International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 5, Sep-Oct 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Optimized Face Detection for Digital Forensics Using YOLO on the WIDER FACE Dataset

Serkan KARAKUŞ[1]*, Mustafa KAYA[2], Adamu Muhammad[2], Akıner ALKAN[1]
[1](Research and Development, Siemens A.S, Istanbul-Turkey
Email: serkan.karakus@siemens.com, akiner.alkan@siemens.com)
[2] (Department of the Digital Forensics Engineering, Fırat University, Elazığ-Turkey
Email: mkaya@firat.edu.tr, adamu.muhammad@rayhaanuniv.edu.ng)

----------**********----------

Abstract:

Face detection is a critical task in digital forensic investigations, supporting essential activities like suspect tracking, victim identification, and multimedia triage. Traditional and even modern deep learning methods often falter under forensic conditions, due to factors like low resolution, motion blur, occlusion, and suboptimal lighting. This study investigates the efficacy of optimized YOLO-based architectures (YOLOv8, YOLOv10, YOLOv12) for forensic face detection by retraining nano, small, and medium variants of each model on a refined subset of the WIDER FACE dataset. The proposed preprocessing approach eliminates images below 640×640 pixels to enhance learning efficiency and detection accuracy. Experimental results demonstrate significant improvements across precision, recall, and inference speed metrics compared to pretrained baselines, with YOLOv12 achieving superior latency and precision scores. The findings highlight the importance of task-specific training and dataset refinement in digital forensics.

Keywords **— Digital forensics, face detection, real-time detection, forensic evidence, YOLO, facial recognition technology.**

----------**********----------

I. INTRODUCTION

In digital forensics, face detection plays a vital role, enabling core functions like suspect tracking, victim identification, and image filtering in vast collections of multimedia evidence. It has seen applications in surveillance video analysis, forensic triage of seized devices, social media investigations and border control systems [1]. Forensic evidence from surveillance and archival sources typically suffer from limitations such as low resolution, motion blur, compression artifacts, occlusion, and suboptimal lighting. These issues significantly challenge the reliability of face detection systems, leading to false negatives and missed detections [2], [3]. Traditional methods using handcrafted features, such as the Viola–Jones detector, show rapid performance degradation in complex scenarios [4]. Advances in deep learning, including one-stage detectors like Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO), have improved robustness but still struggle under harsh forensic conditions like small faces in cluttered scenes [5], [6]. Moreover, processing large datasets quickly is essential in digital forensics, making accuracy and speed critical factors [7].

A. Problem Statement

Forensic practitioners require face detection algorithms that can handle poor-quality inputs while delivering fast and reliable results. However, existing detectors face several setbacks, including:

1) Low-resolution degradation

Faces captured in surveillance contexts often occupy few pixels, leading to substantial drops in detection and recognition accuracy [8], [9].

2) Complex environmental factors

Variations in pose, occlusion, illumination, and dynamic backgrounds further hinder detection capabilities [2], [10].

3) Operational constraints

Forensic workflows must process extensive image collections under tight time constraints, necessitating high-performance detection [7].

These challenges impact the trustworthiness and effectiveness of automated forensic systems, with missed faces equating to missed leads or evidence

B. Motivation

Electronic devices with data storage are crucial in forensic analysis, as they hold extensive information about users and their environment. Devices collected from crime scenes may contain thousands of files, which are analyzed, classified, and used to identify and report criminal data or individuals.

Facial recognition technology (FRT) has become a vital tool in modern digital forensics and law enforcement globally. The International Biometrics + Identity Association recognizes FRT as one of the fastest-growing biometric modalities due to its contactless nature and increasing accuracy powered by deep learning algorithms [11]. In countries like China, the United States, and the United Kingdom, FRT is widely used for both public safety and commercial purposes[12]. This widespread rollout draws criticism over privacy violations and algorithmic bias. In Turkey for example, while FRT has been deployed to reunite victims of earthquake with their families, it has also been used to identify peaceful protesters, including minors with high potential for misidentification [13].

Given the existing gaps in face detection in forensic applications, it is evident that bridging them requires both data-centric and model-centric strategies. On the data part, focusing training on higher-quality images can help learn clearer facial patterns, thereby improving detection reliability [8], [14]. On the model part, leveraging efficient real time detectors like YOLOv8 has shown great potential in forensic applications, balancing speed and accuracy [7]. By leveraging right dataset tailoring and hardware optimization, making such systems more practical tools for digital forensic applications.

C. Contribution

The study significantly improves accuracy face detection in digital forensic applications by retraining 3 versions of the YOLO models of 3 different weights each, to attain higher frequency through dataset refinement, thus making them a new benchmark in forensic face detection.

D. Paper Organization

This paper addresses optimized face detection for digital forensics applications.

Section II presents the relevant literature and previous works that form the basis of this study.

Section III details the methodology used, including the dataset (WIDER FACE dataset), the preprocessing strategies applied, a holistic description of the architecture as well as the hardware infrastructure utilized for training.

Section IV outlines the experimental results obtained by custom training the PyTorch model files on the customised dataset, highlighting key performance improvements compared to the Ultralytics pretrained weights of the various models.

Section V discusses the findings, providing insights into key performance indicators like precision-recall trade-offs, inference efficiency, as well as real world applicability of the models, particularly in digital forensic analysis.

Section VI concludes the study, summarizing contributions and suggesting future directions for improving face detection in forensic investigations.

II. RELATED WORK

Artificial Intelligence (AI) has gained significant traction across diverse disciplines, including digital forensics, where it is increasingly applied to optimize analytical processes. The integration of AI into forensic investigations has been facilitated by the availability of open-source datasets and advances in machine learning architectures, yielding promising results in areas such as object detection and classification. This section reviews existing studies that have employed AI methodologies to address key challenges in digital forensics, highlighting their contributions, limitations, and relevance to this work.

Riadi et al. [15] ran forensic investigation on the Signal Messenger application on Android devices, particularly during the rise of cybercrimes in the COVID-19 era. Using three tools; Belkasoft, Magnet AXIOM, and MOBILedit Forensic Express within the Digital Forensics Research Workshop (DFRWS) framework, they sought to uncover digital evidence. Their research pinpointed several types of evidence, including chats, media, and account data, with Belkasoft Evidence Center delivering superior accuracy at 78.69%. Their findings offer valuable insights for future forensic research on the Signal Messenger application. In a related study, Korkmaz and Boyacı [16] proposed a hybrid speaker recognition model using long short-term memory (LSTM) networks. Their model demonstrated the potential to be applied to audio files obtained in digital forensics for content analysis.

Artificial intelligence has also been employed in analyzing social media content. Abebaw et al. [17] used multi-channel convolutional neural networks (CNNs) to extract features of hate speech from social media, using Support Vector Machine (SVM) for classification. This methodology facilitated anomaly detection from social media data. Channabasava and Raghavendra [18] built a consensus-based ensemble model for social media link prediction using several features. Through methods like cross-correlation and Principal Component Analysis (PCA), they achieved an accuracy of up to 97%. Incorporating logistic regression, decision trees, and deep-neuro algorithms, their model surpassed other methods with a link-prediction accuracy of 98%.

Digital image analysis in forensics is gaining research attention. Piva [19] provides a detailed overview of passive image forensics techniques used to verify the authenticity of digital images without requiring original data. The study focused on procedures such as copy-move forgery, resampling detection, image enhancement, and compression, offering a solid foundation for modern developments in image forensics using machine learning and AI. CNN architectures in this respect have been applied successfully in steganography [20], watermarking [21], SCI-camera information detection [22], and copy-move forgery [23].

The application of object detection and recognition tasks to image and video files acquired as forensic evidence is gaining momentum in recent years. One of such innovative approaches, proposed by Javed and Jalil [24], is a byte-level object identification method for the forensic examination of digital images. The method deciphers the byte code of each pixel in an image and identifies objects based on their unique byte code.

Facial recognition tasks have also been broadly explored in computer science and digital forensics. Zafeiriou et al. [25] provided a comprehensive discussion on deep learning-based face recognition technologies, datasets, deep learning architectures, and performance, along with future projections. Viola and Jones [4] examined face detection and recognition tasks, exploring the challenges, algorithmic use, and success rates, as well as strategies for minimizing the False Positive Rate (FPR) and to develop real-time applications. Bledsoe's seminal work [26] was pivotal in the development of face recognition technology, proposing the "model method," in which a mathematical model of a person's face would be constructed and compared with other faces. Following this, several classical face recognition applications were developed, including EigenFace [27], FisherFace [28], BayesianFace [29], MetaFace [30], LaplancianFace [31], and Support Vector Machine (SVM)[32].

The advent of deep learning algorithms and advanced graphics cards, particularly after 2010, has significantly impacted face recognition technology. Krizhevsky et al.'s success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with the AlexNet network was a breakthrough[33]. This network used deep neural networks and was trained on graphics cards with parallel processing capability.

Subsequent studies have utilized CNN architectures for face detection tasks. For example, the DeepFace model by Taigman et al. [34], which was created by training a 9-layer CNN model on four million images achieved a performance comparable to human image detection, with a success rate of 97.53%. Sun et al. [35] demonstrated that the DeepID model, trained with a CNN architecture, could perform face recognition in 10,000 classes.

More recent models such as FaceNet [36], VGGFace [37], VGGFace2 [38], and ArcFace [41] have utilized CNN architectures. These models are trained on face data and have weight files that can easily extract unique features of face data.

A more recent study, by Mei and Zhu et al. [40] presents YOLO-AFR, an improved version of YOLOv12 tailored for small and occluded face detection in complex environments. Incorporating novel modules—Feature Reweighting Fusion Network (FRFN), Scale-Consistent Convolution (SC-Conv), and Shared Enhanced Attention Module (SEAM). The YOLO-AFR architecture enhances multi-scale feature representation and cross-scale prediction consistency while mitigating occlusion errors. Evaluated on standard benchmarks, YOLO-AFR demonstrates higher average precision (AP) than baseline YOLOv12 networks, particularly excelling in small-face detection with lower computational overhead [40].

Our prior work by Karakuş et al. [7], which includes two of the current authors, established a benchmark framework for real-time face detection and identification tailored to digital forensic applications. In that study, YOLOv8 object detection models spanning nano to extra-large variants were trained on the WIDER FACE dataset to achieve high-precision face detection across vast forensic image and video archives. The models delivered exceptional results, with mean Average Precision (mAP) values ranging from 97.51% to 99.03%, significantly outperforming YOLOv5 by 7.1% to 8.8%. The system further integrated a VGGFace2-based feature extractor to support suspect identification using cosine similarity, and a desktop application was developed to facilitate real-time analysis by forensic experts. This work serves as a reference standard for applying YOLO architectures in forensic image analysis, demonstrating the effectiveness of scalable models and practical tools in addressing the operational challenges of modern digital forensics.

The reviewed literature reflects significant progress in object and face detection, with deep learning models, particularly the YOLO family, offering a promising balance between accuracy and speed. However, forensic-specific requirements such as low-quality input tolerance, real-time analysis, and operational scalability remain partially unmet. While our previous study set a strong foundation by applying YOLOv8 to forensic evidence, the current research builds upon that benchmark by refining data quality through resolution-based filtering and extending model comparison to newer YOLO versions. This positions the present work to further advance the state-of-the-art in forensic face detection by improving precision, adaptability, and deployment readiness

III. MATERIALS AND METHODS

This study follows a methodological framework that is structured to optimize face detection for forensic applications using retrained variants of the YOLO object detection algorithm. The process begins with the WIDER FACE dataset, an extensively benchmarked and diverse face dataset which was then pre-processed using OpenCV to improve model performance and attain high precision in forensic applications, by filtering out low-resolution images below 640 × 640 pixels.

Training deep learning models, especially convolutional architectures like YOLO, is resource intensive. Traditional Central Processing Units (CPUs) lack the necessary parallelism required for efficient training. Therefore, dedicated hardware equipped with Graphics Processing Unit (GPU) was employed. The YOLOv8 and YOLOv10 models were trained on a local workstation configured with the necessary requirements, while the YOLOv12 model was trained using the Google Colab Pro platform, leveraging its strong cloud-based infrastructure.

The following subsections provide more details on the dataset used, preprocessing strategy, the infrastructure used and the Yolo architecture.

A. Dataset

A good number of datasets have driven advances in object and face detection. Microsoft's COCO dataset features 330,000 images with 1.5 million object instances across 80 categories and is widely adopted for training multi-object detectors in cluttered scenes [41]. PASCAL VOC, another foundational dataset features around 11,000 images

with around 27,000 annotated objects in 20 classes, but is not commonly used for facial detection tasks due to its limited scope [42]. For face-specific detection, datasets such as the FDDB that features 5,171 faces in 2,845 images and the AFW that features 205 images with 473 faces were early benchmarks but are constrained in scale and diversity [43]. Larger datasets like AFLW containing 25,000 faces and IJB-A containing 500 subjects with 5,712 images and 2,085 videos improve on pose and demographic variety, yet they lack comprehensive annotations for occlusion, blur, and fine-grained facial expressions, which are critical in real-world environments [44], [45]. While CelebA and VGGFace2 are widely adopted for face recognition due to identity and attribute labeling, they do not provide dense bounding box annotations optimized for detection [38], [46]. The WIDER FACE dataset, in contrast, includes 32,203 images and 393,703 labeled faces spanning 61 event categories and is specifically constructed to challenge detectors with variations in scale, pose, occlusion, and lighting [47], thereby being more suitable for this work.

Wider Face dataset is organized into three subsets; training (40%), validation (10%), and testing (50%). Samples are also categorized into Easy, Medium, and Hard levels based on detection difficulty. These features make it suitable for challenging real-world applications, including digital forensics, where robustness to diverse imaging conditions is essential. Individual faces in the dataset are labelled with tight bounding boxes and metadata regarding occlusion or blurriness. This granular annotation supports effective training and evaluation of deep learning models designed for fine-grained face detection [47].

Fig. 1 Wider face dataset sample

B. Data Preprocessing

A resolution-based filtering was applied to the train and validation subsets, due to the forensic application's requirement for high accuracy and minimal false positives. The dataset was filtered to images of minimal dimension of 640 x 640 pixels. This led to the train set sliced down to 12102(37.5% of Wider Face dataset) images and the validation set to 2853 (8.9% of Wider Face) images. This resolution threshold is in line with standard input sizes recommended for convolutional neural networks (CNNs) used in object detection frameworks like YOLO, ensuring reliability in detection and computational efficiency [5], [6].

Images with low resolution are known to negatively impact the performance of face detectors, particularly when faces appear small or occluded. Prior studies have found out that model accuracy declines significantly when input image resolution falls below recommended levels, especially in datasets with dense face clusters [48]. Therefore, applying a minimum size threshold enables the model to focus on clearer, information-rich samples, enhancing learning during retraining.

Filtering was done using the OpenCV library in Python, reading and evaluating each image by its height and width. This gave room for removing unsuitable images, i.e. those with dimensions lower than 640 x 640 pixels. Filtering the training and validation set ensured that the pretrained models learn more discriminative features relevant to forensic imagery which mostly demand high precision.

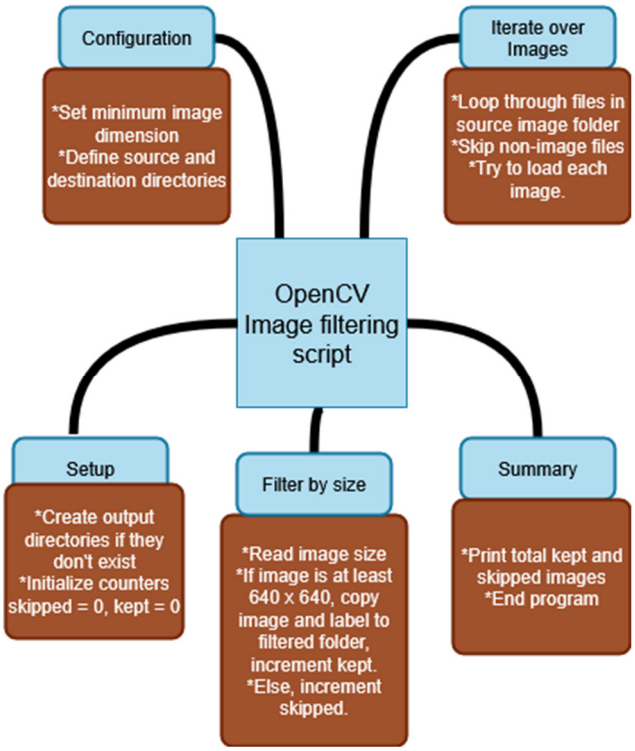

Fig. 2 Dataset preprocessing with OpenCV

C. The YOLO Architecture

The YOLO (You Only Look Once) architecture is a family of detectors that treat object detection as a single end to end regression problem. A YOLO model typically divides an input image in to an S x S grid. Each grid cell is then tasked to predict bounding boxes and class probabilities for objects whose centers fall into the cell. Unlike two-stage detectors like R-CNN(), which generates region proposals before classification, YOLO processes the image in a single forward pass, thereby being extremely fast and efficient for real-time applications [49].

1. Core Components of YOLO

The YOLO architecture comprises of three main components, the backbone, the neck and the head.

Backbone: The backbone typically, is a convolutional neural network (CNN), used to extract hierarchical features from the input image. In YOLOv3–YOLOv5, the backbone was often Darknet-53 or CSPDarknet53. YOLOv8 and later versions use optimized backbones like CSPDarknet-A or PP-YOLOE [6], [50].

Neck: This neck combines feature maps from different depths of the network to improve object localization at multiple scales. Common neck modules include the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN), which enhance semantic flow across layers [51].

Head: The head is in charge of final predictions. It outputs bounding box coordinates, objectness scores, and class probabilities. Modern YOLO versions (YOLOv8+) adopt an anchor-free detection head, where the model directly predicts center points and box sizes, thereby reducing complexity and increasing generalization across datasets [50], [52].

2. Training Protocols and Loss Functions

Training is augmented with techniques like Mosaic augmentation, MixUp, and color jittering, which improve generalization and robustness to diverse conditions. The loss function combines Localization Loss (e.g., CIOU or DIOU) for bounding box accuracy, Confidence/Objectness Loss to penalize incorrect object presence predictions, and Classification Loss for multi-class detection accuracy [50], [53]. Non-Maximum Suppression (NMS) is then applied, after inference to filter out overlapping boxes based on their Intersection over Union (IoU) scores

3. Evolution of the YOLO architecture

Since its introduction in 2016, YOLO has undergone significant architectural transformations, each improving detection performance, speed, and efficiency.

YOLOv1 unveiled the single-stage detection paradigm by using a unified convolutional architecture to predict bounding boxes and class probabilities from full images in a single pass [49]. YOLOv2 introduced anchor boxes, batch normalization, and multi-scale training to improve accuracy and localization [54]. YOLOv3 adopted a deeper Darknet-53 backbone and incorporated multi-scale predictions using feature pyramids for better small-object detection [54]. YOLOv4 combined CSPDarknet as the backbone with advanced data augmentation and bag of freebies, significantly boosting mAP and speed [6].

YOLOv5, although not released by the original authors, became widely adopted due to its modular PyTorch implementation and continued improvements to speed and usability [50].

YOLOv7 introduced E-ELAN, re-parameterized convolutional modules, and model scaling strategies that optimized both speed and accuracy in real-time settings [50].

YOLOv8 moved toward an anchor-free detection head and implemented an improved backbone with better generalization and training speed, packaged into a unified PyTorch framework [50].

YOLOv10 further improved efficiency and latency by integrating a bi-level routing attention mechanism and a simplified detection head. It adopted a decoupled head structure and retained anchor-free operation, making it particularly suitable for resource-constrained environments [40].

YOLOv12, the latest of the YOLO family, leverages lightweight attention modules and transformer-based enhancements for long-range feature modeling, delivering robust performance in complex, cluttered scenes [55].

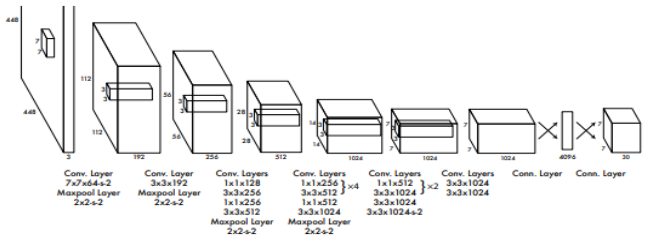

Fig. 3 YOLO generic architecture[49]

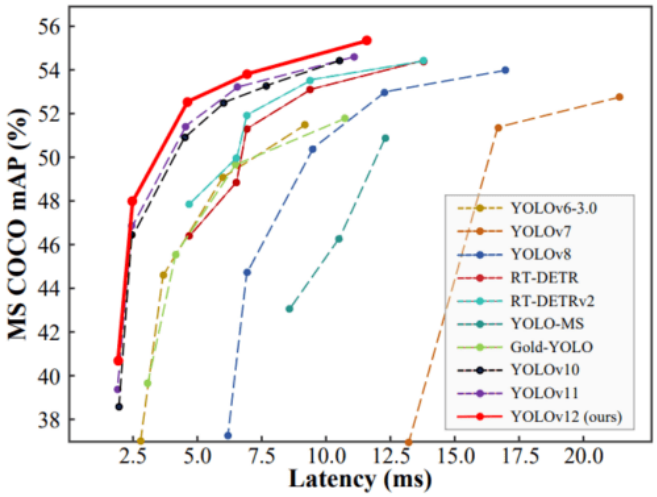

Fig. 4 YOLO models and other related detectors comparisons in terms of latency-accuracy (left) and FLOPs-accuracy (right) trade-offs [55].

D. Forensic Face Detection Workflow

Our implementation of YOLO models for face detection follows a five-step pipeline.

In the first stage, images are preprocessed, resized to 640x640 and normalized.

In the second stage, the Backbone extracts feature maps from the normalized image. The Neck, in the third stage, the feature aggregation stage, following a combination of Feature Pyramid Network and Path Aggregation Network (FPN + PAN) computation, then fuses features across multiple scales.

In the fourth stage, the prediction stage, the detection head predicts bounding boxes, objectness scores, and class labels in an anchor-free manner.

Finally, in the post-processing stage, the NMS (Non-maximum suppression) removes redundant boxes, leaving only high-confidence predictions.

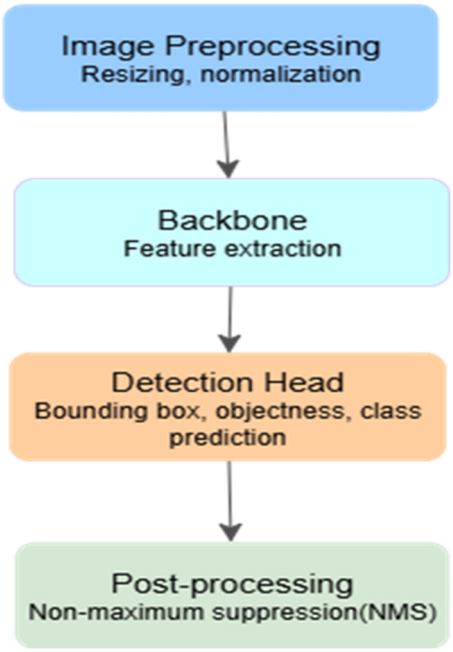

Fig. 5 Forensic Face Detection Pipeline

E. Hardware

Training AI models is resource-intensive; therefore, traditional CPUs cannot handle the task. The task requires hardware capable of parallel processing, such as Graphics Processing Units (GPUs). Additionally, updating the weights requires images to be loaded into the RAM and then transferred to the GPU, as shown in [7].

In this research, the different weights of YOLO versions 8 and 10 were trained on a local workstation at Firat University's digital forensics lab with an Intel Core i7 processor, an NVIDIA RTX 2070 SUPER GPU (8 GB VRAM) with a compute capability of 7.5 [56], capable of approximately 9.1 TFLOPS single-precision (FP32) and 72 TFLOPS tensor-core performance [57], and 32 GB DDR6 RAM. The YOLOv12 models were trained using Google Colab on an NVIDIA A100 Tensor Core GPU (40 GB VRAM), with a compute power of 8.0 [56], offering up to 19.5 TFLOPS FP32, 312 TFLOPS FP16, and 156 TFLOPS TensorFloat32 (TF32) performance, along with 9.7 TFLOPS FP64 capabilities, with 25 GB system RAM[57].

IV. EXPERIMENTAL RESULT

This section presents a comprehensive evaluation of YOLOv8, YOLOv10, and YOLOv12 models, each retrained on a refined subset of the Wider Face dataset. Performance metrics such as mAP@0.5, precision, recall, inference speed, GFLOPs, and parameter counts are analyzed across Nano, Small, and Medium variants. These results are compared against their respective pretrained baselines to assess improvements in accuracy and efficiency, metrics essential for robust digital forensic applications.

A. Yolov8

The Ultralytics pretrained YOLOv8 models achieved inference speeds of 0.99, 1.20, and 1.83ms using the coco dataset on an A100 TensorRT GPU. Our training and inference on the refined Wider Face dataset on an NVIDIA RTX 2070 SUPER GPU attained high accuracy and efficiency as shown in table 2 against the original models depicted in Table I.

TABLE I

ULTRALYTICS YOLOV8 MODEL PERFORMANCE METRICS[58]

Model	Size (pixels)	mAPval 50-95 (%)	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	Params (M)	FLOPs (B)
Yolov8n	640	37.3	80.4	0.99	3.2	8.7
Yolov8s	640	44.9	128.4	1.2	11.2	28.6
Yolov8m	640	50.2	234.7	1.83	25.9	78.9

Table I presents the performance metrics of the Ultralytics-pretrained YOLOv8 models, evaluated on the COCO dataset across three model sizes: Nano, Small, and Medium. Inputs of size 640 x 640 were processed, demonstrating progressive improvements in detection accuracy, with mAPval (50–95%) increasing from 37.3% for Nano to 50.2% for Medium. The accuracy gains notably called for higher computational demands, as indicated by increased FLOPs from 8.7B to 78.9B and parameter sizes 3.2M to 25.9M.

TABLE II

CUSTOM TRAINED YOLOV8 PERFORMANCE METRICS

Model	mAP@0.5	Precision	Recall	Inference speed(ms)	GFLOPs	Layers	Parameters (m)
Yolov8n	0.968	0.955	0.904	20.9	8.1	72	3.0
Yolov8s	0.984	0.953	0.938	9.1	78.7	92	25.8
Yolov8m	0.985	0.953	0.94	9.0	78.7	92	25.8

Table II summarizes the performance of the custom-trained YOLOv8 models, Nano, Small, and Medium, after retraining on the refined Wider Face dataset. All weights demonstrate significant improvements in accuracy, with high mAP@0.5 values. Precision and recall scores remain consistently high across all versions, reflecting high robustness.

B. Yolov10

The Ultralytics pretrained YOLOv10 models have latencies of 1.84, 2.49, and 4.74ms. This is based on the coco dataset and a TensorRT FP16 on T4 GPU. Our training and inference on the refined Wider Face dataset on an NVIDIA RTX 2070 SUPER GPU attained high accuracy and efficiency as shown in table 4 against the original models depicted in Table III.

TABLE III

ULTRALYTICS YOLOV10 PERFORMANCE METRICS[40]

Model	Input Size	APval (%)	FLOPs (G)	Latency (ms)
YOLOv10n	640	38.5	6.7	1.84
YOLOv10s	640	46.3	21.6	2.49
YOLOv10m	640	51.1	59.1	4.74

Table III presents the performance metrics of the pretrained YOLOv10 models Nano, Small, and Medium evaluated on the COCO dataset with an input resolution of 640 × 640 pixels. The models show a balance between accuracy and complexity, with Average Precision values increasing as the model size increases. Similarly, the computational cost increases with the increase in weights. FLOPs grow from 6.7G in the Nano model to 59.1G in the medium variant, while latency rises from 1.84ms to 4.74ms, measured using TensorRT FP16 on a T4 GPU.

TABLE IV

CUSTOM-TRAINED YOLOV10 MODELS PERFORMANCE METRICS

Model	mAP@0.5	Precision	Recall	Inference speed(ms)	GFLOPs	Layers	Parameters (m)
Yolov10n	0.96812	0.94985	0.9039	9.3	8.7	223	2.78
Yolov10s	0.98	0.952	0.926	2.4	21.4	106	7.2
Yolov10m	0.984	0.954	0.939	5.0	58.9	136	15.3

Table IV reports the performance of the custom-trained YOLOv10 models Nano, Small, and Medium optimized on the refined WIDER FACE dataset. The three variants all achieved remarkable accuracy, with mAP@0.5 values exceeding 0.96. Precision and recall scores notably remain high. The Small model stands out with the fastest inference time of 2.4ms, making it highly suitable for real-time detection without compromising accuracy.

C. Yolov12

The Ultralytics pretrained YOLOv12 models achieved inference speeds of 1.64, 2.61, and 4.86ms using the coco dataset on a T4 TensorRT GPU. Our training and inference on the refined Wider Face dataset on an NVIDIA A100 Tensor Core GPU attained high accuracy and efficiency as shown in table VI against the original models depicted in table V.

TABLE V

ULTRALYTICS YOLOV12 MODEL PERFORMANCE METRICS[55]

Model	Size (pixels)	mAPval 50-95 (%)	Speed T4 TensorRT (ms)	Params (M)	FLOPs (B)
YOLOv12n	640	40.6	1.64	2.6	6.5
YOLOv12s	640	48	2.61	9.3	21.4
YOLOv12m	640	52.5	4.86	20.2	67.5

Table V outlines the performance metrics of the pretrained YOLOv12 models, Nano, Small, and Medium on the COCO dataset using 640 × 640 pixel inputs. The models display a progressive increase in detection accuracy, with mAPval (50–95%) ranging from 40.6% for the Nano variant to 52.5% for the Medium variant. Inference latency also scales accordingly, from 1.64ms to 4.86ms when tested on a T4 GPU using TensorRT, highlighting the balance between speed and accuracy. Model complexity, reflected in parameter count and FLOPs, increases significantly with model size.

TABLE VI

CUSTOM TRAINED YOLOV12 PERFORMANCE METRICS

Model	mAP@0.5	Precision	Recall	Inference speed(ms)	GFLOPs	Layers	Parameters (m)
Yolov12n	0.959	0.948	0.883	0.5	5.8	376	2.5
Yolov12n	0.978	0.959	0.917	1.1	19.3	376	9.07
Yolov12m	0.984	0.962	0.932	1.9	59.5	402	19.6

Table VI presents the performance of the custom-trained YOLOv12 models, Nano, Small and Medium,

on the refined WIDER FACE dataset. All three variants delivered strong results, with the medium model achieving the highest Mean Average Precision (mAP@0.5) of 0.984 and the best recall of 93.2%, indicating highly reliable detection capabilities. The Small model offered an excellent balance of speed and accuracy, with a low inference time of 1.1ms and precision of 95.9%. The Nano model, while maintaining the smallest parameter count of 2.5M and lowest GFLOPs 5.8, delivered a solid mAP of 0.959 and an exceptionally fast inference speed of 0.5ms.

V. DISCUSSION

The experimental evaluation of the YOLOv8, YOLOv10, and YOLOv12 models on the refined WIDER FACE dataset demonstrates the benefits of applying models in task-specific environments, as in our case of facial recognition for digital forensics.

The pretrained YOLO model weights from Ultralytics provided strong baseline performance metrics, however, custom training significantly enhanced face detection accuracy, precision, recall, and inference speed across all model sizes, nano, small, and medium. These improvements can be attributed to both dataset refinement and the forensics-specific learning objectives. A comparative analysis between the pretrained and custom-trained models is presented, with focus on precision-recall trade-offs, computational efficiency, and inference speeds. This will offer a practical evaluation of the models' suitability for application in investigative scenarios.

A. Yolov8

The experimental results for YOLOv8 models indicate notable improvements in face detection performance when trained on the refined WIDER FACE dataset compared to the official COCO-based benchmarks. While the Ultralytics pretrained models reported mAP scores of 37.3%, 44.9%, and 50.2% for the nano, small, and medium variants respectively, our custom-trained counterparts achieved impressive mAP scores of 96.8%, 98.4%, and 98.5%. These results highlight the effectiveness of dataset filtering and task-specific training in enhancing model accuracy. A point to note is that inference speeds in our configuration showed variations due to hardware differences, the YOLOv8 small model which achieved an inference speed of 1.2ms on an A100 TensorRT GPU achieved 9.1ms on our NVIDIA RTX 2070 SUPER GPU. While this is a higher latency, it is still within an acceptable range for real-time applications and demonstrates strong performance even on mid-range GPUs. Additionally, our models maintained a consistently high precision-recall balance making them much suitable in forensic image analysis.

B. Yolov10

The YOLOv10 model custom training experimental results further underscore the impact of dataset filtering on the models. The Ultralytics pretrained models, evaluated on COCO and a TensorRT T4 environment, achieved mAPval scores of 38.5%, 46.3%, and 51.1% for the nano, small and medium weights of the model. Following the fine-tuning of the Wider Face dataset, the YOLOv10 models achieved mAP@0.5 values of 96.8%, 98.0%, and 98.4%, for the nano, small and medium weights respectively. The custom-trained models further achieved substantial reductions in inference latency, down to 9.3ms for nano, 2.4ms for small, and 5.0ms for medium models, thereby suggesting improved deployment efficiency. Additionally, the recall scores surpassed 93% across all three variants, demonstrating their reliability in identifying subtle and occluded facial features. This confirms the suitability of the models for high-speed forensic applications without compromising precision.

C. Yolov12

The YOLOv12 models delivered the most consistent balance between accuracy and efficiency across all sizes. While the original Ultralytics benchmarks on COCO reported mAPval scores of 40.6%, 48.0%, and 52.5%, our refined versions trained on WIDER FACE reached mAP@0.5 scores of 95.9%, 97.8%, and 98.4%, respectively. The small and medium weights of the YOLOv12 achieved high recall values of 91.7% and 93.2%, indicating robust sensitivity to face instances in complex scenes. Furthermore, latency was reduced substantially, with

the nano version achieving a lightning-fast 0.5ms per inference, outperforming its Ultralytics pretrained counterpart's 1.64ms. The reduction in GFLOPs, especially for the nano model, with 5.8 GFLOPs, reinforces the efficiency of the model even in resource-constrained environments. The YOLOv12 architecture, when custom-trained, stands out as the most optimal solution for forensic applications requiring rapid and precise face detection.

VI. CONCLUSION

A systematic evaluation of the YOLO architectures, v8, v10 and v12, tailored for enhanced face detection in digital forensics was performed. In the process, a filtered Wider Face dataset for high resolution images was leveraged, and three weights, nano, small and medium for each of the YOLO versions 8, 10 and 12 were custom-trained, yielding substantial performance gains in accuracy, precision, recall and inference latency. The research reaffirms the suitability of the models for the Forensic Application they will be integrated in which among other forensic investigation purposes, features a facial recognition module on image and video files retrieved from digital evidence.

Comparative analysis of the three YOLO architectures reveals distinct performance characteristics that inform their suitability for forensic applications. YOLOv12 emerged as the superior model across all evaluated metrics, demonstrating the most consistent balance between accuracy and computational efficiency. The YOLOv12-medium variant achieved the highest mAP@0.5 of 98.4% with a recall of 93.2%, while the YOLOv12-nano delivered exceptional inference speed of 0.5ms with only 5.8 GFLOPs, making it the optimal choice for resource-constrained environments. This superior performance can be attributed to YOLOv12's integration of lightweight attention modules and transformer-based enhancements for long-range feature modeling, which enable robust detection in complex, cluttered scenes typical of forensic evidence. YOLOv10 demonstrated competitive performance with mAP@0.5 values ranging from 96.8% to 98.4% and notably achieved the fastest small-variant inference time of 2.4ms, owing to its bi-level routing attention mechanism and simplified detection head architecture. However, the YOLOv10-nano exhibited relatively higher latency (9.3ms) compared to its YOLOv12 counterpart, suggesting less optimized performance in the smallest configuration. YOLOv8, while still achieving impressive mAP@0.5 scores of 96.8% to 98.5%, exhibited the least favorable performance profile among the three architectures. The YOLOv8-small model recorded an inference time of 9.1ms on the NVIDIA RTX 2070 SUPER GPU, significantly slower than both YOLOv10 (2.4ms) and YOLOv12 (1.1ms) small variants. This performance gap can be attributed to YOLOv8's lack of advanced attention mechanisms and architectural optimizations present in the newer versions. Furthermore, YOLOv8 demonstrated lower recall values (ranging from 89.6% to 91.8%) compared to YOLOv10 (93.0%-94.4%) and YOLOv12 (91.7%-93.2%), indicating reduced sensitivity in detecting subtle and occluded facial features—a critical limitation in forensic scenarios. The comparative evaluation conclusively establishes YOLOv12 as the most optimal solution for forensic face detection applications, offering superior accuracy, efficiency, and deployment flexibility, while YOLOv8, despite its acceptable performance, represents the least suitable option due to higher latency and reduced recall capabilities.

REFERENCES

[1] A. K. Jain, A. Ross, and S. Prabhakar, “An Introduction to Biometric Syst. Video Technol., vol. 14, no. 1, pp. 4–20, Jan. 2004, doi: 10.1109/tcsvt.2003.818349.

[2] Y. Zhou, D. Liu, and T. Huang, “Survey of Face Detection on Low-quality Images,” 2018, arXiv. doi: 10.48550/ARXIV.1804.07362.

[3] O. A. Aghdam, B. Bozorgtabar, H. K. Ekenel, and J.-P. Thiran, “Exploring Factors for Improving Low Resolution Face Recognition,” 2019, arXiv. doi: 10.48550/ARXIV.1907.10104.

[4] P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” Int. J. Comput. Vis., vol. 57, no. 2, pp. 137–154, May 2004, doi: 10.1023/b:visi.0000013087.49260.fb.

[5] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE, July 2017. doi: 10.1109/cvpr.2017.106.

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 23, 2020, arXiv: arXiv:2004.10934. doi: 10.48550/arXiv.2004.10934.

[7] S. Karakuş, M. Kaya, and S. A. Tuncer, “Real-Time Detection and Identification of Suspects in Forensic Imagery Using Advanced YOLOv8 Object Recognition Models,” Trait. Signal, vol. 40, no. 5, pp. 2029–2039, Oct. 2023, doi: 10.18280/ts.400521.

[8] P. Li, L. Prieto, D. Mery, and P. Flynn, “Face Recognition in Low Quality Images: A Survey,” Mar. 28, 2019, arXiv: arXiv:1805.11519. doi: 10.48550/arXiv.1805.11519.

[9] "Effectiveness of State-of-the-Art Super Resolution Algorithms in Surveillance Environment," in Advances in Intelligent Systems and Computing, Cham: Springer International Publishing, 2021, pp. 79–88. doi: 10.1007/978-3-030-74728-2_8.

[10] A. S. Sanchez-Moreno, J. Olivares-Mercado, A. Hernandez-Suarez, K. Toscano-Medina, G. Sanchez-Perez, and G. Benitez-Garcia, "Efficient Face Recognition System for Operating in Unconstrained Environments," J. Imaging, vol. 7, no. 9, p. 161, Aug. 2021, doi: 10.3390/jimaging7090161.

[11] International Biometrics + Identity Association, "IBIA Position on Face Recognition and AI for Law Enforcement," 2021. Accessed: July 15, 2025. [Online]. Available: https://www.ibia.org/download/datasets/6401/IBIA%20Position%20on%20Face%20Recognition%20and%20AI%20for%20Law%20Enforcement.pdf

[12] Biometric Update, "Study Examines the Need to Balance Innovation and Privacy in Facial Recognition," 2025. Accessed: July 15, 2025. [Online]. Available: https://www.biometricupdate.com/202502/study-examines-the-need-to-balance-innovation-and-privacy-in-facial-recognition

[13] Hürriyet Daily News, "Facial Recognition Software Helps to Identify 144 Children," 2023. [Online]. Available: https://www.hurriyetdailynews.com/facial-recognition-software-helps-to-identify-144-children-180914

[14] A. K. Jain, B. Klare, and U. Park, "Face recognition: Some challenges in forensics," in Face and Gesture 2011, Santa Barbara, CA, USA: IEEE, Mar. 2011, pp. 726–733. doi: 10.1109/fg.2011.5771338.

[15] I. Riadi, Herman, and N. H. Siregar, "Mobile Forensic Analysis of Signal Messenger Application on Android using Digital Forensic Research Workshop (DFRWS) Framework," Ingénierie Systèmes Inf., vol. 27, no. 6, pp. 903–913, Dec. 2022, doi: 10.18280/isi.270606.

[16] Y. Korkmaz and A. Boyacı, "Hybrid voice activity detection system based on LSTM and auditory speech features," Biomed. Signal Process. Control, vol. 80, p. 104408, Feb. 2023, doi: 10.1016/j.bspc.2022.104408.

[17] Z. Abebaw, A. Rauber, and S. Atnafu, "Design and Implementation of a Multichannel Convolutional Neural Network for Hate Speech Detection in Social Networks," Rev. Intell. Artif., vol. 36, no. 2, pp. 175–183, Apr. 2022, doi: 10.18280/ria.360201.

[18] U. Channabasava and B. K. Raghavendra, "Ensemble Assisted Multi-Feature Learnt Social Media Link Prediction Model Using Machine Learning Techniques," Rev. Intell. Artif., vol. 36, no. 3, pp. 439–444, June 2022, doi: 10.18280/ria.360311.

[19] A. Piva, "An Overview on Image Forensics," ISRN Signal Process., vol. 2013, pp. 1–22, Jan. 2013, doi: 10.1155/2013/496701.

[20] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-Scale JPEG Image Steganalysis Using Hybrid Deep-Learning Framework," IEEE Trans. Inf. Forensics Secur., vol. 13, no. 5, pp. 1200–1214, May 2018, doi: 10.1109/tifs.2017.2779446.

[21] H. Kandi, D. Mishra, and S. R. K. S. Gorthi, "Exploring the learning capabilities of convolutional neural networks for robust image watermarking," Comput. Secur., vol. 65, pp. 247–268, Mar. 2017, doi: 10.1016/j.cose.2016.11.016.

[22] H. Yao, T. Qiao, M. Xu, and N. Zheng, "Robust Multi-Classifier for Camera Model Identification Based on Convolution Neural Network," IEEE Access, vol. 6, pp. 24973–24982, 2018, doi: 10.1109/access.2018.2832066.

[23] J. Ouyang, Y. Liu, and M. Liao, "Copy-move forgery detection based on deep learning," in 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai: IEEE, Oct. 2017, pp. 1–5. doi: 10.1109/cisp-bmei.2017.8301940.

[24] A. R. Javed and Z. Jalil, "Byte-Level Object Identification for Forensic Investigation of Digital Images," in 2020 International Conference on Cyber Warfare and Security (ICCWS), Islamabad, Pakistan: IEEE, Oct. 2020, pp. 1–4. doi: 10.1109/iccws48432.2020.9292387.

[25] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: Past, present and future," Comput. Vis. Image Underst., vol. 138, pp. 1–24, Sept. 2015, doi: 10.1016/j.cviu.2015.03.015.

[26] W. W., Bledsoe, "The Model Method in Facial Recognition," Panoramic Research, Inc., Palo Alto, California, Technical Report, 1964.

[27] M. Turk and A. Pentland, "Eigenfaces for Recognition," J. Cogn. Neurosci., vol. 3, no. 1, pp. 71–86, Jan. 1991, doi: 10.1162/jocn.1991.3.1.71.

[28] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 7, pp. 711–720, July 1997, doi: 10.1109/34.598228.

[29] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian face recognition," Pattern Recognit., vol. 33, no. 11, pp. 1771–1782, Nov. 2000, doi: 10.1016/s0031-3203(99)00179-x.

[30] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Metaface learning for sparse representation based face recognition," in 2010 IEEE International Conference on Image Processing, Hong Kong, Hong Kong: IEEE, Sept. 2010, pp. 1601–1604. doi: 10.1109/icip.2010.5652363.

[31] Xiaofei He, Shuicheng Yan, Yuxiao Hu, P. Niyogi, and Hong-Jiang Zhang, "Face recognition using Laplacianfaces," IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 3, pp. 328–340, Mar. 2005, doi: 10.1109/tpami.2005.55.

[32] Guodong Guo, S. Z. Li, and Kapluk Chan, "Face recognition by support vector machines," in Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), Grenoble, France: IEEE Comput. Soc, pp. 196–201. doi: 10.1109/afgr.2000.840634.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[34] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA: IEEE, June 2014. doi: 10.1109/cvpr.2014.220.

[35] Y. Sun, X. Wang, and X. Tang, "Deep Learning Face Representation from Predicting 10,000 Classes," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA: IEEE, June 2014, pp. 1891–1898. doi: 10.1109/cvpr.2014.244.

[36] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA: IEEE, June 2015, pp. 815–823. doi: 10.1109/cvpr.2015.7298682.

[37] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," in Procedings of the British Machine Vision Conference 2015, Swansea: British Machine Vision Association, 2015, p. 41.1-41.12. doi: 10.5244/c.29.41.

[38] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A Dataset for Recognising Faces across Pose and Age," in 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an: IEEE, May 2018. doi: 10.1109/fg.2018.00020.

[39] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA: IEEE, June 2019. doi: 10.1109/cvpr.2019.00482.

[40] J. Mei and W. Zhu, "BGF-YOLOv10: Small Object Detection Algorithm from Unmanned Aerial Vehicle Perspective Based on Improved YOLOv10," Sensors, vol. 24, no. 21, p. 6911, Oct. 2024, doi: 10.3390/s24216911.

[41] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," Feb. 21, 2015, arXiv: arXiv:1405.0312. doi: 10.48550/arXiv.1405.0312.

[42] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, June 2010, doi: 10.1007/s11263-009-0275-4.

[43] Jain, Vidit and Learned-Miller, Erik, "FDDB: A Benchmark for Face Detection in Unconstrained Settings," University of Massachusetts Amherst, 2010. [Online]. Available: https://people.cs.umass.edu/~elm/papers/fddb.pdf

[44] M. Kostinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark localization," in 2011 IEEE International Conference on

Computer Vision Workshops (ICCV Workshops), Barcelona, Spain: IEEE, Nov. 2011. doi: 10.1109/iccvw.2011.6130513.
[45] B. F. Klare et al., "Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA: IEEE, June 2015, pp. 1931–1939. doi: 10.1109/cvpr.2015.7298803.
[46] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," in 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile: IEEE, Dec. 2015, pp. 3730–3738. doi: 10.1109/iccv.2015.425.
[47] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A Face Detection Benchmark," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, June 2016. doi: 10.1109/cvpr.2016.596.
[48] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," IEEE Trans. Pattern Anal. Mach. Intell., pp. 1–1, 2021, doi: 10.1109/tpami.2021.3059968.
[49] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, June 2016, pp. 779–788. doi: 10.1109/cvpr.2016.91.
[50] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Mach. Learn. Knowl. Extr., vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
[51] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE, July 2017. doi: 10.1109/cvpr.2017.106.
[52] S. Xu et al., "PP-YOLOE: An evolved version of YOLO," Dec. 12, 2022, arXiv: arXiv:2203.16250. doi: 10.48550/arXiv.2203.16250.
[53] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," Proc. AAAI Conf. Artif. Intell., vol. 34, no. 07, pp. 12993–13000, Apr. 2020, doi: 10.1609/aaai.v34i07.6999.
[54] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," Dec. 25, 2016, arXiv: arXiv:1612.08242. doi: 10.48550/arXiv.1612.08242.
[55] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," Feb. 18, 2025, arXiv: arXiv:2502.12524. doi: 10.48550/arXiv.2502.12524.
[56] "CUDA CPU Compute Capability," NVIDIA CORPORATION. [Online]. Available: https://developer.nvidia.com/cuda-gpus
[57] NVIDIA Corporation, "NVIDIA A100 Tensor Core GPU Datasheet," 2020. Accessed: July 15, 2025. [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-us-nvidia-1758950-r4-web.pdf
[58] M. Yaseen, "What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," Aug. 28, 2024, arXiv: arXiv:2408.15857. doi: 10.48550/arXiv.2408.1585

