IT Journal Research and Development (ITJRD)
Vol.10, No.2, March 2026, E-ISSN : 2528-4053 | P-ISSN : 2528-4061
DOI : 10.25299/itjrd.2025.22402

Empirical Analysis of Deep Learning Models for Real-time Face Detection on Resource-constrained Devices

Bassey Isong[1], Sedzani Ndouvhada[2], Otshepeng Kgote[3]
Department of Computer Science, North-West University
Bassey.isong@nwu.ac.za[1], 34568018@mynwu.ac.za[2], 27788636@mynwu.ac.za[3]

Article Info

Article history:

Received May 12, 2025
Revised Jun 09, 2025
Accepted Sep 18, 2025

Keyword:

Face Detection
YOLOv8
SSD
Faster RCNN
Mobile Devices

ABSTRACT

Face detection (FD) is central to biometric systems used in mobile authentication. However, on resource-constrained devices, real-time use requires balancing accuracy and efficiency. Additionally, variations in pose, lighting, occlusions, dataset quality, and hardware often limit how well the system works in real use. This study presents a comprehensive empirical evaluation of deep learning-based object detection techniques, specifically YOLOv8, SSD, and Faster RCNN, to assess their effectiveness in addressing real-world scalability and performance demands. These models were trained on diverse datasets and evaluated using key performance metrics, including accuracy, precision, recall, and frames per second (fps). YOLOv8 achieved superior performance, achieving 42.32 fps with an accuracy of 86%, surpassing two-stage models in real-time processing speed while maintaining comparable accuracy. The findings underscore the importance of dataset quality and diversity in enhancing model performance and positioning YOLOv8 as an effective solution for balancing speed and accuracy on the COCO dataset. The study envisions a future exploration of hybrid models that integrate YOLOv8's efficiency with Faster RCNN's precision to develop more robust FD solutions tailored to real-world challenges.

Corresponding Author:

Bassey Isong
Department of Computer Science
North-West University
Mafikeng, South Africa
Email: bassey.isong@nwu.ac.za

1. INTRODUCTION

Face detection (FD) technology has transformed human-computer interaction over recent decades, enabling machines to identify faces in digital images and videos through artificial intelligence, machine learning (ML), statistical models, and image processing [1, 2, 3]. By detecting features like eyes, lips, and nose with precision honed on large datasets, FD supports applications from biometric security and social media features to augmented reality and intuitive mobile interfaces [1, 2]. In mobile applications, FD enables biometric authentication, photography enhancements, and intuitive interface design [1, 2]. However, its deployment is challenging due to limited processing power, memory, and battery life, alongside environmental variables like inconsistent lighting and diverse facial orientations [3]. Current algorithms struggle to balance speed and accuracy: one-stage models like You Only Look Once (YOLO) prioritise speed over precision, while two-stage models like Faster Region Convolutional Neural Networks (Faster RCNN) provide higher accuracy at a greater computational cost [3, 4]. Biases from non-diverse

training datasets, lacking variation in age, ethnicity, or image quality, and hardware constraints further impair performance [5, 6, 7].

Furthermore, over the years, FD has evolved from feature-based methods to ML-enhanced techniques, with the Viola-Jones framework enabling faster real-time applications and DL, particularly CNNs, now leading due to their robustness against occlusions and lighting variations [6, 8, 9]. These advancements have expanded FD's applications in security, consumer technology, and healthcare, including patient identification and emotional recognition [4, 10, 11]. This study evaluates DL-based object detection for an optimised FD framework to achieve high accuracy and efficiency for real-time deployment on resource-constrained mobile devices. Leveraging CNN architectures, the study seeks to evaluate performance across diverse datasets while ensuring practicality in terms of which model strikes a balance between efficiency and accuracy [4, 12], advancing FD in mobile computing. The main contributions of this paper are:

1. Explores DL algorithms and conducts comparative performance analysis, specifically the one-stage and two-stage detector models, to enhance speed and accuracy for FD tasks using different datasets.
2. Further evaluate the best-performing model on additional datasets, including the Labelled Faces in the Wild (LFW) dataset.
3. A real-time evaluation of the best-performing model is provided to demonstrate its practical applicability.

The remainder of this paper is organised as follows: Section 2 provides background information on DL models and FD applications' technology, Section 3 reviews related works, while Section 4 presents the study's methodology. Section 5 presents the results and the discussion of the findings, while Section 6 concludes the paper.

2. LITERATURE REVIEW

2.1. FD Methods Overview

FD has been a dynamic research field for decades, initially relying on handcrafted features and statistical models. The Viola-Jones algorithm, leveraging Haar-like features and cascaded classifiers, marked a significant advance for real-time FD but struggled with complex conditions like varying lighting, occlusions, and non-frontal poses, as did other classical methods like edge detection and template matching [8, 9].

The advent of DL revolutionised FD by enabling precise, adaptable, and context-aware approaches through artificial neural networks, particularly CNNs [3, 4]. CNNs excel in extracting hierarchical features from grid-structured data, making them ideal for FD tasks such as image classification and object detection [1, 3-5, 14, 16]. Modern FD frameworks are categorised into one-stage and two-stage object detection models. One-stage detectors, such as YOLO and SSD, predict bounding boxes and class probabilities in a single pass, offering high speed for real-time applications through optimised anchor boxes and hyperparameters [4, 19-21]. In contrast, two-stage detectors, including RCNN, Fast R-CNN, Faster R-CNN, and multi-task CNN (MTCNN), generate region proposals before refining classifications and bounding boxes, achieving superior precision for tasks like pedestrian detection, video surveillance, and object tracking [19, 20]. In addition, several studies have demonstrated that these detectors perform robustly in face localisation and recognition across diverse settings, leveraging large-scale datasets and powerful computing resources [4]. Advanced techniques, including federated learning, transfer learning, attention mechanisms, and lightweight models like MobileNet and EfficientNet, further enhance accuracy and enable efficient FD on resource-constrained mobile devices [4, 5]. These developments highlight FD's growing potential in applications requiring high performance and scalability.

Furthermore, FD technology drives innovation across multiple sectors by enabling precise facial recognition and analysis. In security, FD enhances access control, surveillance, and prisoner management through biometric identity verification, aiding police investigations and locating individuals in crowded settings [2, 22, 23]. In transportation, FD improves safety by monitoring driver alertness, detecting pedestrians for autonomous vehicles, and activating safety features like airbags to prevent accidents [22, 24]. In banking, FD strengthens secure remote transactions by verifying customer identities, using algorithms like Haar Cascades, YOLO, and Faster R-CNN to

counter fraud and face spoofing [25-27]. In healthcare, FD supports patient identification, diagnoses facial abnormalities, detects conditions like breast and lung cancer using CNN-based models, and provides personalised dermatology recommendations by analysing facial expressions. During the COVID-19 pandemic, FD facilitated temperature screening and mask compliance monitoring [7, 28, 29]. Additionally, FD enhances human-computer interaction by recognising gestures and expressions in virtual meetings, enabling intuitive and responsive user experiences [30, 31]. These diverse applications underscore FD's transformative impact across industries.

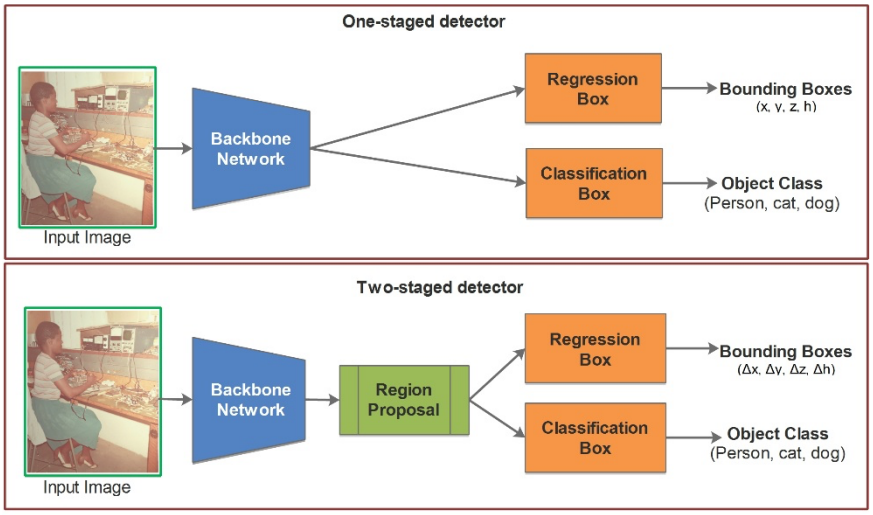


Figure 1. One and Two-Stage Detector Models.

2.2. Related Works

This section examines the current literature regarding DL approaches for object detection, with a particular focus on two dominant frameworks in real-time detection tasks: CNNs and YOLO, as summarised in Table 1. A recurring theme across these studies is the ongoing effort to balance speed and accuracy, particularly in resource-constrained environments. In the context of FD, several technical and deployment challenges are highlighted. Zhang et al. [3] and Phatak et al. [8] identified pose variation, occlusion, ageing, and lighting as persistent obstacles, although practical optimisation strategies and robust deployment frameworks were often underdeveloped. Similarly, Liu et al. [5] proposed an FL to address edge computing constraints, though their work lacked empirical validation. Furthermore, various solutions have been explored for real-time FD Phankokkruad and Jaturawat [32] introduced secure transmission methods via WebRTC and HTTPS, but again, without thorough experimentation, while Guo and Wünsche [9] focused on mobile robot applications but overlooked newer techniques and diverse datasets. Similarly, Majeed et al. [6] demonstrated the practical potential of FD in security applications via combined YOLOv5s, RetinaFace, and Facenet512 for anti-theft systems but struggled with scalability in larger areas. Liu et al. [33] also tackled wrinkle detection using neural networks but failed to address biases on diverse skin types, though their approach needs testing across diverse demographics.

Recent advancements and alternative approaches provide additional insight into improving FD systems. Zhang et al. [35] suggested an enhanced AdaBoost algorithm to reduce false detections and improve efficiency, while Sirivarshitha et al. [11] used OpenCV-based Python libraries to tackle lighting, pose, and expression issues, finding OpenCV effective but limited by its dependence on high-quality images. Also, Garg et al. [36] presented a YOLO-based method that performed well on the LFW dataset yet faced limitations with detecting smaller faces and managing extreme angles, while also suffering from high computational demands. In the same vein, advanced techniques, such as the YOLOv8-based model by Al-obidi et al. [34] and the MobileNet and GhostNet-optimised YOLOv4 by Shi and Gao [4], showed innovation in model

design but suffered from benchmarking and reproducibility issues. While [34] developed a YOLOv8-based FD model with six distinct facial classes using custom datasets, authors in [4] optimised YOLOv4 with MobileNet and GhostNet MobileNet and GhostNet but struggled with replication. In addition, Ranjan et al. [10] explored deception detection but called for ethical and standardised datasets. In the same vein, emerging DL models like those proposed by Sun et al. [38] and Wang et al. [37] showed promise in balancing performance with real-time constraints, though challenges in scalability and robustness remain. While [38] proposed a Faster RCNN-based FD scheme balancing accuracy and speed with multiscale training, but left scalability for real-time detection unexplored. [37] Introduced a Region Attention Network (RAN)for facial expression recognition, but faced limitations in extreme conditions. Meanwhile, integrated systems like that of Majeed et al. [6] demonstrate the practical potential of FD in security applications via combined YOLOv5s, RetinaFace, and Facenet512 for anti-theft systems, but struggled with scalability in larger areas.

Table 1. Summary of Related Works

Study	Focus Area	Model/Method	Datasets	Strengths	Limitations	Future Directions
[3]	FD challenges (pose, occlusion)	MTCNN, P-Net, R-Net, O-Net, YOLOv3	Wider Face	Highlights speed vs. accuracy	Limited dataset insights, Lower accuracy, landmark localisation issues	Optimise MTCNN; improve practical applications.
[8]	FD and recognition system issues	General DL models	-	Covers multiple deployment issues	Slow processing, accuracy under variation, complex architecture, no empirical results	Address model robustness & deployment constraints
[5]	DL on edge devices	Optimised DL and FL, VGG16, DQN, HERDQN	-	Smart application focus	Limited processing/storage, complexity for real-time tasks.	Practical testing on edge environments
[32]	Real-time FD and security	HTTPS, WebRTC, optimisation, Haar-like features, CLM	-	Real-time system discussion	Image/connection quality affects accuracy, trade-offs in FD performance	Integration with mobile apps; empirical validation
[33]	Wrinkle detection	NN + skeleton line methods	1021 facial wrinkle images	Novel dermatology use	Biased data; narrow scope	Broaden demographics; improve generalisation
[9]	FD in mobile robots	Viola-Jones, HOG, MTCNN, MobileNet-SSD	AFW, Wider Face	Comparative analysis	Outdated models & datasets	Apply newer techniques; broader evaluation
[10]	Deception detection	Psychological traits	Wider Face, UMD Faces, MS-Celeb-1M	Ethical perspective	Poor reproducibility, accuracy variation, FAR and FRR issues	Dataset standardisation; algorithm transparency
[34]	Facial feature detection	YOLOv8, custom dataset		Diverse facial features	No benchmark comparison	Expand use cases; validate broader generalizability.
[35]	Efficient FD	Improved	Public	Reduced	Improving	Test vs. state-of-

		AdaBoost	Face	false detection	detection rate, reducing false positives	the-art methods
[11]	FD/FR via OpenCV	OpenCV libraries	-	Effective under ideal conditions	High-quality image reliance	Lightweight models for low-res devices
[36]	FD under extreme angles	YOLO-based	FDDB	State-of-the-art on LFW	High compute cost, Lower accuracy on small/partial faces, details on optimisation lacking	Optimise for efficiency; improve on small faces.
[4]	Fast FD enhancement	YOLOv4, MobileNe, GhostNet	Wider Face, LFW	Higher accuracy	Missed detections; hard to replicate	Enhance robustness and reproducibility
[6]	Real-time access control	YOLOv5s, RetinaFace, Facenet512	-	Strong detection	Struggles in large areas	Focus on scalability for resource-limited setups
[37]	Facial expression recognition	RAN	-	Prioritises facial regions	Weak in extreme conditions	Improve robustness for field use
[38]	Enhanced FD pipeline	Faster RCNN, multi-detector setup	-	High accuracy & speed	Scalability not addressed	Real-time deployment scalability

As shown in Table 1, the studies emphasise advancing DL methods, particularly CNNs and YOLO models, to balance speed and accuracy in real-time applications. Researchers tackle challenges like pose variation, lighting, occlusion, and resource limitations through lightweight algorithms, enhanced architectures, and novel datasets [3, 4, 34]. Ethical considerations, including transparency, bias reduction, and dataset standardisation, aim to ensure fairness and applicability across diverse populations [10, 37]. A trend toward empirical validation highlights the need for practical testing to bridge theoretical models and real-world scenarios [5, 38]. In addition, scalability remains a key focus for DL methods, enabling efficient performance across resource-constrained devices and large-scale applications such as security systems and IoT [6, 33]. These efforts advance DL towards effective, adaptable, and ethical solutions for diverse domains.

3. RESEARCH METHOD

In this section, we outline the process of the proposed FD solution, encompassing data collection and preprocessing, model selection, training, evaluation, and real-time analysis. This systematic approach ensured a reliable assessment of the FD's performance. The evaluation prioritised metrics including accuracy, precision, and real-time performance, as shown in Fig. 2.

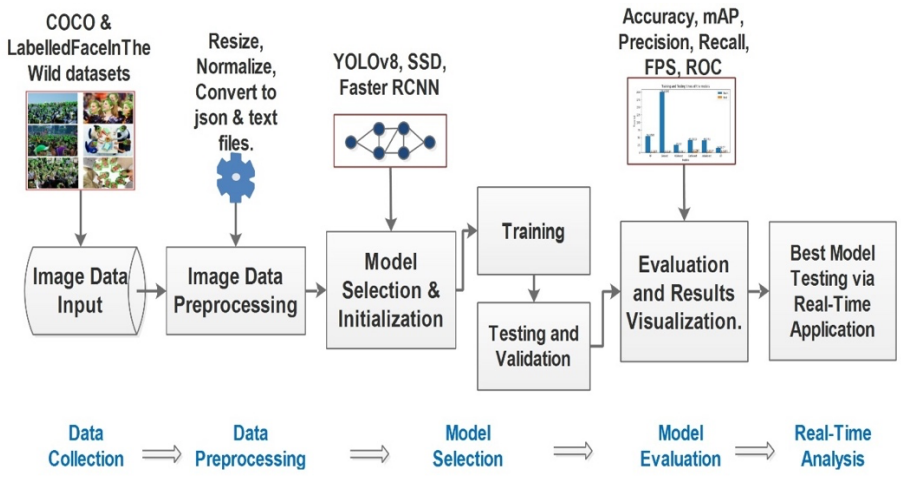


Figure 2. Proposed Workflow

3.1. Data Collection

This study utilised the COCO dataset to evaluate the models considered. The COCO dataset is a widely used benchmark for object detection and segmentation that features a diverse range of objects, including humans, animals, and vehicles, sourced from platforms such as Microsoft and Flickr. Initially released in 2014 and updated in 2017, the latest version includes 118K training images, 5K for validation, and a 41K image test subset of 118 K. Its rich variety of scenes and object instances, often with cluttered backgrounds, occlusions, and varied human poses, improved the robustness of our model. Moreover, the Labelled Faces in the Wild (LFW) dataset [4], a labelled extension of the Wider Face dataset [3, 4, 9, 10], was employed specifically for FD and recognition tasks in unconstrained environments. It contains 13,233 images of 5,749 individuals, gathered from various online sources. Due to compatibility constraints, we evaluated LFW solely using the YOLOv8 model. The availability of face annotations significantly supported the training and validation processes. Data availability:

1. COCO Dataset: https://www.kaggle.com/datasets/sabahesaraki/2017-2017, accessed (21 August 2024).
2. Labelled Faces In The Wild dataset: https://www.kaggle.com/datasets/jessicali9530/lfw-dataset, accessed (25 August 2024).

3.2. Data Preprocessing

Two different datasets were used to lay the groundwork for training and evaluating the detector models. However, one of the challenges we faced in this study is computational limitation, and this constraint was due to limited hardware resources or facilities with capabilities to process and train the models on larger data. Therefore, we ended up utilising a total of 250 images per dataset to train each model. With 175 images allocated for training and 35 for validation. In the case of the COCO dataset, which contains diverse object classes, annotations were reformatted and JSON files to include only two categories: person (representing images with human faces and related features) and not a person (representing all other objects). This simplification aligned with our study's focus on FD, while filtering the dataset to include only images with human faces helped reduce computational overheads. Moreover, the number of images used labelled wider face dataset consisted of image files in jpeg file format and with a labels file in .txt format. Due to compatibility constraints, this dataset was evaluated exclusively with the YOLOv8 model. The YOLOv8 framework supported the native annotation format without requiring conversion. Each face was annotated using bounding box coordinates in the format [xmin, ymin, width, height], accurately marking face locations within the images.

3.3. Model Selection

The subsection presents the selected one and two-stage detector models considered in this study, which are YOLO version 8, SDD, and Faster R-CNN. Table 2 summarises their strength and limitations. YOLO is a detector that processes an image in a single pass, which divides it into a grid where each cell predicts both bounding boxes and class probabilities. Its one-stage architecture enables fast detection of multiple objects, including faces, making it ideal for real-time applications. Introduced by Redmon et al. in 2015, YOLO has evolved into a leading object detection framework, estimating object presence, bounding box coordinates, and confidence scores for each grid cell [6, 21, 39-41]. Similarly, the model uses a feed-forward CNN and multiple feature maps to detect objects at various scales, improving its ability to recognise objects of different sizes in one pass. When combined with lightweight architectures such as MobileNet, SSD optimises performance for resource-constrained devices. It combines the speed of YOLO and the accuracy of two-stage detectors, such as Faster R-CNN, providing a balanced solution for real-time tasks [39-44]. In contrast, Faster R-CNN is a two-stage detector that uses a region proposal network to generate candidate object regions, which are then classified and refined. While slower than one-stage detectors, Faster R-CNN offers superior accuracy. The model uses backbone networks like VGG or ResNet to extract feature maps, which the RPN analyses to propose object locations, followed by classification and bounding box refinement, with final detections optimised through non-maximum suppression to remove redundancies [19, 30, 41, 45].

Table 2. Selected Models

Model	Type	Description	Advantages	Disadvantages	Ref.
YOLO	One-stage Detector	Fast real-time detection; predicts bounding boxes and class probabilities directly.	High speed, low latency.	Needs large datasets; struggles with small faces and complex architecture.	[6, 21, 39-41]
SSD	One-stage Detector	Uses multiple feature maps for single-pass predictions.	Good accuracy and speed.	High computational demand; lower accuracy for small faces.	[39, 41, 43]
Faster R-CNN	Two-stage Detector	Precise localisation via Regional Proposal Network.	Very accurate for crowded images.	Slower processing; unsuitable for real-time applications, resource-intensive.	[19, 30, 41, 45]

3.4. Model Evaluation

In this subsection, we discuss the training and evaluation process. Moreover, we outline the metrics utilised to evaluate models.

3.4.1. Model Training and Evaluation

Experiments were conducted to train and evaluate DL based FD models, which include YOLOv8, SSD and Faster RCNN, under constrained CPU and virtual environment without GPU access. This has led to the adjustment of the training strategies to accommodate these computational limitations. This was done through leveraging the pre-trained models, reducing batch sizes and minimising training epochs. Firstly, the computational resource constraint led the study to employ the manual hyperparameter tuning approach using manual grid search rather than the advanced automated grid and Bayesian search hyperparameters. The key parameters that were evaluated on a small scale include: learning rates (tested on 0.001 to 0.005), batch sizes (starting with 16 then 8), epochs (10), and IoU threshold (ranging from 0.3 to 0.7). The process of training and evaluation was mainly iterative; we refined each testing specification from a higher value, dropping to the best optimal value that is in line with our testing environment and datasets, which are detailed in Table 4.

Basic data augmentation techniques were implemented to improve the model generalisation using horizontal flipping, image rescaling, random cropping, pixel scaling, and brightness adjustments to simulate variable lighting conditions as supported by the built-in augmentation tools in PyTorch libraries and as they are commonly used in FD literature. However, the more advanced augmentation techniques and training optimisation strategies, such as quantisation and model pruning, were considered but not fully implemented due to limited hardware constraints. Lastly, the training and evaluation were executed in separate Jupyter (local) and Google Colab environments to accommodate the dependency conflicts and resource management of the one-stage and two-stage models. The local machine, using an HP desktop with just 12 GB RAM and no dedicated GPU, experienced limitations when the training dataset was greater than 250 images. The training process would usually crash or be stalled at ~65% progress due to memory overload or kernel timeout. Then with the Google Colab virtual environment, it offered temporary GPU acceleration when it was available due to the free version, then performance was impacted by the session timeouts, electricity power and network instabilities in the lab. This then led to larger datasets of ~500 images not being consistently trained and completed on either platform due to resource capacity limits.

However, stable training was achieved using a subset of ~250 images per model, and performance was monitored via manual checkpointing. Despite these limitations, models were evaluated using a consistent set of parameters that are documented in Table 4 in Section 4, to ensure a fair comparative analysis and reliable outcomes. Similar limitations were experienced in a related study by [36], where authors experienced computational expenses with YOLO and Faster RCNN-based models, and challenges of training on larger datasets without GPU acceleration. This then reinforces the practical boundaries we encountered and supports the methodology approaches that were proposed in this study of training each model independently in separate environments to mitigate compatibility issues.

3.4.2. Performance metrics

The models' effectiveness was assessed using important performance measures like classification accuracy, mean Average Precision (mAP), recall and precision of face proportionality, Receiver operating characteristics curve (ROC), confusion matrix, and processing time for detecting faces. Each metric is formally defined as follows:

In the object detection field, Average Precision (AP) and mean Average Precision (mAP) are common metrics used to measure performance and accuracy [44, 46]. AP is the average detection precision under a wide various recalls and is mainly evaluated for each object class independently [46]. In contrast, mAP is used to measure the mean average score for all classes in the dataset to evaluate a model's performance and accuracy [27, 46]. It ranges from 0 to 1, with 1 being the best performance. Equations 1 and 2 defined both metrics as follows:

$$AP = \int_0^1 p(r)dr \tag{1}$$

$$mAP = \frac{AP_n}{N} \tag{2}$$

Where p is the precision, r is the recall, and N represents the number of object categories.

The confusion matrix [39, 44] is used to evaluate the model's performance, showing how well its predictions align with the actual values. The actual key values include true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Accordingly, while the TP represents a face that was detected correctly, the TN is a non-face object correctly classified as a non-face. Similarly, FP represents a non-face object that has been incorrectly classified as a face, while FN represents a face that was not detected at all.

Prediction accuracy presents the overall correct predicted faces amongst the proportion of the correct and total predicted faces [47]. That is, it is utilised to predict whether an object in the image is a person or not a person as defined in terms of the matrix key values in equation 3.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

Precision is the ratio of correctly detected faces to the total number of objects detected. It is represented in terms of TP and FP predicted bounding boxes [39, 44]. In addition, it uses the Gamma Value, which is a measure of the detection accuracy at a specific FP rate (FPR) and is defined in Equation 4.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

Recall measures the ratio of correctly detected faces to the total number of actual faces in the dataset. It is also known as the detection rate (DR) [39, 44] and is shown in equation 5.

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

In the same vein, the F1-score is the metric for the equilibrium value of the detection accuracy in a model. It is visualised as the area under the precision-recall curve, or is more of a harmonic mean of precision and recall [39]. It is expressed in equation 6 as follows:

$$F1\_Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (6)$$

Frames Per Second (FPS) is a critical metric in object detection, which measures the speed at which the frames of images can be processed in one second by a model. In this case, any processing speed above 20 is ideal for real-time applications; otherwise, it may result in a lag [47]. It is expressed in equation 7 as follows:

$$FPS = \frac{Total\ Frames}{Total\ Time\ (in\ seconds)} \quad (7)$$

The ROC curve graphically demonstrates how well the algorithms correctly identify faces and the proportion of non-faces incorrectly identified as faces. It is used to illustrate the trade-off between recall and precision, highlighting how well the model performed [47].

3.5. Real-time analysis and Setup

Table 3 shows the hardware and software specifications used to carry out the entire experiment. To evaluate the selected detector models, we conducted a series of experiments on an HP All-in-One Desktop PC running Windows 11 Pro (64-bit), equipped with a 12th Gen Intel® Core™ i5-12500 processor (3.00 GHz), 8GB RAM, and a 512GB SSD. In addition, to ensure an isolated and reproducible environment, the implementation was conducted within a Python virtual environment. In this case, the development and image label augmentation were performed using PyCharm, while model training and testing were executed in Jupyter Notebook and Google Colab. The experiment utilised several libraries for the implementation and evaluation of both one-stage and two-stage detector models, including TensorFlow, PyTorch, Facenet, Ultralytics, Keras, Scikit-learn, OpenCV, Matplotlib, Seaborn, Pandas, and NumPy. Furthermore, to evaluate the models' effectiveness for real-time applications, we conducted a real-time analysis using the best-performing model among YOLOv8, SSD, and Faster RCNN. To achieve this, we used the desktop computer's built-in 5MP camera that was initially used, then from the Open CV library, we used image capturing functionalities which were set up to take a snapshot of an image by pressing the keyboard key "Q". This captured the images in real-time together with their accuracy levels without the detection speeds.

Table 3. System Properties

Parameters	Category	Specifications
Hardware	System Model	HP EliteOne 840 23.8-inch G9 All-in-One Desktop PC
	Processor	Processor 12th Gen Intel(R) Core (TM) i5-12500, 3GHz 6 Core(s), 12 Logical Processor(s)
	RAM	8GB
	SSD	512GB
	Camera	5MP IR Webcam (Integrated)
Software	Operating System	Windows 11 Pro, 64-bit

Local Environment	Python (3.9.0), JupyterLab (24.25) ,OpenCV(4.10.0.84) , Ultralytics(8.2.100) ,TensorFlow (2.11.0),Keras(3.6.0) , Pytorch (2.5.0) , Matplot (3.9.2) , Seaborn (0.13.2)
Virtual Environment	Google Collab: NVIDIA Tesla K80, T4, P100, RAM 12 GB, DISK 107GB.

4. RESULTS AND ANALYSIS

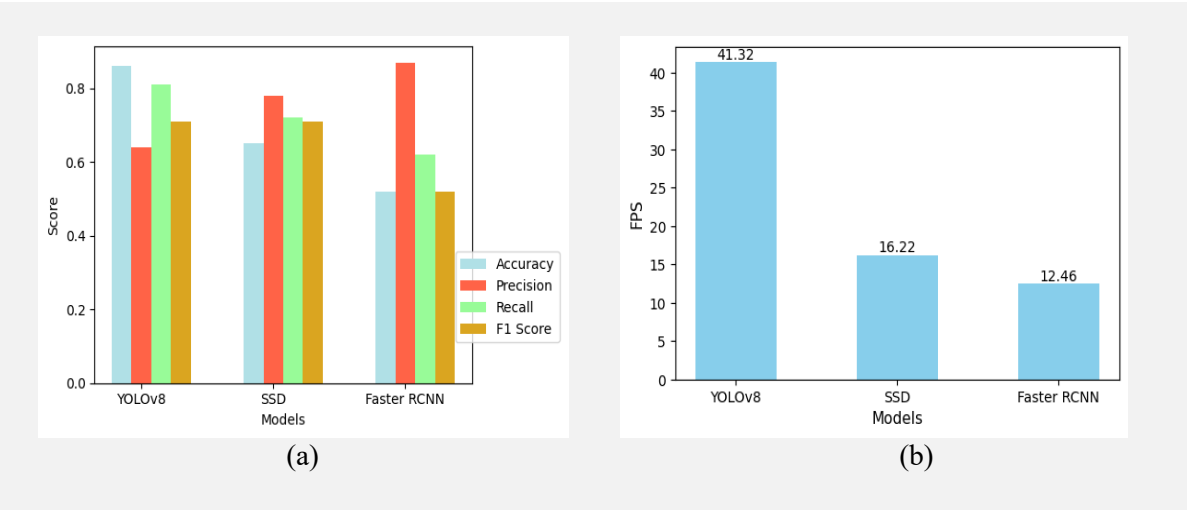
This section presents results from experiments after evaluating models on both datasets, with subsections on sample images and real-time analysis to compare the performance. The best-performing model underwent further analysis. For object detection, human faces were classified as "person" and other objects as "not person" due to limited face-specific annotated datasets. Performance metrics focused on one-stage and two-stage detectors. Figs. 3 to 8 display post-training performance, including loss curves from 10-epoch training with a batch size of 32, a learning rate of 0.005, and adjusted dataset sizes to avoid kernel timeouts, as summarised in Table 4. Moreover, decreasing loss curves indicate effective training, with the potential for optimising hyperparameters like learning rate and epoch count. A batch size of 16 proved more efficient than the 32 used in prior studies [5]. Sample images assessed model accuracy in face identification.

Table 4. Experimental parameters

Parameters	Values
Batch Size	16
Training Set	70 % of the dataset
Threshold	0.3
Test and Validation Set	15% each set
Learning Rate	0.005
Epochs	10 Epochs on each Model
Momentum	0.9
Weigh decay	0.0005
Images	In JPG, .jpeg formats

4.1. Comparative analysis of models

This subsection presents the evaluation results of models trained on 250 images per dataset, with 175 images for training and 35 for validation.



(a) (b)

Figure 3. Performance across datasets: (a) detector models' performance, (b) detector models' detection time.

As shown in Fig. 3, models were assessed based on success criteria requiring accuracy above 75%. The top-performing model was further tested on additional datasets to validate performance under resource constraints. On the COCO dataset, YOLOv8 achieved the highest accuracy at 86%,

while the two-stage detector, Faster RCNN, recorded 52%, suggesting that additional data could improve its Region Proposal Network (RPN) performance.

For precision, Faster RCNN led with 87% on COCO, followed by SSD at 78% and YOLOv8 at 64%, indicating YOLOv8's higher susceptibility to false positives. For recall, YOLOv8 excelled with 81%, followed by SSD at 72% and Faster RCNN at 62%, highlighting the strength of one-stage detectors in recall-critical applications, such as person recognition for security and pedestrian detection [6, 21, 39-41]. Due to its computational complexity, Faster RCNN's mean Average Precision (mAP) could not be calculated. YOLOv8 achieved the highest F1 score of 71% on COCO, reflecting a strong balance between precision and recall, followed closely by SSD at 71%, while Faster RCNN scored 52%, limited by dataset size and diversity.

YOLOv8 demonstrated superior speed, achieving 41.32 FPS on COCO, making it ideal for real-time applications, particularly on mobile devices. In contrast, Faster RCNN, with a more complex architecture, recorded 12.46 FPS [19, 30, 41, 45]. Fig. 3 illustrates the performance comparison between one-stage (YOLOv8) and two-stage (Faster RCNN) models, confirming YOLOv8's advantage in both speed and accuracy.

Table 5. YOLOv8 model performance across the COCO and LFW datasets

Dataset	Accuracy	Precision	Recall	F1 score	mAP	FPS	AUC
COCO	86%	65%	52%	71%	71%	41.32fps	88%
LFW	83%	71%	82%	83%	78%	15.60fps	69%

To further evaluate YOLOv8's efficiency and robustness, we also examined it on the LFW dataset, with the results presented in Table 5. The results, compared to the COCO dataset results, revealed that the model dropped by 3% in terms of accuracy when evaluated on the LFW dataset. However, it achieved better precision, recall, F1 score, and mAP with 71%, 82%, 83%, and 78%, respectively. Although the FPS and area under the curve (AUC) were lower at 15.60s and 69%, the evaluation demonstrates the effectiveness of YOLOv8 in terms of performance and speed, making it a suitable choice for real-time applications.

4.2. Comparison with existing methods

In this study, we also compared our findings with prior studies in the field. As presented in Table 6, the results indicate that the evaluated models, particularly YOLOv8, outperformed others in accuracy and recall, affirming that FD remains a dynamic and advancing domain. For instance, prior studies reported accuracy rates of 92%, precision of 94% and 79%, recall of 84% and 87%, and a mean mAP of 91% for their top-performing YOLOv8 models [34, 39, 48]. These studies used the Face Mask Dataset [48], the CelebA dataset from Kaggle [34], and the FDDB dataset [39]. By contrast, our YOLOv8 model achieved an accuracy of 86%, precision of 64%, recall of 81%, mAP of 71%, and an inference speed of 41.32 FPS. These results reflect the growing adoption of YOLO-based object detection frameworks, though differences emerged: our model showed a 6% lower accuracy, 30% lower precision, 6% higher recall, and 20% lower mAP compared to the cited studies, which did not report inference speed.

Table 6: Comparative Analysis of Results

Study	Model	Datasets	Accuracy	Precision	Recall	mAP	FPS
[48]	YOLOv8	FMD	-	0.94	0.84	0.91	-
[34]	YOLOv8	CelebA	-	0.79	0.87	0.91	-
[39]	YOLOv8	FDDB	0.92	-	-	-	-
This Work	YOLOv8	COCO, LFW	0.86	0.64	0.81	0.71	41.32

While our model is competitive, opportunities for improvement exist. Future advancements, such as expanding dataset size and quality, leveraging high-performance GPUs, and improving

processing speeds, could further enhance model effectiveness and contribute to the evolution of face detection methods.

4.3. Evaluation on sampled images

This subsection assessed the effectiveness of the models utilising sample images, with the results visualised in Figs. 4 to 6. In Fig. 4, we present the performance of the YOLOv8 model, demonstrating its robustness. In a sample image featuring six people, YOLOv8 achieved an accuracy of 91% to 94% in detecting the "person" class. However, the training loss decreased from 1.224 to 0.142 by the 10th epoch, indicating a significant improvement in pattern learning. The confusion matrix reveals 129 TPs, 86 TNs, 21 FPs, and 14 FNs. Additionally, the ROC curve achieved an AUC of 0.88, highlighting the model's ability to differentiate between classes. Similarly, Fig. 5 illustrates the performance of the SSD model. It correctly detects all four faces in a sample image, with the training loss ranging from 1.656 to 0.162 by the 10th epoch. The confusion matrix shows 81 TPs, 81 TNs, 36 FPs, and 52 FNs. Accordingly, its ROC curve achieves an AUC of 0.71, indicating that while the model performs reasonably well, there is potential for further improvement by fine-tuning.

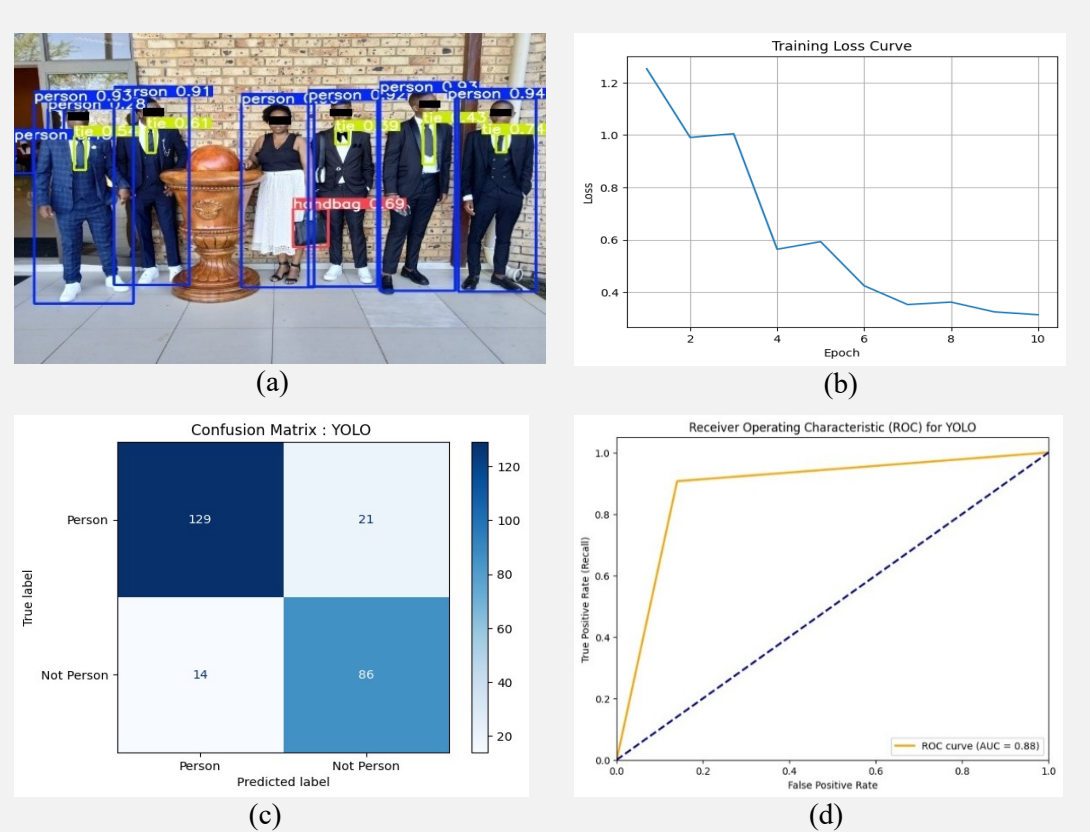


Figure 4. YOLOv8 Performance (a) Sample image (b) Train Loss curve (c) Confusion matrix (d) ROC curve

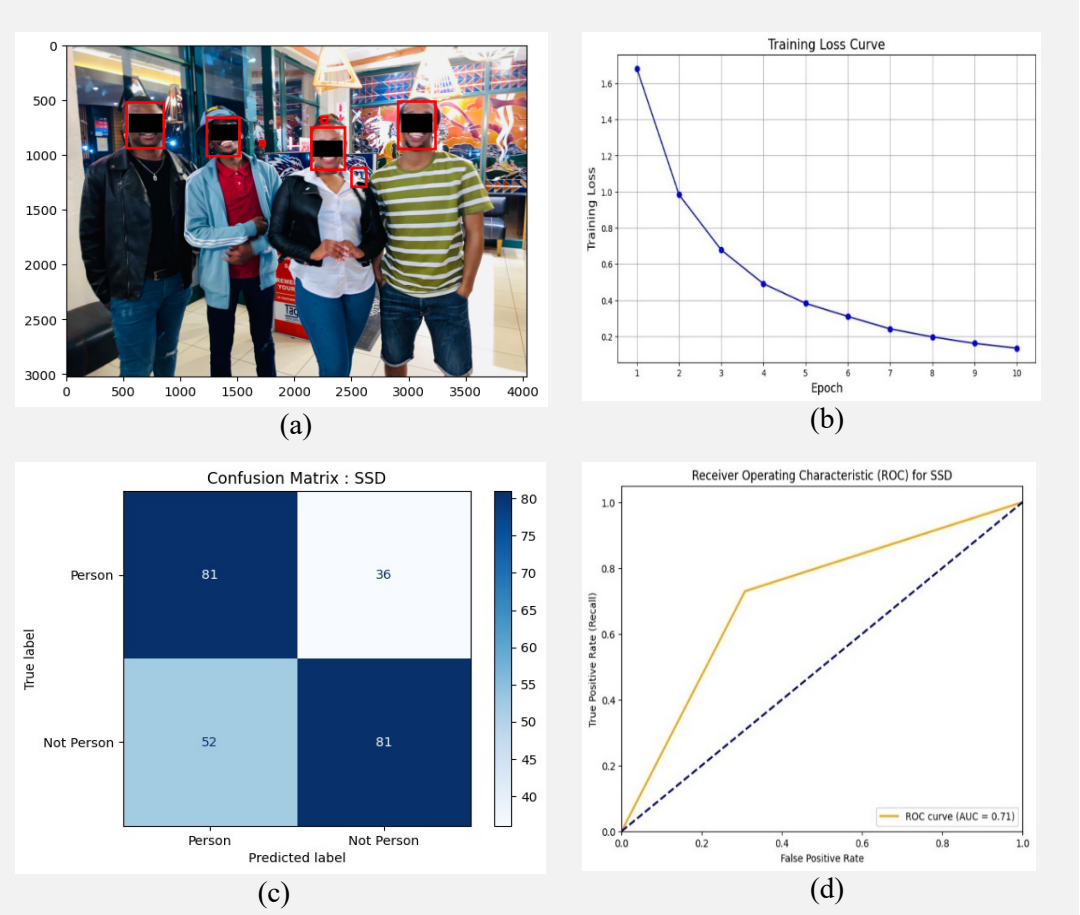
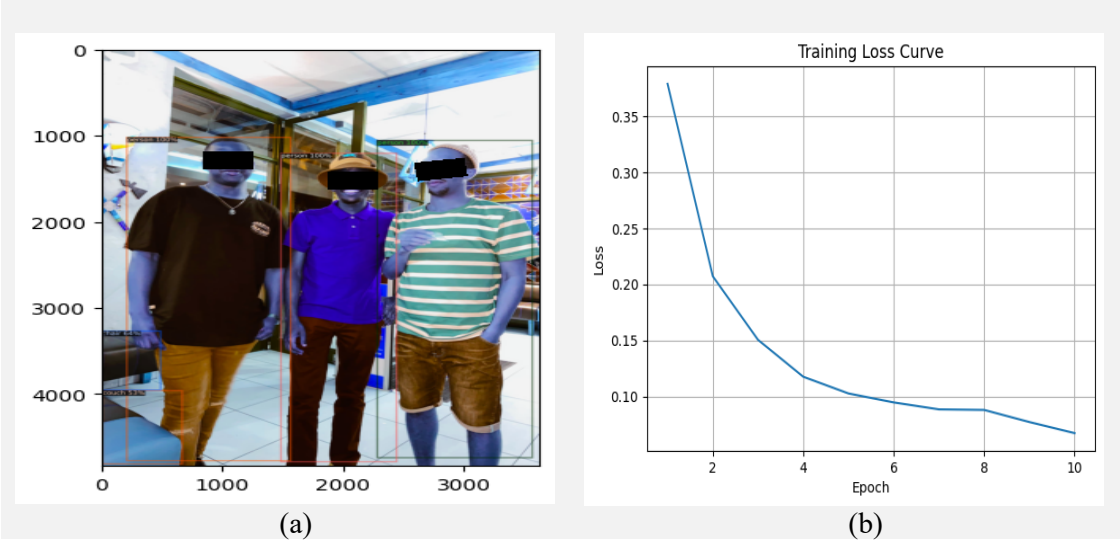


Figure 5. SSD Performance (a) Sample image (b) Train Loss curve (c) Confusion matrix (d) ROC curve

Furthermore, Fig. 6 demonstrates the robustness of the Faster RCNN model's performance, achieving 99% accuracy in detecting the 'person' class in a sample image with three people. The training loss decreases from 0.35 to 0.02 by the 10th epoch, indicating improved pattern learning. Similarly, the confusion matrix shows 65 TPs and 48 TNs, but also 72 FPs and 14 FNs. The ROC curve achieved an AUC of 0.61, which suggests that the model could benefit from hyperparameter tuning to reduce incorrect predictions.



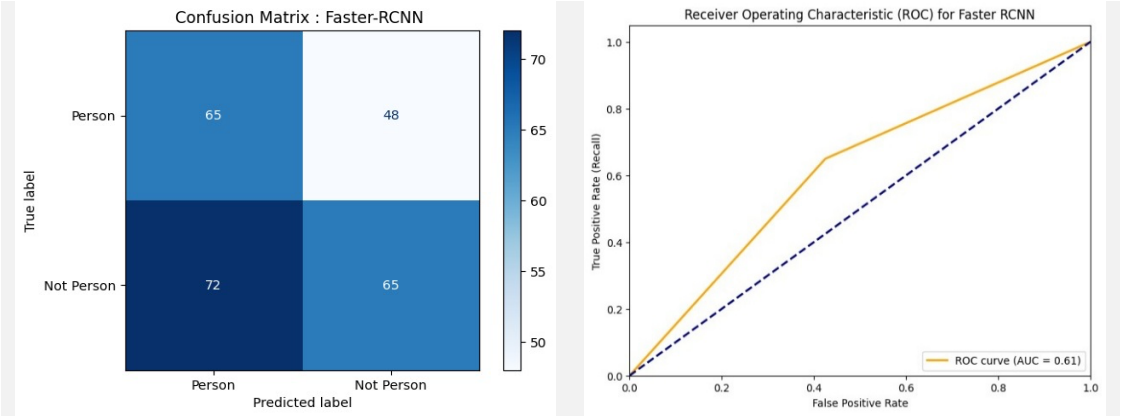


Figure 6. Faster RCNN Performance (a) Sample image (b) Train Loss curve (c) Confusion matrix (d) ROC curve

4.4. Real-time evaluation

This subsection assesses the real-time applicability of the trained models on mobile devices. Given its superior performance, the YOLOv8 model was specifically tested for real-time usage on a mobile computer under various conditions, including different poses, lighting, and skin tones. Fig. 7 visualises the detection accuracy under these diverse conditions. The model obtained an accuracy of 92% for a standard image with typical pose, expression, and lighting conditions. Moreover, tested under varying lighting conditions, the accuracy varied: 86% for the person in normal lighting, 60% for a person with an occluded face (wearing a hat and glasses), 56% for a person with a hat and a distant face, and 73% for another person in a different pose and lower lighting. These results demonstrate that the YOLOv8 model consistently maintains an accuracy above 50% even in challenging situations, indicating its robustness and suitability for real-time applications on mobile devices.

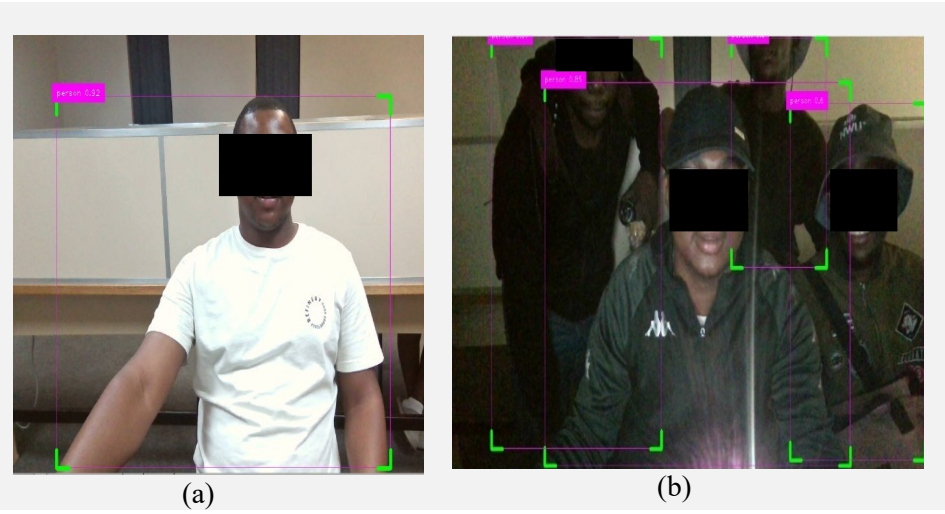
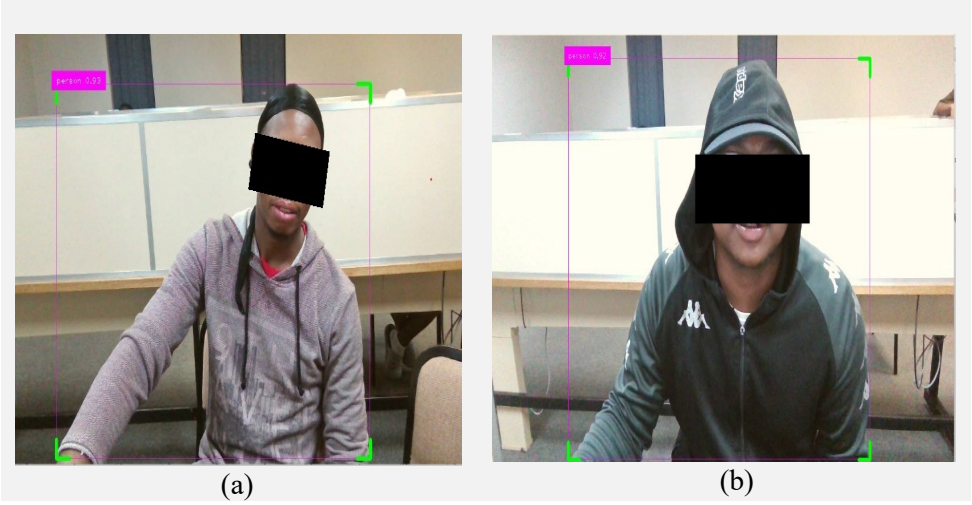


Figure 7. YOLOv8 performance in real-time (a) Normal Picture Sample (b) Bad Lighting Sample

Similarly, in Fig.8, pose and occlusion were tested using two different scenarios. For a person with a slant pose and a smiling expression, the detection accuracy of 93% was achieved. In another test with an occluded face (with glasses and a jacket hood), the model achieved 92% accuracy. These scenarios demonstrate the reliability, speed, and accuracy of the YOLOv8 model in real-time FD. The detection simulation was conducted using the computer's built-in camera, revealing its potential for mobile device applications.



(a) (b)

Figure 8. YOLOv8 performance in real-time (a) Pose Picture Sample (b) Occlusion Sample

As presented above, this study highlights the advancements in detection models, focusing on their strengths and limitations. In terms of balancing accuracy and speed, YOLOv8 achieves this balance well, offering high accuracy and impressive speed, suitable for real-time applications. However, this often involves trade-offs, such as in YOLOv8's slight precision sacrifices, which can lead to FPs. In contrast, Faster RCNN prioritises accuracy and precision but is slower due to its complex architecture, making it less effective for time-critical tasks. Moreover, SSD offers versatility and provides better performance with fine-tuning. The real-world tests demonstrate YOLOv8's robustness in maintaining accuracy above 50% under different conditions. This highlights the need for model selection based on specific requirements, and fine-tuning or hardware optimisation to enhance performance.

5. CONCLUSION

In this study, we evaluated and presented the performance of DL-based models for FD in real-time mobile applications across various datasets. The performance was thoroughly documented and visualised as presented in this paper. The findings revealed that YOLOv8 performance demonstrated an optimal balance between speed and accuracy on the COCO dataset. Further evaluations on the LFW dataset further confirmed YOLOv8’s robustness, supported by optimised hyperparameters. However, while YLOv8 showed promising results, there is still room for improvement. Thus, to advance FD efficacy, our future study will focus on:

1. Expand training datasets to include diverse demographics, poses, and lighting conditions, addressing limitations observed in Faster RCNN’s performance due to dataset constraints.
2. Enhance YOLOv8’s precision through hyperparameter refinement and lightweight architectures, ensuring efficiency on mobile platforms.
3. Explore hybrid models integrating YOLOv8’s real-time speed with Faster RCNN’s precision to enhance FD for complex scenarios.
4. Explore the latest versions of the YOLO detector model and other two-stage detector models such as MTCNN.

These strategies aim to enhance FD models' generalizability and performance, improving their impact in security, healthcare, and human-computer interaction.

ACKNOWLEDGEMENTS

This research was supported by the UDSC, the Department of Computer Science at the North-West University Mafikeng campus.

REFERENCES

[1] K. Dang and S. Sharma, "Review and comparison of face detection algorithms," in *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, 2017, pp. 629-633.

[2] B. Kranthikiran and P. Pulicherla, "Face detection and recognition for use in campus surveillance," *International Journal of Innovative Technology and Exploring Engineering,* vol. 9, pp. 2908-2913, 2020.

[3] N. Zhang, J. Luo, and W. Gao, "Research on face detection technology based on MTCNN," in *2020 International Conference on Computer Network, Electronics and Automation (ICCNEA)*, 2020, pp. 154-158.

[4] M. Shi and Y. Gao, "Lightweight real-time face detection method based on improved YOLOv4," in *2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI)*, 2021, pp. 273-277.

[5] X. Liu, Y. Zhang, Z. Wang, and J. Yang, "Research on Deep Learning Model and Optimisation Algorithm in Edge Computing," in *2023 5th International Conference on Applied Machine Learning (ICAML)*, 2023, pp. 242-246.

[6] F. Majeed, F. Z. Khan, M. Nazir, Z. Iqbal, M. Alhaisoni, U. Tariq*, et al.*, "Investigating the efficiency of deep learning based security systems in a real-time environment using YOLOv5," *Sustainable Energy Technologies and Assessments,* vol. 53, p. 102603, 2022.

[7] M. Wieczorek, J. Siłka, M. Woźniak, S. Garg, and M. M. Hassan, "Lightweight convolutional neural network model for human face detection in risk situations," *IEEE Transactions on Industrial Informatics,* vol. 18, pp. 4820-4829, 2021.

[8] S. S. Phatak, H. S. Patil, M. W. Arshad, B. Jitkar, S. Patil, and J. Patil, "Advanced face detection using machine learning and AI-based algorithm," in *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, 2022, pp. 1111-1116.

[9] Y. Guo and B. C. Wünsche, "Comparison of face detection algorithms on mobile devices," in *2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 2020, pp. 1-6.

[10] R. Ranjan, A. Bansal, J. Zheng, H. Xu, J. Gleason, B. Lu*, et al.*, "A fast and accurate system for face detection, identification, and verification," *IEEE Transactions on Biometrics, Behaviour, and Identity Science,* vol. 1, pp. 82-96, 2019.

[11] A. K. Sirivarshitha, K. Sravani, K. S. Priya, and V. Bhavani, "An approach for face detection and face recognition using OpenCV and face recognition libraries in Python," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2023, pp. 1274-1278.

[12] S. Reddy, S. Goel, and R. Nijhawan, "Real-time face mask detection using machine learning/deep feature-based classifiers for face mask recognition," in *2021 IEEE Bombay Section Signature Conference (IBSSC)*, 2021, pp. 1-6.

[13] M. Anand and S. Babu, "Multi-class facial emotion expression identification using DL-based feature extraction with classification models," *International Journal of Computational Intelligence Systems,* vol. 17, p. 25, 2024.

[14] M. K. Hasan, M. S. Ahsan, S. S. Newaz, and G. M. Lee, "Human face detection techniques: A comprehensive review and future research directions," *Electronics,* vol. 10, p. 2354, 2021.

[15] Y. Zennayi, S. Benaissa, H. Derrouz, and Z. Guennoun, "Unauthorised access detection system to the equipment in a room based on the person's identification by face recognition," *Engineering Applications of Artificial Intelligence,* vol. 124, p. 106637, 2023.

[16] S. J. Prince, J. Elder, Y. Hou, M. Sizinstev, and E. Olevskiy, "Towards face recognition at a distance," in *2006 IET Conference on Crime and Security*, 2006, pp. 570-575.

[17] A. Figueira and B. Vaz, "Survey on synthetic data generation, evaluation methods and GANs," *Mathematics,* vol. 10, p. 2733, 2022.

[18] T. He, R. Kong, A. J. Holmes, M. Nguyen, M. R. Sabuncu, S. B. Eickhoff*, et al.*, "Deep neural networks and kernel regression achieve comparable accuracies for functional connectivity prediction of behaviour and demographics," *NeuroImage,* vol. 206, p. 116276, 2020.

[19] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sensing,* vol. 13, p. 89, 2020.

[20] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," in *Journal of Physics: Conference Series*, 2020, p. 012033.

[21] M. B. Ullah, "CPU based YOLO: A real time object detection algorithm," in *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 552-555.

[22] R. Rameswari, S. N. Kumar, M. A. Aananth, and C. Deepak, "Automated access control system using face recognition," *Materials Today: Proceedings,* vol. 45, pp. 1251-1256, 2021.
[23] R. Fatima, R. Sadiq, I. Ullah, S. Manzoor, S. A. Memon, and U. Khan, "Multiple passive-sensor distributed target tracking approach with machine learning feedback," *Expert Systems with Applications,* vol. 238, p. 122344, 2024.
[24] A. Fernández, R. Usamentiaga, J. L. Carús, and R. Casado, "Driver distraction using visual-based sensors and algorithms," *Sensors,* vol. 16, p. 1805, 2016.
[25] A. Aldhaheri, F. Alwahedi, M. A. Ferrag, and A. Battah, "Deep learning for cyber threat detection in IoT networks: A review," *Internet of Things and cyber-physical systems,* vol. 4, pp. 110-128, 2024.
[26] S. Alfattama, P. Kanungo, and S. K. Bisoy, "Face Recognition from Partial Face Data," in *2021 International Conference in Advances in Power, Signal, and Information Technology (APSIT)*, 2021, pp. 1-5.
[27] B. Amirgaliyev, M. Mussabek, T. Rakhimzhanova, and A. Zhumadillayeva, "A Review of Machine Learning and Deep Learning Methods for Person Detection, Tracking and Identification, and Face Recognition with Applications," *Sensors,* vol. 25, p. 1410, 2025.
[28] J. Ahmad, S. Akram, A. Jaffar, Z. Ali, S. M. Bhatti, A. Ahmad*, et al.*, "Deep learning empowered breast cancer diagnosis: Advancements in detection and classification," *Plos one,* vol. 19, p. e0304757, 2024.
[29] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, "A survey and performance evaluation of deep learning methods for small object detection," *Expert Systems with Applications,* vol. 172, p. 114602, 2021.
[30] Y. Liu, "An improved faster R-CNN for object detection," in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, 2018, pp. 119-123.
[31] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters,* vol. 23, pp. 1499-1503, 2016.
[32] M. Phankokkruad and P. Jaturawat, "An evaluation of technical study and performance for real-time face detection using Web Real-Time Communication," in *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, 2015, pp. 162-166.
[33] Z. Liu, Q. Qi, S. Wang, and G. Zhai, "A novel approach to the detection of facial wrinkles: Database, detection algorithm, and evaluation metrics," *Computers in Biology and Medicine,* vol. 174, p. 108431, 2024.
[34] D. Al-obidi and S. Kacmaz, "Facial Features Recognition Based on Their Shape and Color Using YOLOv8," in *2023 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2023, pp. 1-6.
[35] C. Zhang, G. Liu, X. Zhu, and H. Cai, "Face detection algorithm based on improved AdaBoost and new haar features," in *2019 12th International Congress on Image and signal processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2019, pp. 1-5.
[36] D. Garg, P. Goel, S. Pandya, A. Ganatra, and K. Kotecha, "A deep learning approach for face detection using YOLO," in *2018 IEEE Punecon*, 2018, pp. 1-4.
[37] K. Wang, X. Peng, J. Yang, D. Meng, and Y. Qiao, "Region attention networks for pose and occlusion robust facial expression recognition," *IEEE Transactions on Image Processing,* vol. 29, pp. 4057-4069, 2020.
[38] X.. Sun, P. Wu, and S. C. Hoi, "Face detection using deep learning: An improved faster RCNN approach," *Neurocomputing,* vol. 299, pp. 42-50, 2018.
[39] J. Guo, Z. Wang, and S. Zhang, "FESSD: Feature enhancement single shot multibox detector algorithm for remote sensing image target detection," *Electronics,* vol. 12, p. 946, 2023.
[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.
[41] R. T. Hasan and A. B. Sallow, "Face detection and recognition using OpenCV," *Journal of Soft Computing and Data Mining,* vol. 2, pp. 86-97, 2021.
[42] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and learning systems,* vol. 30, pp. 3212-3232, 2019.
[43] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu*, et al.*, "SSD: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 2016, pp. 21-37.
[44] D. Demetriou, P. Mavromatidis, P. M. Robert, H. Papadopoulos, M. F. Petrou, and D. Nicolaides, "Real-time construction demolition waste detection using state-of-the-art deep learning methods; single–stage vs two-stage detectors," *Waste Management,* vol. 167, pp. 194-203, 2023.

[45] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, pp. 1137-1149, 2016.

[46] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237-242.

[47] E. L. T. Jun, M.-L. Tham, and B.-H. Kwan, "A Comparative Analysis of RT-DETR and YOLOv8 for Urban Zone Aerial Object Detection," in *2024 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, 2024, pp. 340-345.

[48] C. Dewi, D. Manongga, and E. Mailoa, "Deep Learning-Based Face Mask Recognition System with YOLOv8," in *2024 16th International Conference on Computer and Automation Engineering (ICCAE)*, 2024, pp. 418-422.