part of the architecture of the YOLOv7face face detection network is illustrated in Figure 3, reflecting the modifications listed above: addition of a the ResNet-v1d backbone and replacement of the neck in YOLOv7 with a PANet neck [5, 16, 33, 40].

These changes enable YOLOv7face to detect facial keypoints with high accuracy, even in challenging conditions such as low light and occlusion.

III. Methodology

A. Data collection and processing

The Wider Face dataset is a large-scale face detection benchmark that was introduced in a research paper by Shuo Yang et al [39]. The dataset was created to address the limitations of existing face detection benchmarks, which typically contain only a few thousand faces with limited variations in pose, scale, facial expression, occlusion, and background clutters.

The Wider Face dataset, on the other hand, consists of 32,203 images with 393,703 labeled faces, making it 10 times

larger than the current largest face detection dataset. The faces in the dataset vary largely in appearance, pose, and scale, as shown in Figure 4. In order to quantify different types of errors, the dataset includes annotations for multiple attributes, including occlusion, pose, and event categories, which allows for in-depth analysis of existing algorithms.

The dataset is divided into three subsets: training, validation, and testing, represented as WIDER_train, WIDER_val, and WIDER_test. The training set contains 12,880 images with 158,446 labeled faces, while the validation set contains 3,226 images with 40,282 labeled faces. The testing set contains 16,097 images with no annotations.

The Wider Face dataset has become a popular benchmark for face detection research, and has been used to evaluate the performance of many state-of-the-art face detection algorithms. The dataset has also been used to study the impact of different factors on face detection performance, such as the effect of occlusion, pose, and scale. This can be seen in Figure 4. Overall, the Wider Face dataset has significantly contributed to the advancement of face detection research and has helped to bridge the gap between current face detection performance and real-world requirements.

B. Training data requirements

To adapt the dataset for use with YOLO-based models, the provided labels, i.e. the ground truth data, were converted into YOLO format, as the original dataset contains information about object bounding boxes in a specific proprietary format. This information was transformed so as to include the object class index, the normalized coordinates of the bounding box center, and its width and height relative to the image dimensions.

The label conversion process was performed separately for the training, validation, and test subsets, resulting in separate ground truth files for each subset.

The downloaded files and folders are organized as follows:

```
./Widerface
    - WIDER_test/
        - images/
            - 0–Parade/
            - ...
        - labels.txt
    - WIDER_train/
        - images/
            - 0–Parade/
            - ...
        - labels.txt
    - WIDER_val/
        - images/
            - 0–Parade/
            - ...
        - labels.txt
    - ground_truth/
        - wider_easy_val.mat
        - wider_medium_val.mat
        - wider_hard_val.mat
        - wider_face_val.mat
```

As mentioned in the previous section, the images have been split in accordance with the ratios 40:10:50 for the training, validation and testing sets. The training set was used to teach the model, the validation set for hyperparameter tuning and monitoring, and the testing set for the final assessment of the models.

C. Model training

In this section, we delve into the critical aspect of model training, a fundamental step in the development of keypoint detection models based on YOLOv5 and YOLOv7 modifications. Proper training is pivotal for achieving high-performance results in custom datasets. The training process involves several key components, including dataset processing, hyperparameter tuning, and optimization techniques.

On execution of the code for model training, the custom dataset was subjected to meticulous preprocessing. This step includes data augmentation techniques such as mosaic, rotation, scaling, and flipping to enhance model robustness and reduce overfitting. Additionally, data normalization was applied to standardize pixel values, ensuring consistent input to the models.

The choice of batch size affects training efficiency and memory usage. Batch sizes of 16 and 32 were used for the YOLOv5face and YOLOv7face respectively. Regarding training epochs, we have employed 250 epochs for YOLOv5face and 300 for YOLOv7face. Early stopping was employed to monitor validation loss, terminating training when improvements plateaued. On the other side, we have kept the same custom loss function tailored for keypoint detection implemented by the developers, given the criticality of this component for the training of any neural model. Transfer learning was implemented in each of YOLOv5face and YOLOv7face by respectively initializing the model weights with YOLOv5s and YOLOv7-tiny-face pre-trained models.

Model training was performed on an Apple M2 fitted with an 8-core CPU, 10-core GPU, 8GB unified memory, 512GB storage. Training progress was regularly monitored and logged for analysis. Checkpoints were saved periodically to allow for resuming training from the most recent point in case of interruption.

In summary, model training, as a multifacted process, has involved careful data processing, hyperparameter tuning and optimizaction stages. During training, its effectiveness has signifitcantly impacted the performance of YOLOv5face and YOLOv7face in keypoint detection on the dataset.

D. Evaluation metrics and performance measures

In the context of face detection, the terms *easy*, *medium* and *hard* are often used to describe different subsets of a dataset based on the difficulty of detecting faces within the images. These subsets help evaluate the performance of face detection algorithms under varying conditions and challenges [29]. We explain in the following what each term typically means:

Figure 4. Examples of images of the Wider Face dataset depicting variations in scale, pose, occlusion, expression, makeup, and illumination.

1) Easy Dataset: Images in the *easy* subset of a face detection dataset typically contain faces that are well-lit, well-posed, and easily distinguishable from the background. Faces in easy images may have minimal occlusions (objects covering part of the face) or variations in scale, rotation, or facial expressions. The background of the images is usually simple and uncluttered, making it relatively straightforward for a face detection algorithm to locate and identify faces.
2) Medium Dataset: The *medium* subset of a face detection dataset includes images with a moderate level of difficulty. Faces in medium images may have some occlusions, variations in lighting conditions, moderate pose variations, or facial expressions. The background in medium images might be more cluttered or contain distractions, making it somewhat challenging for a face detection algorithm.
3) Hard Dataset: Images in the *hard* subset are the most challenging for face detection algorithms. Faces in hard images may have significant occlusions, extreme lighting conditions (e.g., strong shadows or overexposure), substantial pose variations, or complex facial expressions. The background in hard images may be highly cluttered or contain multiple faces in close proximity, making it difficult for a face detection algorithm to accurately locate and distinguish individual faces.

Researchers and practitioners use these subsets to assess the robustness and performance of face detection algorithms in real-world scenarios. Algorithms that perform well across all subsets, including the hard dataset, are considered more reliable and suitable for practical applications where face detection can be challenging, such as surveillance, image analysis, or facial recognition systems under various lighting and environmental conditions.

The evaluation process has considered both quantitative and qualitative aspects, including the visual inspection of predictions as seen in Figure 5, in Figure 6 and in Figure 7.

Model performance has been quantitatively assessed by means of the *mean Average Precision (mAP)*, the *localization error* of the bounding boxes for the highest overlapping pair, error in the width and height predictions and the *average time taken to predict* (the average inference time) across each of the *easy*, *medium* and *hard* subsets.

The mAP is the current benchmark metric used by the computer vision research community to evaluate the robustness of object detection models. This metric is defined in terms of *precision* (P) and *recall* (R). The *precision* is calculated as the ratio of true positives TP against the total positive predictions, i.e. TP + FP, while the *recall* is calculated as the ratio of true positives TP versus the total of true positives, i.e. TP + FN. Hence, P is affected by false positives, i.e. wrong positive predictions, while R is affected by false negatives, i.e. missing positives.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

The *Average Precision* (AP) and the mAP encapsulate the tradeoff between precision and recall as shown in the following equations: (the approximation of AP reproduced next is one among several others suggested in the literature, in connection with the PASCAL VOC challenge)

$$AP = \int_0^1 P(R)dR \approx \sum_{R=0:0.1:1} P(R)/11$$

$$mAP = \sum_c AP(c)$$

In the context of object detection evaluation, positives correspond to detections with an *Intersection over Union* (IoU) above 0.5, where the IoU is defined as follows:

$$IoU(\mathrm{bb}_p) = \frac{\mathrm{bb}_p \cap \mathrm{bb}_{gt}}{\mathrm{bb}_p \cup \mathrm{bb}_{gt}}$$

where bb_p is a predicted bounding box, bb_{gt} is the bounding box from the ground truth with the largest overlap, and $\cap$

and $\cup$ mean, respectively, the intersection and the union of bounding boxes.

On the other side, the *localization error* measures the accuracy in object detection tasks. It is calculated as the distance between the center points of the predicted bounding box and the true bounding box for the pair with the highest overlap. This error quantifies how well the predicted bounding box aligns with the ground truth, providing insight into the precision of object localization.

Errors in *width and height prediction* in object detection tasks are also crucial metrics because they directly impact the accuracy and reliability of the detected bounding boxes. Large errors can indicate that the model is failing to accurately predict the size and location of objects in an image. They can also significantly impact the IoU score

Inference time is another relevant metric for assessing the models' performance. It is the time required by the model to process an input image and produce predictions. Assessing inference time helps determine whether the model can operate efficiently in practical scenarios. Models with faster inference times are better suited for applications where quick responses are required, as they can process images in near-real-time or real-time. Therefore, inference time is a critical performance metric when evaluating the suitability of a face detection model for specific use cases.

IV. Comparative Analysis

In this section, we will delve into the comparative analysis of YOLOv5face and YOLOv7face modifications for face detection on the custom dataset. As already mentioned, we evaluate various aspects of both models including the qualitative evaluation of models' predictions, the mAP, the localization and size errors, the inference speed, the properties of the architecture, the generalization capability, and specific features and advancements [32].

A. Qualitative evaluation of YOLOv5face and YOLOv7face

We next comment at a qualitative level the performance observed from YOLOv5face and YOLOv7face. For a start, with regard to the *easy* dataset, Figure 5(a) and (k) shows the detection of faces on the same image of a swimmer with YOLOv5face and YOLOv7face respectively. In Figure 5(a), the YOLOv5face wrongly predicts the hand of the swimmer as a face, while YOLOv7face is unable to detect any face on the particular image. In Figure 5(b), YOLOv5face detects the reflection as another face, but in Figure 5(l) the YOLOv7face is able to demonstrate the capacity to differentiate between genuine human faces and reflections in water surfaces, thus avoiding false positives. In Figure 5(d) and (g) the YOLOv5face wrongly identifies the shoulder of the little boy and the camera as faces in the images. Figure 5(h) and (i) show the YOLOv5face detection on people queuing to vote and spectators in an ice hockey game. The YOLOv7face is able to identify more faces as seen in Figure 5(r) and 5(s)

Now regarding the *medium* dataset, in Figure 6(a), YOLOv5face is unable to detect a face whereas YOLOv7face identifies three faces from the people having a picnic in Figure 6(k). In Figure 6(b), YOLOv5face detects only one face, and surprisingly YOLOv7face does not identify any faces in Figure 6(l). YOLOv5face does not seem to work well from far away and under direct sunlight, as it fails to identify any faces in Figure 6(d), while YOLOv7face identifies a good number as seen in Figure 6(n). As seen in Figure 6(e), YOLOv5face does not do well on blurred images when compared with YOLOv7face in Figure 6(o). YOLOv7face does better in detecting faces in the background, as shown in Figure 6(q), compared to YOLOv5face in Figure 6(g).

Finally, in the *hard* dataset, from Figure 7(a) through (e) and Figure 7(k) through (o), we illustrate gatherings of people demonstrating or protesting. While both models do a good job of identifying faces in the crowd, YOLOv7face is significantly better than YOLOv5face. Figure 7(f), (p), (g), (q), (h) and (r) show accident scenes with people there and the detections from YOLOv5face and YOLOv7face look similar, although upon closer inspection, YOLOv7face is capable of more useful detections, including *not-face* detections. Figure 7(i), (j), (s) and (t) show images from concerts under varied illumination conditions and YOLOv7face is able to detect more faces than YOLOv5face.

B. Detection performance of YOLOv5face and YOLOv7face

Table I
YOLOv5face and YOLOv7face Face Detection mAP

Model	Easy mAP	Medium mAP	Hard mAP
YOLOv5face	0.75	0.71	0.44
YOLOv7face	0.93	0.91	0.83

Table I and Figure 8 present a comparative analysis of the YOLOv5face and YOLOv7face models' performance in terms of mAP categorized into the three subsets *easy*, *medium* and *hard*. The corresponding mAP scores provide insights into how well each model handles different scenarios.

It is noteworthy that YOLOv7face consistently outperforms YOLOv5face across all difficulty levels. This superior performance, particularly in the *hard* subset, suggests that YOLOv7face has a remarkable ability to detect faces accurately, even in challenging conditions. The table underscores the advancements made by YOLOv7face, positioning it as a promising choice for high-precision facial keypoint detection tasks, such as face recognition and expression analysis. This heightened level of accuracy can be attributed to the robust architecture and dedicated keypoint head of the model.

C. Localization error

As mentioned before, localization error measures the accuracy of predicting the bounding boxes that encapsulate detected faces compared to the ground truth bounding boxes. The localization error results provide valuable insights into the precision and accuracy of the models in describing the exact location and extent of detected faces.

The mean localization error of YOLOv5face for the *easy*, *medium*, and *hard* datasets are 6px, 1px, and 0px respectively, while the mean localization error of YOLOv7face for the *easy,*

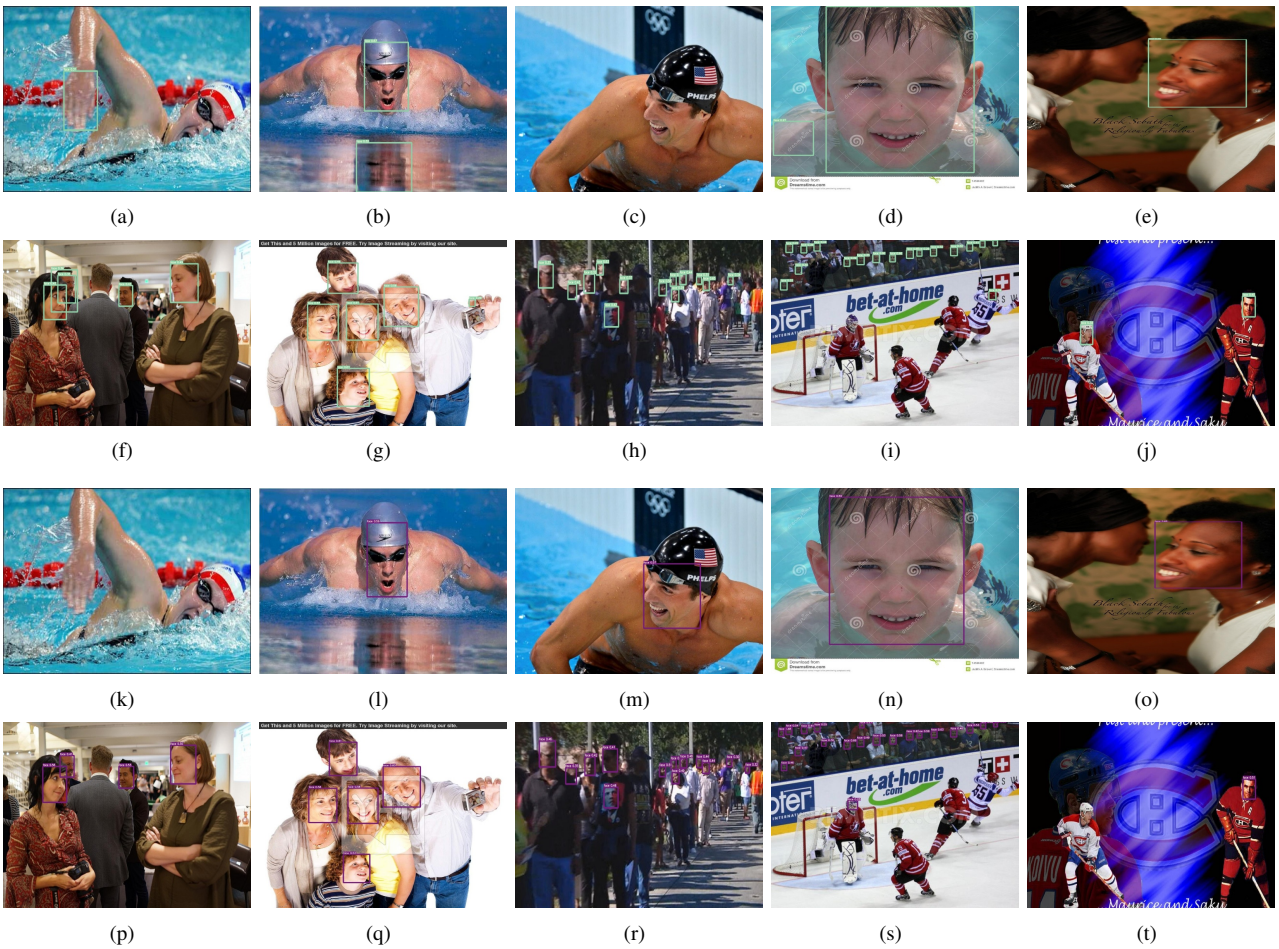

Figure 5. Examples of face detections from the *easy* dataset: (a)-(j) are detections from YOLOv5face, while (k)-(t) are detections from YOLOv7face.

medium, and hard datasets is 5px, 1px, and 1px, respectively. These mean values indicate that YOLOv7face slightly excels above the YOLOv5face in accurate face localization.

The minimum localization error of both YOLOv5face and YOLOv7face for the *easy*, *medium*, and *hard* datasets are all 0px. The 0px indicates cases where both YOLOv5face and YOLOv7face achieve perfect localization.

The maximum localization error of YOLOv5face for the *easy*, *medium*, and *hard* datasets are 114px, 122px, and 0px respectively, while the maximum localization error of YOLOv7face for the *easy, medium, and hard* datasets is 116px, 155px, and 80px, respectively.

Figure 9 clearly highlights the distribution of localization errors across different error levels. It is evident that a significant proportion of these errors falls within the category of very low errors.

These localization error results demonstrate the impressive accuracy of both YOLOv5face and YOLOv7face, particularly YOLOv7face, in localizing faces within images. The mean errors, especially in the *hard* dataset for YOLOv7face, signify their suitability for high-precision facial analysis tasks. These models exhibit consistent performance, and occasional challenges are reflected in the maximum error.

Overall, these results reinforce the robustness and precision of YOLOv5face and YOLOv7face in face detection, emphasizing their potential for various real-world applications, including facial recognition, expression analysis, and more.

D. Error in width and height prediction

The mean error in the width prediction of YOLOv5face for the *easy*, *medium*, and *hard* datasets are 4px, 51px, and 36px respectively, while the mean error in the width prediction of YOLOv7face for the *easy, medium, and hard* datasets is 3px, 52px, and 34px, respectively. These mean values indicate that YOLOv7face excels above the YOLOv5face.

The minimum error in the width prediction of YOLOv5face and YOLOv7face for the *easy*, *medium*, and *hard* datasets are 0px, 1px, and 1px respectively. This indicates cases where both YOLOv5face and YOLOv7face the predicted width exactly matches the ground truth width.

The maximum error in the width prediction of YOLOv5face for the *easy*, *medium*, and *hard* datasets are 114px, 202px, and 82px respectively, while the mean error in the width prediction of YOLOv7face for the *easy, medium, and hard* datasets are 145px, 193px, and 121px, respectively.

For the errors in the height prediction of YOLOv5face for the *easy* data set, they range from 0px to 131px. The *medium* and *hard* datasets range from 4px to 150px and 6px to 97px, respectively. The errors in the height prediction of YOLOv7face for the *easy*, *medium*, and *hard* datasets

(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

Figure 6. Examples of face detections from the *medium* dataset: (a)-(j) are detections from YOLOv5face, while (k)-(t) are detections from YOLOv7face.

range from 0px to 115px, 0px to 132px, and 0px to 218px respectively. The mean error value in the height predictions of YOLOv5face for the *easy*, *medium*, and *hard* datasets are 6px, 49px, and 38px, respectively, while for YOLOv7face they are 4px, 44px, and 59px, respectively.

Figure 11 and Figure 10 show the distribution of errors in width and height between ground truth and predictions. A wider range of error values can also be observed among the height predictions compared to the width.

E. Inference speed

Efficiency and speed are crucial considerations for real-time applications, and YOLOv5face is designed with these priorities in mind [22]. For YOLOv5face, the mean inference times for the *easy*, *medium*, and *hard* datasets have resulted to be 80.85 ms, 86.68 ms, and 81.65 ms, respectively, for images ranging from 640×640 pixels to 1024×1538 pixels. Images of any size is accepted as input and is then resized to 640×640 which is accepted by the backbone. The minimum inference times for these datasets are 61.0 ms, 50.2 ms, and 44.8 ms, while the maximum times are 131.9 ms, 128.9 ms, and 150.1 ms.

The average frames per second (fps) for YOLOv5face on the *easy* dataset has been calculated as 12.37. This indicates that, on average, the model is capable of processing video frames at a rate of 12.37 frames per second for images falling under the *easy* category. For the *medium* and *hard* datasets, the average frames per second are 11.54 fps and 12.25 fps respectively.

The model's speed performance is noteworthy, allowing for swift keypoint detection. This makes it a suitable choice for applications where real-time or near-real-time processing is required.

Similarly, YOLOv7face maintains a high level of speed while achieving superior accuracy. Its use of a ResNet-v1d backbone, along with other optimizations, ensures efficient keypoint detection [21]. For YOLOv7face, the mean inference times for the *easy*, *medium*, and *hard* datasets are 116.10 ms, 124.74 ms, and 120.05 ms, respectively. The minimum inference times for these datasets are 89.5 ms, 67.7 ms, and 72.1 ms, while the maximum times are 209.8 ms, 193.0 ms, and 179.3 ms.

The average fps for YOLOv7face on the *easy*, *medium* and *hard* datasets, are 8.61 fps, 8.02 fps and 8.33 fps respectively

Figure 12 visualizes the distribution of the times required to make inference on an image.

While YOLOv5face offers lower mean inference times across the datasets, YOLOv7face demonstrates competitive performance, particularly considering its higher accuracy. YOLOv7face's slightly longer mean inference times are still

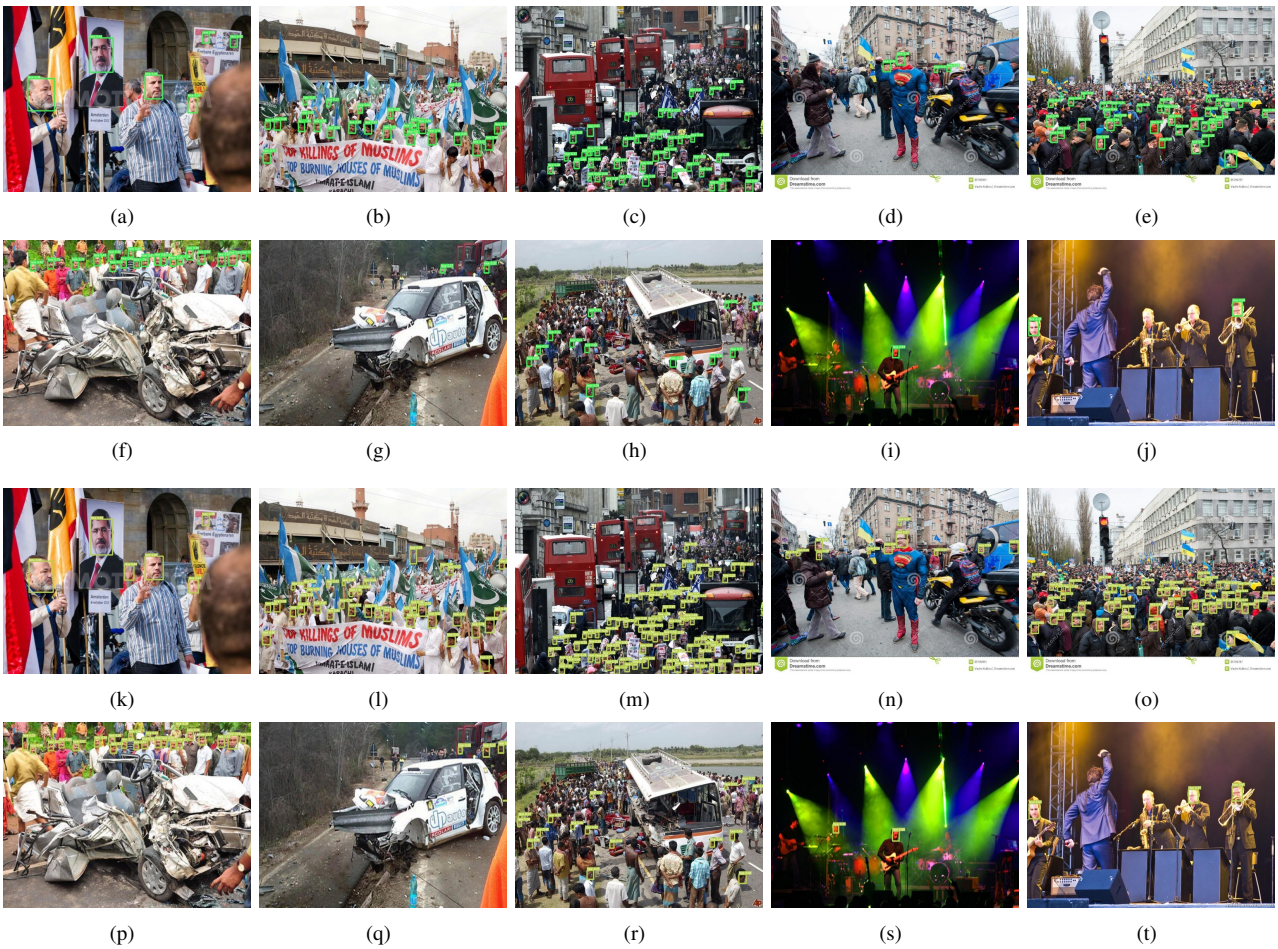

Figure 7. Examples of face detections from the *hard* dataset: (a)-(j) are detections from YOLOv5face, while (k)-(t) are detections from YOLOv7face.

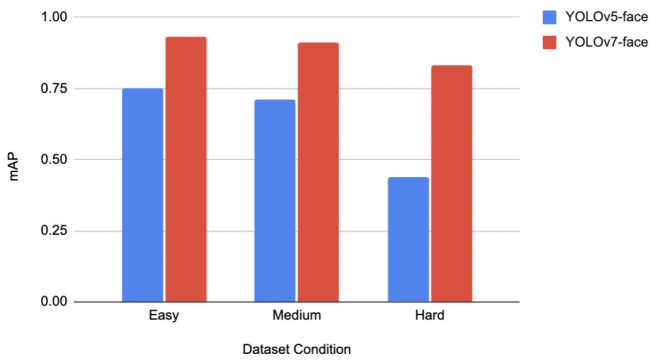

Figure 8. Mean Average Precision (mAP) over the *easy*, *medium* & *hard* datasets.

well within the range for real-time or near-real-time applications. Therefore, both models are suitable for various real-time face detection scenarios, with YOLOv7face excelling in accuracy and YOLOv5face excelling in speed. The choice between the two would depend on the specific requirements of the application.

F. Comparison with Other Existing Models

In this section, we conduct a comparative analysis of several face detection models based on their mean Average Precision (mAP) scores. Specifically, we evaluate the performance of four other models, together with YOLOv5face and YOLOv7face, namely Bresee_team2, WeFace, cv3ifly and RERe. These models resulted from the competition 'Wider Face & Person Challenge 2019 - Track 1: Face Detection' [37].

Comparing these models based on their mAP scores reveals interesting insights as seen in Figure 13. YOLOv7face and YOLOv5face outperform the other models, with YOLOv7face achieving the highest precision. This suggests that YOLOv7face is well-suited for applications where precise facial keypoint detection is crucial.

The relative differences in mAP scores highlight the significance of model selection. Depending on the specific requirements of an application, one might prioritize precision over other factors like speed. Moreover, the choice of the best model should consider factors such as computational resources and real-time processing constraints.

G. Architecture

The architecture of YOLOv5face is rooted in the YOLOv5 object detector, a renowned model known for its efficiency

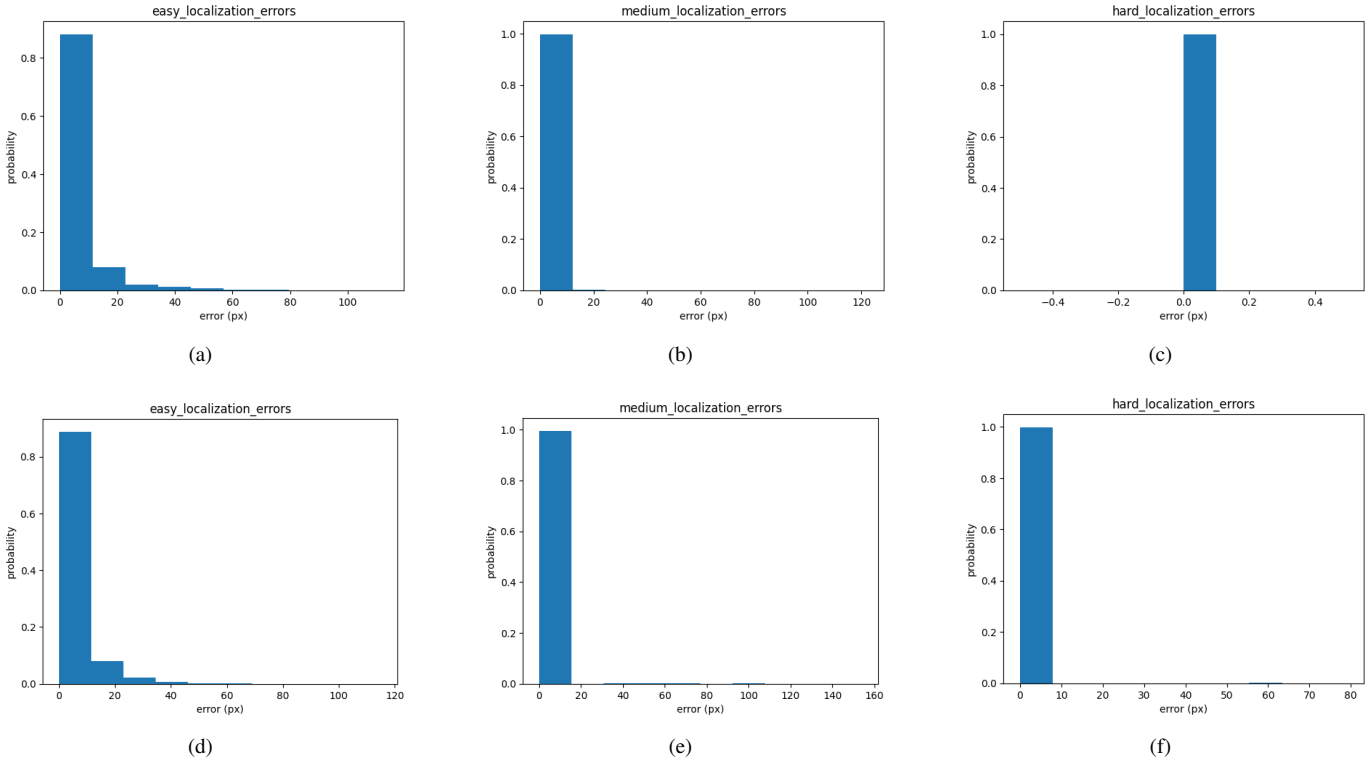

Figure 9. Distribution of the localization error for YOLOv5face and YOLOv7face over the *easy*, *medium* & *hard* datasets: (a)-(c) are from YOLOv5face, while (d)-(f) are from YOLOv7face.

and speed. YOLOv5face builds upon this foundation by introducing key modifications that tailor it for keypoint detection, particularly in the context of facial landmarks. These modifications encompass the addition of a landmark regression head and adjustments to the loss function to accommodate keypoint predictions.

The backbone network of YOLOv5face incorporates a CSPNet design, an architectural choice that enhances feature extraction and aids in the localization of facial keypoints. Additionally, a PANet neck is employed for feature aggregation, contributing to the model's ability to capture context and spatial relationships within an image [22].

In contrast, YOLOv7face takes a unique approach to its architecture. It ditches the traditional ImageNet pre-trained backbones and relies solely on training with the COCO dataset. The backbone network utilizes a ResNet-v1d architecture, which is known for its depth and capacity to extract high-level features. A PANet neck is also integrated into the architecture, facilitating feature fusion and enhancing the model's understanding of the image.

A distinguishing feature of YOLOv7face is the incorporation of a dedicated keypoint head. This head is responsible for predicting the facial keypoints, and it is trained on a dataset specifically annotated for facial keypoint detection. To refine keypoint predictions, a Gaussian filter is introduced in the post-processing step, reducing noise and enhancing the accuracy of keypoint localization [21].

H. Generalization Capability

The generalization capability of a model refers to its ability to perform well on unseen or previously unencountered data. It is a critical aspect of a model's performance as it determines its reliability and adaptability to real-world scenarios. In the context of face detection, a model's generalization capability is tested by its performance on subsets of a dataset that vary in difficulty, lighting conditions, occlusions, and other factors.

YOLOv5face demonstrates a noteworthy degree of generalization capability. It achieves competitive mAP scores on both the *easy* and *medium* subsets of the Wider Face dataset. These subsets encompass a wide range of scenarios, from well-lit and well-posed images to images with moderate challenges such as occlusions and variations in lighting conditions. The model's ability to maintain accuracy across these diverse scenarios indicates its versatility and robustness in handling common real-world conditions.

On the other hand, YOLOv7face excels in terms of generalization. It consistently delivers high accuracy across all subsets of the Wider Face dataset, including the *easy*, *medium*, and *hard* categories. YOLOv7face's capacity to maintain its performance under these adverse conditions demonstrates its exceptional generalization capability.

Additionally, YOLOv7face's adaptability to blurred images and its ability to distinguish between genuine faces and reflections in water surfaces highlight its robustness in handling a wide spectrum of real-world scenarios.

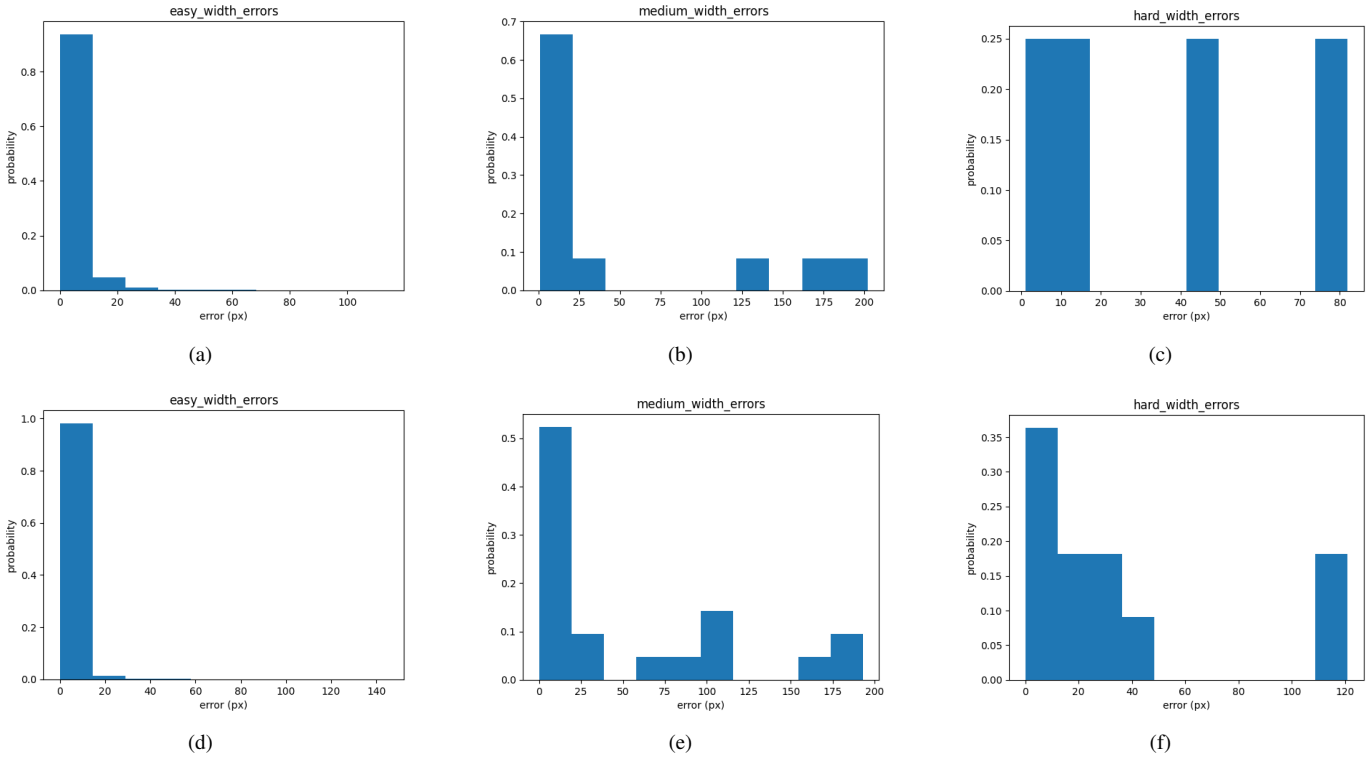

Figure 10. Distribution of the error in width predictions for YOLOv5face and YOLOv7face over the *easy*, *medium* & *hard* datasets: (a)-(c) are from YOLOv5face, while (d)-(f) are from YOLOv7face.

I. Specific Features and Advancements

YOLOv5face introduces a landmark regression head as a notable feature. This addition allows the model to detect and predict facial landmarks accurately. Facial landmarks are pivotal in tasks such as face alignment, which is crucial for subsequent face recognition or expression analysis. The landmark regression head enhances the model's versatility and its suitability for applications requiring fine-grained facial analysis.

Additionally, YOLOv5face leverages the CSPNet-based backbone and PANet neck for feature extraction and aggregation. These architectural choices contribute to its ability to capture both local and global context, aiding in accurate keypoint detection [22].

YOLOv7face builds upon the success of YOLOv7, focusing its capabilities on face detection. Its unique features include the utilization of a ResNet-v1d backbone, which offers increased depth and capacity for feature extraction. The PANet neck enhances feature fusion, enabling the model to understand complex spatial relationships within an image.

One of the key advancements in YOLOv7face is the dedicated keypoint head, which is specifically trained for facial keypoint detection. This head incorporates a novel loss function that combines cross-entropy and smooth L1 loss terms to optimize both classification and regression of keypoints. Furthermore, the inclusion of a Gaussian filter in post-processing refines keypoint predictions, reducing noise and enhancing accuracy.

V. Discussion

A. Discussion of Strengths and Weaknesses of Each Model

In the following we discuss the strengths and weaknesses of YOLOv5face and YOLOv7face in view of the results reported so far and the performance observed throughout the different tests carried out:

- **YOLOv5face strengths.** YOLOv5face demonstrates a remarkable balance between accuracy and speed, rendering it exceptionally suitable for real-time or near-real-time applications. Its efficiency in processing images at impressive speed as observed in Figure 12 makes it an attractive choice for scenarios demanding timely responses.
  Moreover, the inclusion of a landmark regression head significantly enhances YOLOv5face's versatility as observed in Figure 5, in Figure 6, and in Figure 7. This feature empowers the model to excel in tasks requiring precise facial keypoint detection, such as facial recognition and emotion analysis.
- **YOLOv5face weaknesses.** Despite its promising performance in several subsets of the Wider Face dataset, the poor performance on the *hard* subset in Table I, requires further evaluation to assess YOLOv5face's adaptability to diverse and varying environmental conditions. The need for such evaluations arises from the inherent complexities encountered in real-world scenarios, where lighting conditions, occlusions, and facial variations may differ significantly from the benchmark datasets.
- **YOLOv7face strengths.** YOLOv7face emerges as a

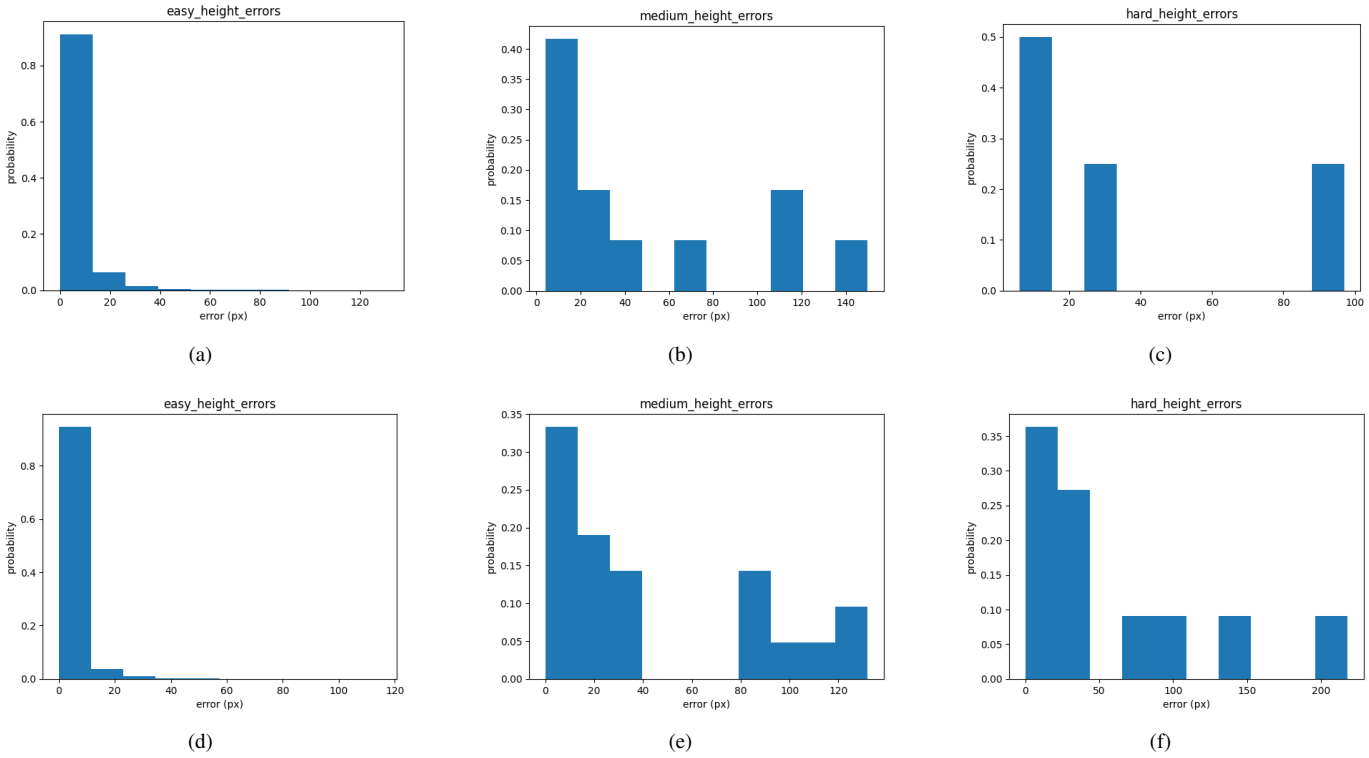

Figure 11. Distribution of the error in height predictions for YOLOv5face and YOLOv7face over the *easy*, *medium* & *hard* datasets: (a)-(c) are from YOLOv5face, while (d)-(f) are from YOLOv7face.

Table II
SUMMARY OF THE QUALITATIVE COMPARATIVE ANALYSIS OF YOLOV5FACE AND YOLOV7FACE

Aspect	**YOLOv5face**	**YOLOv7face**
Architecture	Architecture based on YOLOv5, CSPNet design, PANet neck	Unique architecture, ResNet-v1d backbone, PANet neck, dedicated keypoint head
Generalization Capability	Demonstrates robust generalization	Strong generalization, resilient in adverse conditions
Inference speed	Shorter inference time	Slightly longer time for inference, but not significant
mean Average Precision (mAP)	mAP scores: 0.75 (Easy), 0.71 (Medium), 0.44 (Hard)	mAP scores: 0.93 (Easy), 0.91 (Medium), 0.83 (Hard)
Specific Features	Landmark regression head, CSPNet, PANet	ResNet-v1d backbone, dedicated keypoint head, Gaussian filter

leader in terms of accuracy, consistently outperforming its counterparts across all subsets of the challenging Wider Face dataset as seen in Table I. Its superior accuracy makes it an attractive choice for applications demanding precision, even in the face of challenging scenarios.
This model demonstrates robustness in adverse conditions, such as occlusion and variations in lighting as observed in Figure 5, in Figure 6, and in Figure 7. Its ability to maintain high accuracy under challenging circumstances positions it as a reliable choice for real-world applications where environmental factors can vary unpredictably.

The introduction of a dedicated keypoint head and a novel loss function in YOLOv7face significantly enhances its performance in facial keypoint detection tasks. This specialization ensures precise localization of facial keypoints, making it particularly valuable in applications such as facial feature analysis and tracking.

- **YOLOv7face weaknesses.** While YOLOv7face excels in accuracy, it may require slightly more computational resources compared to YOLOv5face due to its utilization of a deeper ResNet-v1d backbone. This trade-off should be considered when selecting the model for applications with strict resource constraints.

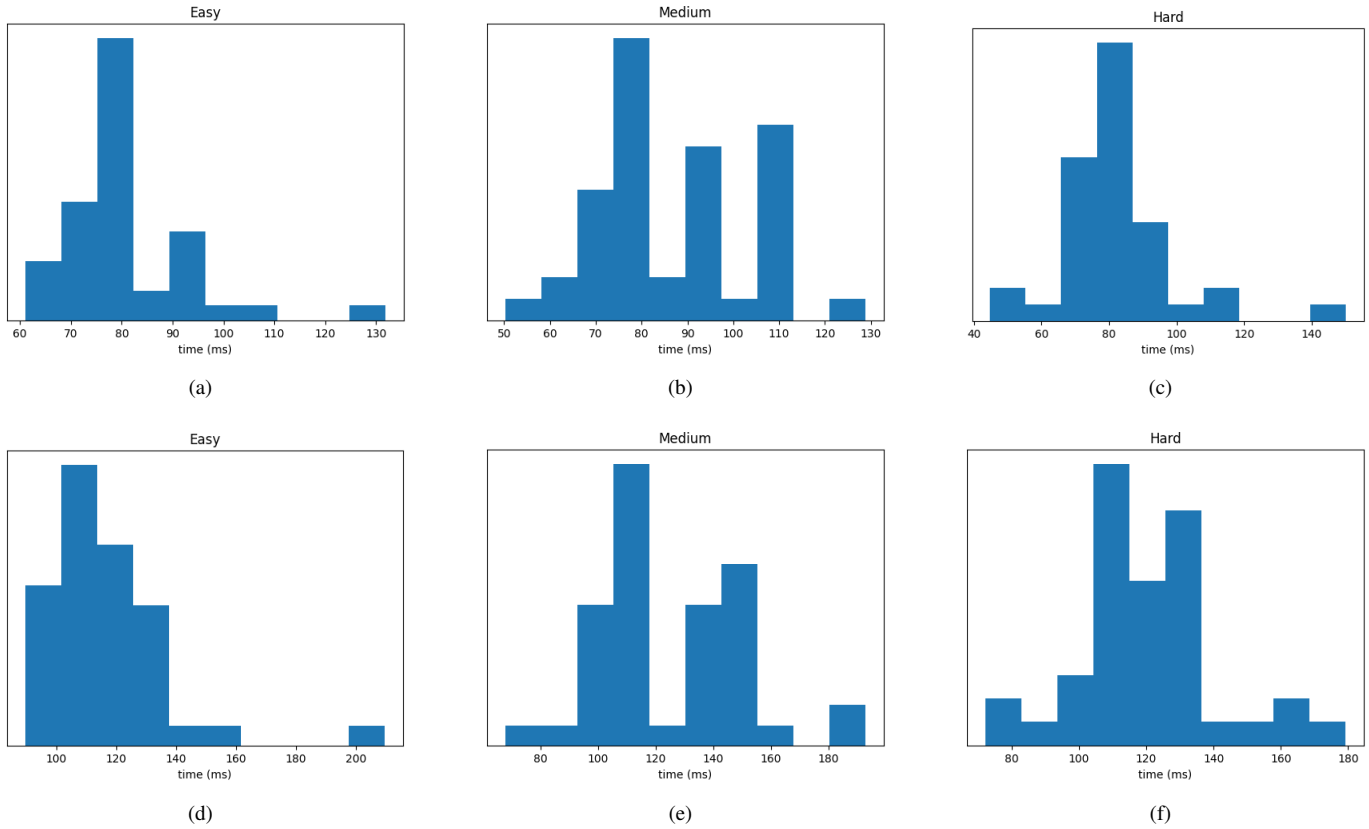

Figure 12. Inference times for YOLOv5face and YOLOv7face over the *easy*, *medium* & *hard* datasets: (a)-(c) are from YOLOv5face, while (d)-(f) are from YOLOv7face.

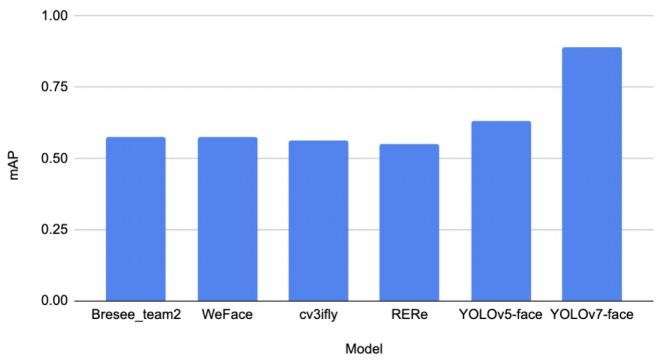

Figure 13. mAP of Several Models on the Wider Face Dataset.

Additionally, it is important to note that, despite its superior accuracy, YOLOv7face sacrifices a slight amount of speed compared to YOLOv5face highlighted in Figure 12. Therefore, the choice between the two models should also take into account the specific requirements of the application, as speed may be a critical factor in certain scenarios.

In summary, both YOLOv5face and YOLOv7face offer unique strengths and exhibit certain weaknesses, making them suitable for different contexts and use cases.

B. *Concluding Remarks*

In conclusion, both YOLOv5face and YOLOv7face exhibit strong performance in keypoint detection on the custom dataset of human faces. While YOLOv5face excels in efficiency and offers competitive accuracy, YOLOv7face stands out with its exceptional accuracy and robustness, positioning it as a promising choice for high-precision facial keypoint detection. Further evaluation and real-world testing will provide valuable insights into the practical applications and strengths of these models. The summary of the above points can be found in Table II.

VI. Conclusion and Future Work

A. *Conclusion*

In this comparative study, we explored the performance of two modified versions of the YOLO algorithms, YOLOv5face, and YOLOv7face, for keypoint detection on a custom dataset of human faces. Our analysis encompassed various aspects, including the mAP, the localization error, the inference speed, the architecture, the generalization capability, and the specific features.

YOLOv5face demonstrated commendable efficiency and versatility. It strikes a balance between accuracy and speed, making it suitable for real-time or near-real-time applications. The addition of a landmark regression head enhances its adaptability to tasks requiring precise facial keypoint detection.

While it performed well on subsets of the Wider Face dataset, further testing in diverse scenarios is warranted.

In contrast, YOLOv7face excelled in accuracy across all subsets of the Wider Face dataset, demonstrating remarkable performance even in challenging conditions. Its robustness in handling occlusions, variations in lighting, and pose variations positions it as a reliable choice for real-world applications. The dedicated keypoint head and novel loss function contribute to its precise keypoint detection capabilities.

B. Future Work

The comparative analysis presented in this study opens the door to several avenues for future research and improvement:

- **Dataset diversity.** Expanding the training and evaluation to include a wider variety of datasets can provide a more comprehensive understanding of the models' generalization capabilities. Testing on datasets with even more different lighting conditions, ethnicities, and age groups can further assess their real-world applicability.
- **Fine-tuning and hyperparameter tuning.** Fine-tuning the models on specific applications or domains can optimize their performance further. Hyperparameter tuning may reveal configurations that maximize accuracy while maintaining efficiency.
- **Real-world testing.** Here, we refer to conducting real-world tests in practical applications, such as robotics, and healthcare, can validate the models' performance and identify potential challenges in deployment.
- **Improvements on efficiency.** Exploring techniques to enhance the efficiency of YOLOv7face, such as model quantization or hardware acceleration, can make it more accessible for resource-constrained environments.
- **Enhancing the generalization capability.** Ensuring reliable performance across a broader spectrum of situations.
- **Transfer learning.** Investigating the effectiveness of transfer learning from YOLOv7face to other keypoint detection tasks beyond facial landmarks can extend the models' applicability.
- **Human-Robot Interaction.** Applying the models to HRI scenarios, such as gesture recognition or emotion analysis, can demonstrate their utility in enhancing human-robot interactions.

The field of face detection has witnessed remarkable advancements, exemplified by models like YOLOv5face, YOLOv7face and many others. These models have brought us closer to achieving the delicate balance between accuracy and speed. The integration of landmark regression heads, specialized loss functions, and robust backbones has propelled the accuracy and versatility of these models.

However, the journey towards perfecting face detection is far from over. As we move forward, we must continue to address the challenges posed by real-world scenarios, including varying environmental conditions, occlusions, and diverse facial expressions.

Moreover, the ethical considerations surrounding face detection and its applications will play a pivotal role in shaping the field. Striking the right balance between innovation and privacy, ensuring fairness and inclusivity, and guarding against misuse will be ongoing challenges.

In the years to come, we can anticipate even more exciting developments in face detection technology. As we continue to push the boundaries of what is possible, the future of face detection holds the promise of safer, smarter, and more inclusive applications that benefit society as a whole.

REFERENCES

[1] T. Baltrušaitis, P. Robinson, and L.-P. Morency. Openface: an open source facial behavior analysis toolkit. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–10, 2016.

[2] A. Barroso-Laguna and K. Mikolajczyk. Key.Net: Keypoint detection by handcrafted and learned cnn filters revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):698–711, 2022.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.

[4] A. Bochkovskiy, C. Wang, and H. M. Liao. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[5] J. Chen and Z. Lian. Textpolar: Irregular scene text detection using polar representation. *International Journal on Document Analysis and Recognition*, 24(4):315–323, 2021.

[6] S. Colaco and D. S. Han. Facial keypoint detection with convolutional neural networks. In *International Conference on Artificial Intelligence in Information and Communication*, pages 671–674, 2020.

[7] P. Dileep, B. K. Bolla, and S. Ethiraj. Revisiting facial key point detection: An efficient approach using deep neural networks. *arXiv preprint arXiv:2205.07121*, 2022.

[8] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun. YOLOX: Exceeding YOLO series. *arXiv preprint arXiv:2107.08430*, 2021.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[10] C. Guo, J. Liang, G. Zhan, Z. Liu, M. Pietikäinen, and L. Liu. Extended local binary patterns for efficient and robust spontaneous facial micro-expression recognition. *IEEE Access*, 7:174517–174530, 2019.

[11] R. Herpers, M. Michaelis, K.-H. Lichtenauer, and G. Sommer. Edge and keypoint detection in facial regions. In *International Conference on Automatic Face and Gesture Recognition*, pages 212–217, 1996.

[12] F. Karray, A. Campilho, and A. Yu. *In International Conference on Image Analysis and Recognition, Part I*, volume 11662. 2019.

[13] A. Kaur, M. Kumar, and M. Jindal. Cattle identification system: A comparative analysis of SIFT, SURF and ORB feature descriptors. *Multimedia Tools and Applications*, pages 1–23, 2023.

[14] Z. Keita. YOLO object detection explained. https://www.datacamp.com/blog/yolo-object-detection-explained, 2022. Last access: 2023-10-09.

[15] D. King, L. Burget, and R. S. Zemel. DLIB: A library of efficient c++/python tools for computer vision. https://github.com/davisking/dlib, 2009. Last access: 2023-09-10.

[16] R. King. MMYOLO: OpenMMLab YOLO series toolbox and benchmark. https://github.com/open-mmlab/mmyolo, 2022.

[17] J. Križaj, V. Štruc, and N. Pavešić. Adaptation of SIFT features for face recognition under varying illumination. In *International Convention on Information and Communication Technology, Electronics and Microelectronics*, pages 691–694, 2010.

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[19] S. Longpre and A. Sohmshetty. Facial keypoint detection. *Facial Detection Kaggle competition*, 2016.

[20] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[21] D. Qi. Yolov7-face. https://github.com/derronqi/yolov7-face, 2022. Last access: 2023-10-01.

[22] D. Qi, W. Tan, Q. Yao, and J. Liu. Yolo5face: why reinventing a face detector. In *European Conference on Computer Vision*, pages 228–244, 2022.

[23] J. Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016. Last access: 2023-07-28.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[25] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 2017.

[26] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[28] M. A. N. Reza, E. A. Z. Hamidi, N. Ismail, M. R. Effendi, E. Mulyana, and W. Shalannanda. Design a landmark facial-based drowsiness detection using DLIB and OpenCV for four-wheeled vehicle drivers. In *International Conference on Telecommunication Systems, Services, and Applications*, pages 1–5, 2021.

[29] B. D. Roads, B. Xu, J. K. Robinson, and J. W. Tanaka. The easy-to-hard training advantage with real-world medical images. *Cognitive Research: Principles and Implications*, 3(1):1–13, 2018.

[30] S. Shi. Facial keypoints detection. *arXiv preprint arXiv:1710.05279*, 2017.

[31] Z. Tian, C. Shen, H. Chen, and T. He. FCOS: A simple and strong anchor-free object detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):1922–1933, 2020.

[32] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161:272–279, 2019.

[33] Ultralytics. Yolov5: Real-time object detection. https://github.com/ultralytics/yolov5, 2021. Last access: 2023-09-10.

[34] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[35] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao. You only learn one representation: Unified network for multiple tasks. *arXiv preprint arXiv:2105.04206*, 2021.

[36] G. Wang. Paddledetection, object detection and instance segmentation toolkit based on paddlepaddle. https://github.com/PaddlePaddle/PaddleDetection, 2019. Last access: 2023-07-25.

[37] Wider Face. Codalab Competition. https://competitions.codalab.org/competitions/20146#results, 2023. Last access: 2023-10-01.

[38] W. Xiong, W. Tian, Z. Yang, X. Niu, and X. Nie. Improved fast corner-detection method. *The Journal of Engineering*, 2019(19):5493–5497, 2019.

[39] S. Yang, P. Luo, C.-C. Loy, and X. Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5533, 2016.

[40] W. Yun, J. P. Kumar, S. Lee, D.-S. Kim, and B.-K. Cho. Deep learning-based system development for black pine bast scale detection. *Scientific reports*, 12(1):606, 2022.

[41] J. Zhang, Z. Chen, and D. Tao. Towards high performance human keypoint detection. *International Journal of Computer Vision*, 129(9):2639–2662, 2021.

[42] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.