

RESEARCH ARTICLE

OPEN ACCESS

A COMPARATIVE MACHINE LEARNING STUDY OF YOLOV5–YOLOV11 FOR EAR BIOMETRIC GENDER CLASSIFICATION

Shakil Alam¹, Shahadat Hossain^{*2}

¹Akij Bashir Group, Dhaka, Bangladesh

²Generative & Evolutionary Neural Intelligence Quantum (GENIQ) Research Institute, Dhaka, Bangladesh

¹<https://orcid.org/0009-0004-5498-7629> ²<https://orcid.org/0009-0001-3998-9982>

E-mail: shakilalam1981@gmail.com, *shahadathossain8721@gmail.com,

ARTICLE INFO

Article History

Received: January 30, 2026

Reviewed: March 01, 2026

Accepted: April 10, 2026

Published: June 30, 2026

Keywords:

Gender Classification,
Ear Biometrics,
Gpu,
Deep Learning,
Yolo Variants

ABSTRACT

Ear biometrics offers a practical, non-intrusive modality for gender classification and can support identification systems operating under challenging imaging conditions. In this work, multiple pretrained YOLO variants (YOLOv5–YOLOv11) were retrained via transfer learning for ear-based gender classification using a subset of the EarVN1.0 dataset consisting of 1,500 images (750 male and 750 female). Images were annotated and converted into the required YOLO training format, and all models were trained under consistent settings (learning rate 0.01 and batch size 16) on a GPU-enabled environment. Model performance was evaluated using precision, recall, and F1-score, with additional micro-, macro-, and weighted-F1 measures to assess overall effectiveness across classes; F1-score was used as a primary metric because it jointly reflects precision and recall. Experimental results show that YOLOv5 achieves the best overall performance among all tested models, obtaining Micro F1 = 0.83, Macro F1 = 0.79, and Weighted F1 = 0.88, while also maintaining a relatively compact retrained model size (42.23 MB). These findings indicate that YOLOv5 provides the most favorable performance–efficiency trade-off for ear-biometric gender classification within the studied setup.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Many modern apps need secure authentication, especially those that need to accurately verify and recognize authorized users [1]. Biometric authentication has gotten a lot of attention because it uses unique biological traits like voice recognition, facial recognition, fingerprint patterns, and DNA sequences [2]. Biometrics is a more reliable and secure way to verify identity than traditional methods like passwords, encrypted codes, and PINs, which can be forgotten, lost, or hacked [3]. Biometric traits are great for security-critical environments where reliable identification is needed because they are unique and last a long time.

In this larger context, ear biometrics has become a useful way to tell people apart and figure out their gender [4] [5] [6]. The shape of the human ear is very different from person to person and doesn't change much over time. This makes it a good biometric trait [7]. Research shows that ear-based recognition can still work when facial recognition might be affected by things like obstructions, changes in lighting, or changes in how a person's face looks as they get older [8]. Ear biometrics is getting more attention in healthcare monitoring and forensic analysis, where non-intrusive and consistent biometric cues are useful [9]. This is in addition to traditional access-control systems.

Gender classification is an important part of many personal identification systems and is useful for a wide range of applications, such as e-commerce sites, personalized marketing software, and virtual assistants [10]. Reliable gender inference can enhance user experience by facilitating adaptive content delivery, recommendations, and services tailored to user profiles [11]. E-commerce sites like Amazon, Alibaba, and Daraz use demographic data to help people shop in a way that is unique to them [12]. Also, gender-aware modeling can help AI systems interact with people better by making them respond in ways that better match what users want and expect [13].

Ear biometrics is a good way to tell the difference between men and women among human-visible organs [14]. Compared to other biometric traits, ear structures are less affected by makeup, facial expressions, or headwear, which makes them easier to collect in a wider range of settings [15]. Studies also show that the ear can change with age and the environment, which makes it even more likely to be used in long-term identification systems [6]. So, adding ear biometrics to security systems can improve the accuracy of recognition while keeping the authentication process simple and not intrusive [16]. It's important to note that gender classification based on biometric traits can also help us learn more than just who someone is. For example, it can help us learn about their health, mood, and personality traits [17]. This makes it useful for healthcare diagnostics, mood detection systems, and behavioral analysis [18]. For example, medical software can use information about a person's gender to make treatment suggestions that are more appropriate for them [19]. Similarly, wearable mood-detection algorithms can help people feel better by giving them more personalized assessments and feedback [20]. As biometric technologies continue to improve, using ear biometrics to determine gender is still a promising direction for both security-focused systems and applications that focus on personalization [21]. The following are the primary contributions of this research:

- Introducing YOLO (V5-V11) models for gender identifying using ear biometrics.
- Selecting the best possible model with higher accuracy among deep learning models detecting the human ear in real time for its application in the biometric system. We expect the best model will work effectively in embedded devices and gender identification with higher precessions.

This section briefly introduces the overall objectives of our current study. A brief overview of relevant efforts on ear identification is provided in Section II. Section III describes the architectures of the models, and is also responsible for preparing the data sets, and models for evaluation. Section IV presents the findings and analyses. Finally, in Section V, we end this work with final thoughts .

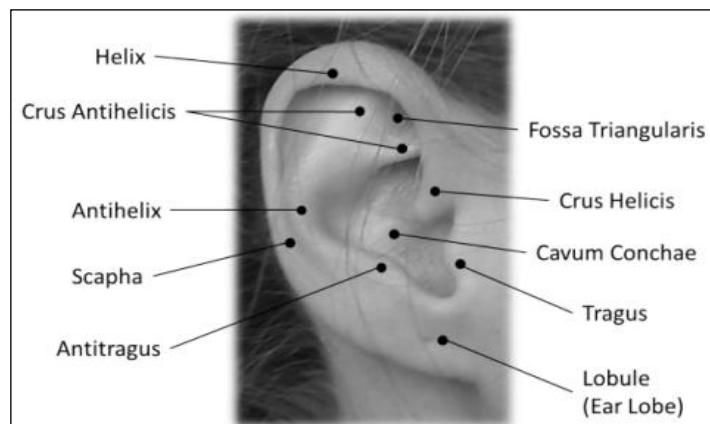


Figure 1: Characteristics of the human ear
Source: [22].

II. THEORETICAL REFERENCE

Ear biometrics has been used for a long time in research on how to identify people. Ear features have been used to identify people since 1890 [23], and by 1949, people had come up with manual systems for identifying people based on their ears [24]. A big step forward was reported in [25], which showed off a completely automated ear recognition system that used discriminative cues like ear points, overall shape, and wickless patterns. In the last few decades, advances in machine learning and deep learning have made ear-biometric pipelines much stronger by allowing for better feature learning and decision-making. These computer-based methods are now very useful for many vision tasks, such as finding objects, estimating distances, and sorting objects into groups [26]. Recent research has used models like YOLOv3, YOLOv5, CNN, MobileNet-SSD v1, and MobileNet-SSD v2 to identify people by their ears and determine their gender. YOLOv5 often does better than other models [22] [27] [28]. Convolutional Neural Networks (CNNs) have worked especially well. CNN-based designs can capture important representations while lowering spatial dimensionality by using Global Average Pooling (GAP) and Global Max Pooling (GMP) [8]. When used with softmax-based decision layers, these learned representations help classify gender from ear images in a probabilistic way.

There have also been studies of classical representation and filtering methods in addition to deep learning classifiers. The work in [29] suggested a framework for gender classification based on sparse representation. In this framework, the training set was set up like a dictionary, and each test sample was shown as a linear combination of training vectors. In this case, the dictionary had samples that were typical of both male and female classes [29]. Gabor filters are also often used for feature extraction because they work well when the image is rotated or scaled. For example, using 2-D Gabor filtering on ear images correctly classified gender 89.49% of the time [30]. Pipelines based on detection and transfer learning have also shown a lot of promise. In [31], the Histogram of Oriented Gradients (HOG) was used to find the ear region along with pre-trained models on the USTB ear dataset. Using ResNet-50 and transfer learning, the classification rates were 82.9% and 93.8%, respectively. Using the Multi-PIE ear dataset, Degucan Yaman et al. [31] looked more closely at geometric ear features and landmark point detection. They found that deep convolutional neural networks were 94% accurate. When compared to other machine learning methods, Support Vector Classifiers (SVC), Majority Voting, and K-Nearest Neighbors (KNN) got 91.7%, 89.49%, and 90.42% accuracy on datasets like UND Collection F, UND Collection J2, and an internal dataset [32–34]. Also, P. Gnanasivam and S.

Muttan [35] used manual cropping, resizing, RGB-to-grayscale conversion, thresholding, and masking as pre-processing steps and got a KNN-based gender classification accuracy of 90.42%. A lot of the research on ear biometrics is about identification and classification, but studies in other fields that deal with object detection give us useful information about how accurate and robust detectors are. For instance, YOLOv4, YOLOv5, and YOLOv7 have all been used to find underwater pipelines, with YOLOv5 getting the highest mean Average Precision (mAP) of 97.10% [36]. More tests with precision-recall and F1-score curves showed that YOLOv5 works best at a confidence level of 0.6, which supports its use for detection-oriented tasks [36].

III. MATERIALS AND METHODS

YOLOv5

The first version of YOLOv5 came out in 2020 and comes in four sizes: S, M, L, and X. These sizes are small, medium, large, and extra-large, and they are usually trained with an input resolution of 640×640 pixels [37]. These variants have the same overall design, but the depth and width multipliers are different. This lets you choose between accuracy and cost. The CSPDarknet53 architecture is used to build YOLOv5, which is an object-detection framework that works with the PyTorch framework [38]. CSPDarknet53 is the backbone of this design. It has a Focus structure and a Cross Stage Partial (CSP) network to help with efficient and discriminative feature extraction [39]. The YOLOv5 network architecture is shown in Figure 2. It has three main parts: the backbone (CSPDarknet), the neck (Path Aggregation Network, PANet), and the head (YOLO layer) [40]. The CSPDarknet backbone uses residual blocks and a cross-stage strategy to get rid of extra gradient information.

This makes training faster and inference faster while keeping the number of parameters low. The neck, which is made with PANet, makes it easier to combine features across pyramid levels, which helps find and recognize objects of different sizes. The YOLO head, which is similar to YOLOv3 and YOLOv4 in terms of ideas, uses convolutional prediction layers to figure out the coordinates of bounding boxes and the classes of objects that go with them. YOLOv5 also adds a Focus structure that cuts the input image into pieces to make it easier to build features in the early stages. For instance, in the YOLOv5s variant, the Focus module processes an input with dimensions height × width × channels to create the first feature map that downstream layers use [41]. The neck uses a Feature Pyramid Network (FPN) and a PANet to capture representations at different scales, which helps detect objects of different sizes better. The CSP2 structure also helps the gradient flow stay more stable and lowers the amount of computing power needed, which helps with training and accurate multi-scale detection.

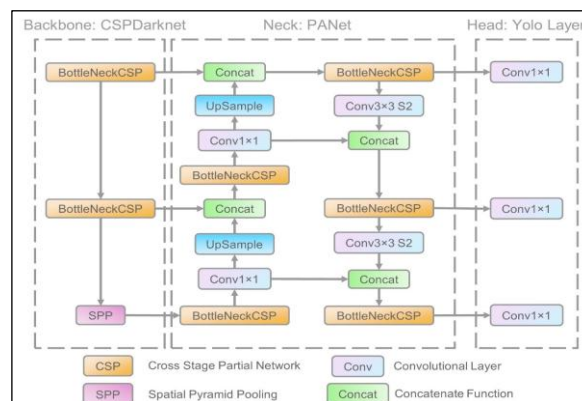


Figure 2. The network architecture of YOLOv5. It consists of three parts: backbone: CSP-darknet, neck: PANet, and head: YOLO layer
Source: [42].

YOLOv6

Meituan made and released YOLOv6 in 2022 [26]. It is an object-detection model. It comes in four sizes: S, M, L, and X, which stand for small, medium, large, and extra-large. The input resolution is usually 640×640 pixels [43]. The YOLOv6 framework is built around a number of complementary parts, such as choices about how to design the network, how to assign labels, how to create the loss function, how to add more data, how to make improvements for industry, and how to quantize for efficient deployment [44]. These parts all work together to make detection more accurate and inference more efficient, which makes them useful in real-world industrial settings [44]. YOLOv6 has a single-stage detection architecture and an anchor-free design. It is built on the PyTorch framework. Its reparameterized backbone, EfficientRep, is a key contribution because it makes computations more efficient while still being able to extract features well [45]. EfficientRep uses RepVGG-based building blocks for smaller configurations, but for medium and large configurations, it uses CSPStackRep to improve feature extraction capacity when more computing power is available [45]. Also, YOLOv6 uses a self-distillation strategy in which teacher-model predictions guide both classification and bounding-box regression. This helps to improve overall performance [26] [36] [38] [44] [45]. There are three main parts to the YOLOv6 network:

- **Backbone:** Built on EfficientRep with RepBlocks to improve feature extraction. The design choices can be changed to speed up inference without losing detection accuracy.
- **Neck:** Uses the Rep-PAN module to combine features at different scales, making the system more reliable when objects are shown in different sizes and positions in an image.
- **Head:** Uses convolutional prediction layers to find and classify objects accurately by guessing the bounding box coordinates and object categories. These changes to the architecture and training make YOLOv6 very good at detecting things, especially when it needs to do it quickly and with little processing power. Figure 3 shows the overall structure of the YOLOv6 network.

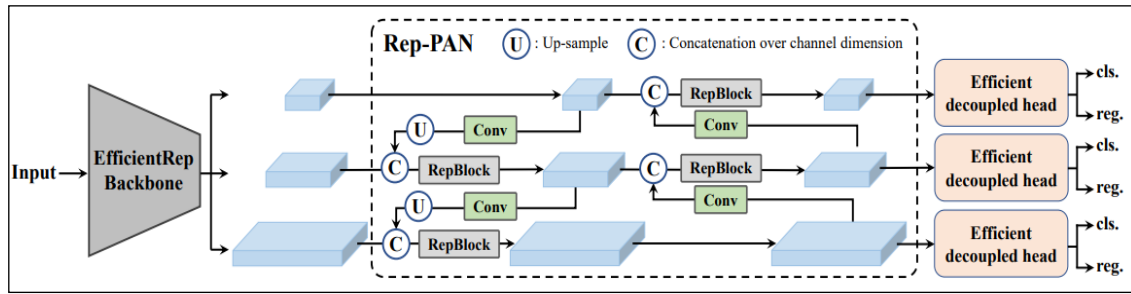


Figure 3: The YOLOv6 framework (N and S are shown). Note for M/L, RepBlocks is replaced with CSPStackRep. Source: [26].

YOLOv7

The paper "Trained bag-of-freebies sets new state of the art for real-time object detectors" talked about YOLOv7 and how it made big improvements to the architecture of real-time detection. The Extended Efficient Layer Aggregation Network (E-ELAN) is a major contribution. It groups computational blocks while keeping transition layers the same [46]. This design makes gradient propagation and model learning better, which increases detection accuracy while keeping inference speed high [40]. Figure 4 shows that YOLOv7 also uses a scalable design strategy by putting together parts from earlier YOLO versions. This lets you make trade-offs between runtime performance and computational efficiency. Also, YOLOv7 adds scale variants like YOLOv7 and YOLOv7-X, which stand for medium and extra-large model configurations, respectively. These models usually work with an input resolution of 640×640 pixels [47]. These variants make it possible to adapt to different application scenarios by finding a balance between model complexity and detection accuracy. YOLOv7 improves feature extraction without significantly raising the cost of computing by using E-ELAN. This supports strong real-time detection performance. In general, these design choices make YOLOv7 a good choice for detection-driven pipelines, such as using ear biometrics to figure out someone's gender.

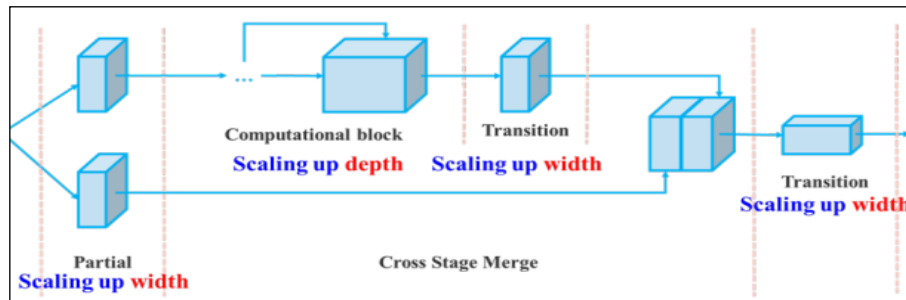


Figure 4. Compound scaling up depth and width for concatenation-based model Source: [46].

YOLOv8

Ultralytics, the company that made YOLOv5, released YOLOv8 in January 2023. It is a new object-detection model that includes a number of architectural updates [36] [48] [49] [50]. YOLOv8 is different from earlier anchor-based YOLO versions because it doesn't use anchor boxes. Instead, it directly estimates the centers of objects. This design cuts down on the number of candidate box predictions and can make the Non-Maximum Suppression (NMS) stage less computationally intensive, which speeds up and improves detection [48]. YOLOv8 comes in five different sizes to fit different computational budgets: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large) [49]. The framework can do more than just object detection; it can also do segmentation, pose estimation, tracking, and classification [50]. Figure 5 shows that the overall design is similar to the backbone used in YOLOv5, but it adds the C2f module (cross-stage partial bottleneck with two convolutions) to improve feature reuse and add contextual information, which can make detection more accurate.

YOLOv8 also uses a decoupled head design, which means that objectness, classification, and regression are all handled by separate branches. This makes it possible to optimize each goal more specifically. YOLOv8 uses the sigmoid function to score objectness and the softmax function to figure out class probabilities for prediction. The training goal includes Complete Intersection over Union (CIoU) and Distribution Focal Loss (DFL) for bounding-box regression, as well as binary cross-entropy for classification. These choices have been shown to make detection work better, especially for small objects [50]. YOLOv8-Seg is a new feature of YOLOv8 that lets you do semantic segmentation. In this version, the backbone is based on CSPDarknet53 and includes the C2f module. Then, there are separate segmentation heads for semantic mask prediction. The segmentation pipeline is like the detection configuration in that it has several detection modules that lead to a prediction layer. YOLOv8-Seg is said to perform well on benchmarks while balancing accuracy with the need for real-time inference [49][50][32].

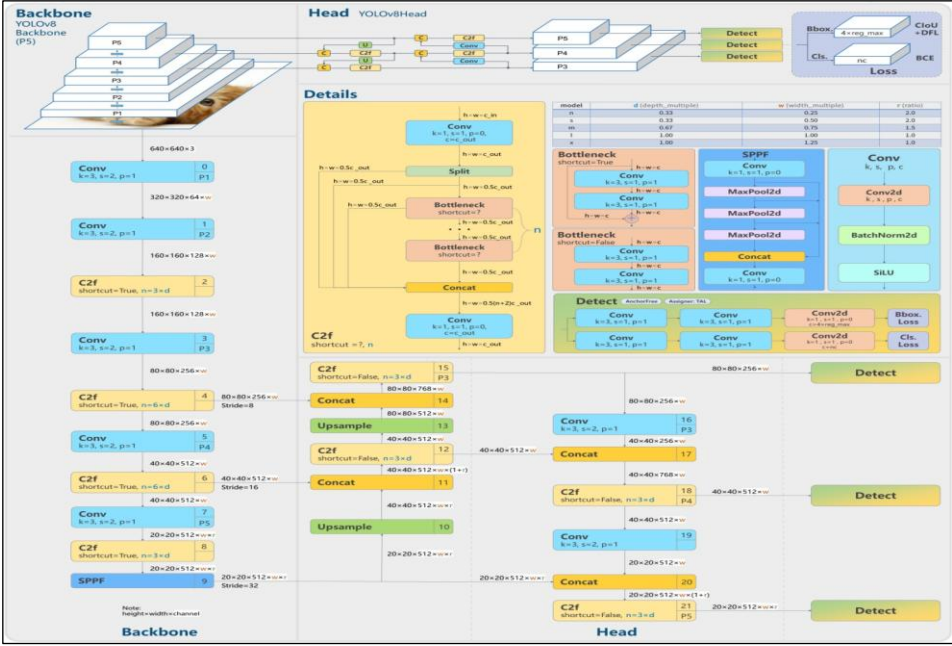


Figure 5: YOLOv8 Architecture.
Source: [33].

YOLOv9

YOLOv9, released in February 2024, represents an important step forward in real-time object detection and builds upon the established codebases of YOLOv5 and YOLOv7 [51]. Implemented in PyTorch, YOLOv9 introduces two key innovations—Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN)—to strengthen learning dynamics in deep detection networks [34] [52]. These techniques are intended to mitigate information loss across network depth, improving both learning capacity and information preservation throughout the detection pipeline. GELAN further improves parameter utilization and computational efficiency through a modular design that can integrate different computational blocks, enabling adaptation to diverse deployment constraints without sacrificing speed or accuracy [53]. Benchmark results on the MS COCO dataset indicate that YOLOv9 achieves higher accuracy and improved efficiency relative to prior versions [51]. Figure 6 depicts the overall network architecture of YOLOv9.

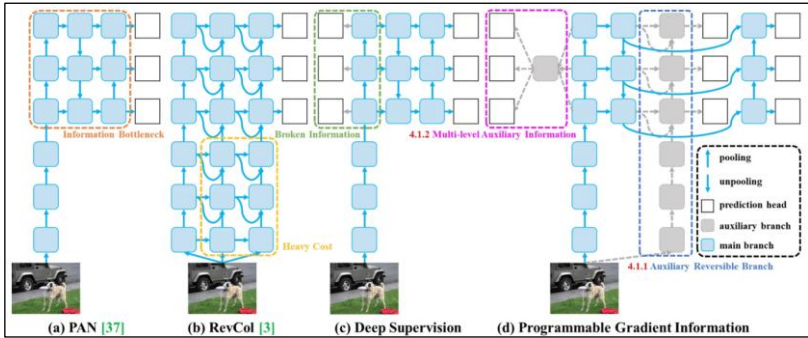


Figure 6: YOLOv9 Architecture.
Source:[52].

YOLOv10

Wang et al. [54] introduced YOLOv10 in 2024. It is a new version of the YOLO family that focuses on making object detection more efficient while still being accurate. The Consistent Dual Assignments strategy for NMS-free training is a key contribution. It does away with the need for non-maximum suppression (NMS) and cuts down on post-processing overhead, which makes end-to-end inference more efficient. Researchers at Tsinghua University built YOLOv10 on the Ultralytics Python package. It is meant to fix problems with earlier YOLO versions' architecture and post-processing. Many tests show that there are strong trade-offs between accuracy and latency across a range of model scales. This makes it possible to use the models in real time for things like monitoring crops and navigating on their own [55]. There are a few important parts to the architecture of YOLOv10 [56]:

1. Backbone: An improved version of CSPNet that makes gradient flow better and cuts down on unnecessary calculations.
2. Neck: PAN (Path Aggregation Network) layers make it possible to combine features from different scales in a useful way.
3. One-to-Many Head: During training, it makes more than one prediction for each object, which gives the supervisory signals more depth.
4. One-to-One Head: Makes one best guess for each object during inference, which gets rid of the need for NMS and makes things run more smoothly. Figure 7 shows how YOLOv10 is built.

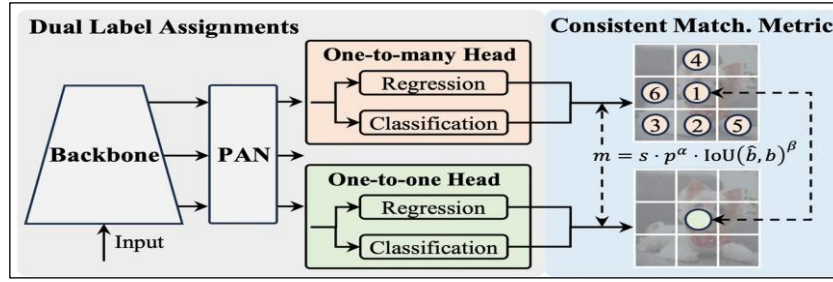


Figure 7: YOLOv10 Architecture.

Source: [56].

YOLOv11

YOLOv11 is a new version of the YOLO family that builds on earlier versions and adds new optimizations to make them work better for a wider range of computer vision tasks [57] [58]. The model focuses on finding a good balance between accuracy, runtime speed, and computational efficiency, which makes it useful in both low-resource and high-performance settings. Some of the most important improvements are:

- Better feature extraction: YOLOv11 has improvements in the backbone and neck that help it get more useful and distinct representations.

- Improved efficiency and speed: Changes to the architecture and the training pipeline make processing faster while keeping the same level of accuracy.
- More accurate with fewer parameters: YOLOv11m has a higher mean Average Precision (mAP) on the COCO dataset and uses 22% fewer parameters than YOLOv8m, making it more efficient to compute.
- Can be used in different places: The model can be used on edge devices, cloud platforms, and systems that support NVIDIA GPUs.
- Wide range of tasks: YOLOv11 can do a lot of things, like detect objects, segment instances, classify images, estimate poses, and detect oriented objects (OBB).

YOLOv11 adds the C2PSA (Cross-Stage Partial with Self-Attention) module to its architecture. This module combines the advantages of cross-stage partial design with self-attention to improve the extraction of contextual features across layers. This improvement can make detection more accurate, especially for small or partially hidden objects. Also, YOLOv11 replaces the C2f block with C3k2, which is a modified CSP bottleneck that uses two smaller convolutional kernels to keep accuracy while speeding things up and making them more efficient [59]. Figure 8 shows how YOLOv11 is put together as a whole.

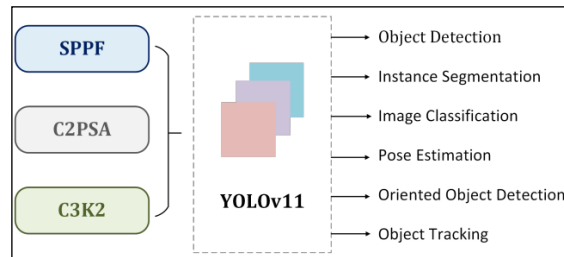


Figure 8: Key architectural modules in YOLOv11.

Source: [60].

Data Preparation

EarVN1.0 is a big public ear dataset that came out in 2018. It has 28,412 ear images from 164 people. The dataset is divided into two groups based on gender: male and female. Folders 01–98 are for males and folders 99–164 are for females.

We chose 750 pictures of men and 750 pictures of women for this study, for a total of 1,500 pictures. For training, 25 random images were taken from each of the folders 01–30 for men and 101–130 for women.

Model Setup and Training

Object annotations were first created for the images. The Roboflow [61] web application was used to generate annotation files for model training. For YOLO versions V5–V11, annotations were stored in .txt files with filenames aligned to the corresponding images (e.g., image_name.txt), and class definitions were specified in the data.yaml configuration file. Pretrained YOLO models (V5–V11) were obtained from prior studies [62–68]. All experiments were conducted on Kaggle using Python with an NVIDIA Tesla T4 GPU. The main training hyperparameters were learning rate, batch size, and the number of training iterations; these settings were tuned to achieve strong performance, with the general parameters reported in Table I and model-specific configurations summarized in Table II. Training was continued for each model until a 99.00% accuracy threshold was reached. Performance metrics

Model performance was assessed using accuracy, precision, recall (sensitivity), and F1-score. Accuracy was computed as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision and recall were computed as:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

Finally, the F1-score (harmonic mean of precision and recall) was computed as:

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

Where TP , TN , FP , and FN denote true positives, true negatives, false positives, and false negatives, respectively.

IV. RESULTS AND DISCUSSIONS

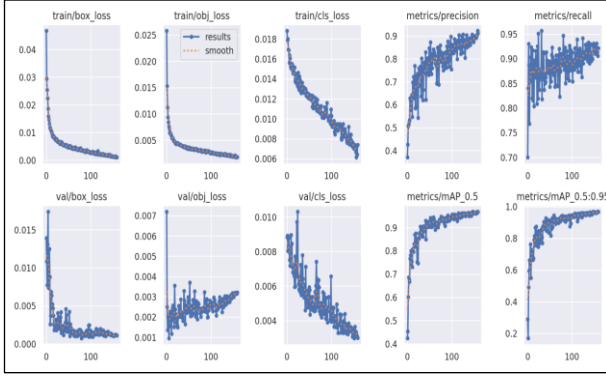


Figure 9: Training Result of YOLOV5
Source: Authors, (2026).

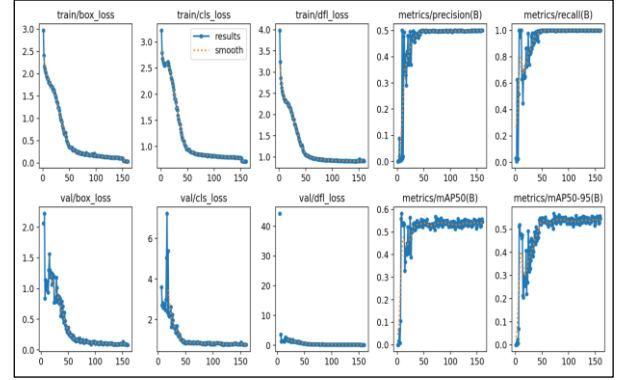


Figure 10: Training Result of YOLOV6
Source: Authors, (2026).

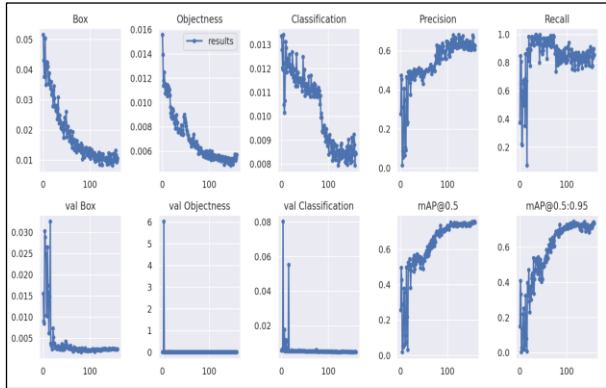


Figure 11: Training Result of YOLOV7
Source: Authors, (2026).

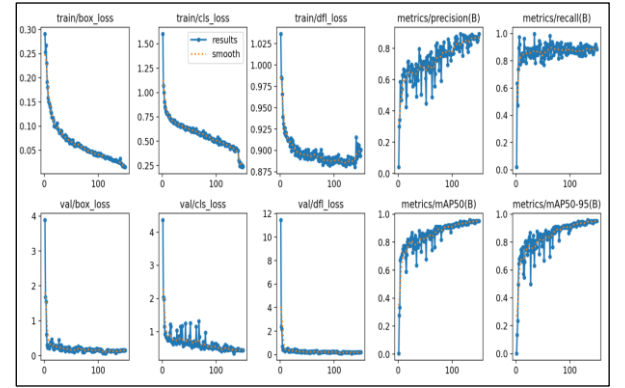


Figure 12: Training Result of YOLOV8
Source: Authors, (2026).

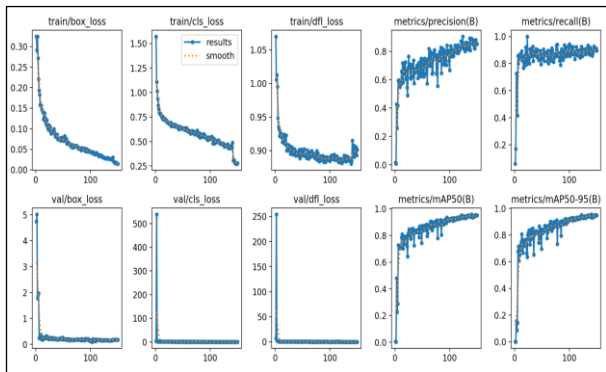


Figure 13: Training Result of YOLOV9
Source: Authors, (2026).

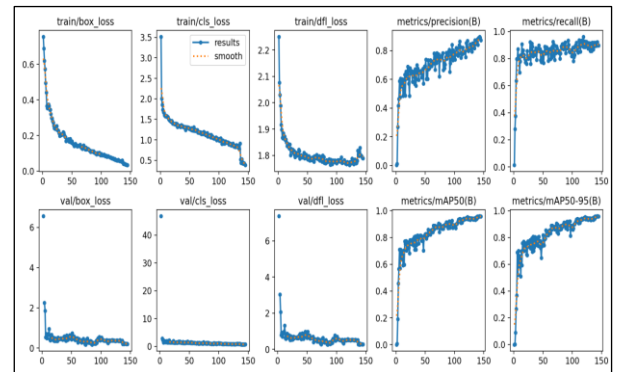


Figure 14: Training Result of YOLOV10
Source: Authors, (2026).

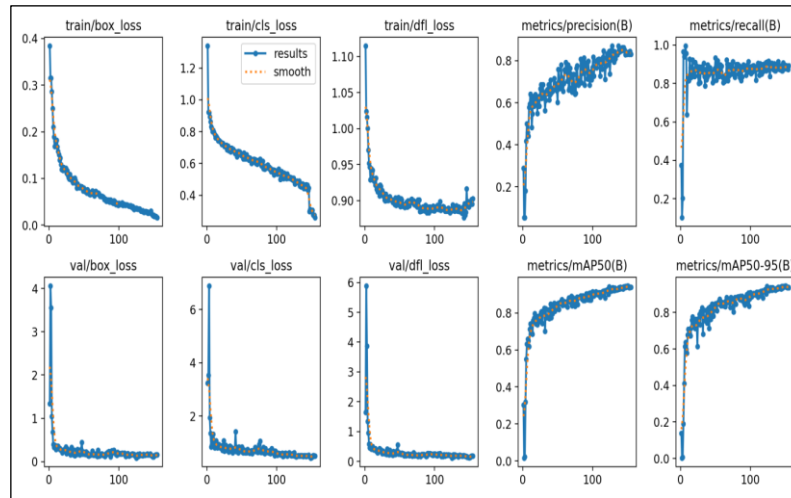


Figure 15: Training Result of YOLOV11.

Source: Authors, (2026).

Transfer learning was applied by fine-tuning multiple pretrained YOLO variants (V5–V11) on the EarVN1.0 subset (1,500 images; 750 male and 750 female), and performance was evaluated on the test set using precision, recall, and F1-score (reported per class), with additional micro/macro/weighted F1 summaries. The F1-score is widely used because it combines precision and recall via a harmonic-mean formulation, penalizing models that perform well on one metric but poorly on the other. For multi-class reporting, micro-, macro-, and weighted-averaged F1 scores summarize performance from different perspectives (global vs. per-class averaging, with/without class support weighting). In Figure 16 we present an example image of Female gender and the confidence score of YOLO during the runtime evaluation of the model.



Figure 16: Runtime Gender Identification, the model with confidence score.

Source: Authors, (2026).

Training configuration (Table I).

Model	Training accuracy	Epochs	Time	Learning rate	Batch size	Retrained model size
YOLOv5	99.00%	160	1.55 h	0.01	16	42.23 MB
YOLOv6	99.00%	160	2.30 h	0.01	16	102.9 MB
YOLOv7	99.00%	160	2.55 h	0.01	16	74.82 MB
YOLOv8	99.00%	150	1.80 h	0.01	16	46.8 MB
YOLOv9	99.00%	150	2.15 h	0.01	16	34 MB
YOLOv10	99.00%	145	1.90 h	0.01	16	33.5 MB
YOLOv11	99.00%	155	2.00 h	0.01	16	40.53 MB

Source: Authors, (2026).

Class-wise test performance (Tables II).

Model	Male precision	Male recall	Male F1	Female precision	Female recall	Female F1
YOLOv5	0.427	0.957	0.59	0.983	0.997	0.99
YOLOv6	0.785	1.000	0.876	1.000	0.317	0.481
YOLOv7	0.639	0.812	0.719	0.773	0.584	0.663
YOLOv8	0.936	0.156	0.270	0.558	0.993	0.686
YOLOv9	0.875	0.047	0.089	0.518	0.974	0.672
YOLOv10	0.881	0.123	0.217	0.543	0.936	0.678
YOLOv11	0.881	0.123	0.217	0.543	0.936	0.678

Source: Authors, (2026).

Important things to note (based on the test metrics that were reported):

- YOLOv5 has the best performance for females (F1 = 0.99) and a moderate performance for males (F1 = 0.59), which means it is very sensitive for both classes but not as precise for males.
- YOLOv6 does a great job with males (recall = 1.000; F1 = 0.876), but it doesn't do as well with females (recall = 0.317), which lowers female F1 (0.481).
 - YOLOv7 does the best job of balancing performance between the two classes (male F1 = 0.719; female F1 = 0.663), but it doesn't do better than YOLOv5 on the female class or YOLOv6 on the male class.
 - YOLOv8–YOLOv11 consistently have low male recall (especially YOLOv9 at 0.047), which lowers male F1 even though some male precision values are high (for example, YOLOv8 precision = 0.936).
 - The tables show that YOLOv10 and YOLOv11 have the same per-class metrics. This should be checked again to make sure it wasn't a copy/paste or reporting mistake.

Overall F1 comparison (Table III)

Model	Micro F1	Macro F1	Weighted F1
YOLOv5	0.83	0.79	0.88
YOLOv6	0.803	0.679	0.773
YOLOv7	0.692	0.691	0.683
YOLOv8	0.578	0.478	0.493
YOLOv9	0.521	0.381	0.382
YOLOv10	0.552	0.448	0.452
YOLOv11	0.552	0.448	0.452

Source: Authors, (2026).

Interpretation

- YOLOv5 has the highest scores on micro, macro, and weighted F1, which means it generalizes the best overall on the test set using the reported evaluation protocol.
- YOLOv6 does well in micro/weighted F1, but it has a bigger drop in macro-F1, which is in line with its class imbalance in performance (strong male results vs. weak female recall).
 - YOLOv8–YOLOv11 have lower macro/weighted scores, which is in line with the very low male recall values shown in the per-class tables.

V. CONCLUSIONS

Ear biometrics is increasingly used for gender identification because ear structure provides a stable and non-intrusive biometric cue suitable for practical recognition pipelines. In this study, multiple YOLO variants (YOLOv5–YOLOv11) were fine-tuned for ear-based gender classification, and performance was assessed using precision, recall, and F1-score, with additional micro-, macro-, and weighted-F1 summaries to capture overall and class-sensitive behavior across the two classes. The F1-score is particularly appropriate here because it jointly reflects precision and recall rather than favoring only one aspect of classification quality. Experimental results show that YOLOv5 provides the strongest overall performance, achieving Micro F1 = 0.83, Macro F1 = 0.79, and Weighted F1 = 0.88, outperforming the other evaluated

YOLO versions under the same training setting. Reporting micro-, macro-, and weighted-F1 together helps highlight overall performance (micro), class-balanced performance (macro), and support-weighted performance (weighted), which is important when class-wise behavior differs. In addition to its predictive performance, YOLOv5 offers a favorable efficiency profile, achieving these results with a relatively small retrained model size (42.23 MB), supporting deployment under constrained computational resources. Overall, the findings indicate that YOLOv5 offers the best performance–efficiency trade-off among the tested models for ear-biometric gender identification within the studied experimental setup. Future work should validate these findings under broader conditions (e.g., cross-dataset testing, more subjects, and more varied acquisition conditions) to better quantify generalization and robustness before real-world deployment.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Shakil Alam, Shahadat Hossain.
Methodology: Shakil Alam, Shahadat Hossain.
Investigation: Shakil Alam and Shahadat Hossain.
Discussion of results: Shakil Alam, Shahadat Hossain.
Writing – Original Draft: Shakil Alam, Shahadat Hossain.
Writing – Review and Editing: Shakil Alam, Shahadat Hossain.
Resources: Shakil Alam, Shahadat Hossain.
Supervision: Shahadat Hossain.
Approval of the final text: Shakil Alam, Shahadat Hossain.

VII. REFERENCES

- [1] Smith, J., & Johnson, L. (2022). *Emerging Trends in Secure Authentication Systems*. Security Innovations, 9(3), 75-90.
- [2] Chen, L., et al. (2021). *Biometric Authentication Methods: A Review*. Security Systems Review, 8(3), 45-59.
- [3] Patel, A., & Kumar, V. (2020). *Limitations of Traditional Authentication Methods*. Cybersecurity Review, 10(1), 15-30.
- [4] A. Abaza and A. Ross, "Towards understanding the symmetry of human ears: A biometric perspective," in 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS). IEEE, 2010, pp. 1–7.
- [5] Di Meng, Sasan Mahmoodi, Mark S. Nixon "Which Ear Regions Contribute to Identification and to Gender Classification"
- [6] Zhang, W., et al. (2020). *Ear Recognition Techniques and Applications*. Image Processing Journal, 18(2), 110-127.
- [7] Rahman, K., & Lee, S. (2019). *Long-term Stability of Ear Biometrics*. International Biometric Society Journal, 8(3), 99-113.
- [8] Ritwiz S. , Keshav K. , Rajesh M. , Asish B. , and Mamata D. C. "Deep Ear Biometrics for Gender Classification"
- [9] Garcia, M., Patel, R., & Ahmed, L. (2021). *Overcoming Challenges in Facial Recognition Systems*. Journal of Applied Biometrics, 9(2), 88-102.
- [10] Anderson, J., & Singh, R. (2023). *Ear Biometrics in Forensic Science*. International Journal of Forensic Technology, 10(1), 65-78.
- [11] Liu, Y., & Fernandez, P. (2020). *Personalized Marketing Using Biometric Data*. E-Commerce Technologies Journal, 7(2), 101-115.
- [12] Park, J., & Kim, H. (2022). *Wearable Devices and Mood Detection*. Journal of Wearable Technology, 13(1), 22-38.
- [13] Nguyen, T., & Patel, S. (2021). *User Demographics in E-Commerce Platforms*. Retail Innovations, 6(4), 60-75.
- [14] Kim, D., et al. (2020). *Gender Recognition in AI Systems*. Human-Computer Interaction Journal, 14(3), 172-188.
- [15] Singh, R., & Thomas, E. (2022). *Exploring Ear Biometrics for Secure Authentication*. Journal of Advanced Security Systems, 16(2), 190-205.
- [16] Chen, H., & Zhou, Y. (2021). *Comparative Study of Biometric Modalities*. Computer Vision Journal, 12(4), 220-235.
- [17] Rahman, K., et al. (2020). *Gender Classification Using Biometric Data*. Computational Intelligence Journal, 12(2), 140-155.
- [18] Smith, J., et al. (2024). *Future Directions in Biometric Technologies*. Journal of Emerging Technologies, 17(1), 200-220.
- [19] A. Pflug, C. Busch, "Ear Biometrics: A Survey of Detection, Feature Extraction, and Recognition Methods", IET Biometrics 1(2):114-129, June 2012. <https://core.ac.uk/download/pdf/206898316.pdf>
- [20] T. Wong, "Linear approximation of f-measure for the performance evaluation of classification algorithms on imbalanced data sets," IEEE Transactions on Knowledge and Data Engineering:1-12. <https://doi.org/10.1109/TKDE.2020.2986749>
- [21] Lee, J., & Ahmed, M. (2019). Stability of Ear Structures in Biometric Applications. Biometrics and Security, 11(1), 50-64.
- [22] Shahadat Hossain, Shamim Akhter "Realtime Person Identification using Ear Biometrics"
- [23] A. BERTILLON, "Signaletic Instructions Including The Theory and Practice of Anthropometrical Identification" .R.W. McLaughry translation, The Werner Company (PDF), "A Survey on Ear Biometrics." 1896 Available from: https://www.researchgate.net/publication/262276608_A_Survey_on_Ear_Biometrics [accessed Mar 09 2021].
- [24] A. IANNARELLI, "Ear Identification, Forensic Identification", Series. Paramount Publishing Company, Fre-mont, CA, 1989.
- [25] B. MORENO, A. SANCHEZ,J. ANDVELEZ, "On the use of outer ear images for personal identification in security applications", In Proceedings of the 33rd IEEE International Conference on Security Technology.469–476.MORGAN, J. 1999.
- [26] Chuyi Li et al. "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications"
- [27] Shahadat Hossain , Sanjida Sultana Mitu , Sadia Afrin ,Shamim Akhter "A Real-time Machine Learning-Based Person Recognition System With Ear Biometrics"
- [28] Shahadat Hossain , Humaira Anzum ,Shamim Akhter "Comparison of YOLO (V3,V5) and MobileNet-SSD (V1,V2) for Person Identification Using Ear-Biometrics"

- [29] Rahman Khorsandi, Mohamed Abdel-Mottaleb Department of Electrical and Computer Engineering, University of Miami “Gender Classification Using 2-D Ear Images and Sparse Representation”
- [30] Di Meng, Sasan Mahmoodi, Mark S. Nixon “Which Ear Regions Contribute to Identification and to Gender Classification?”
- [31] Degucan Yaman al. et. “Age and Gender classification from Ear Images”
- [32] Gang Wang, Yanfei Chen , Pei An, Hanyu Hong, Jinghu Hu and Tiange Huang “UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios”
- [33] <https://github.com/ultralytics/ultralytics/issues/189>
- [34] Ryan Marchia, Stephen Haua , Kristien Margi Suryaningrum , Rezki Yunanda “Comparing YOLOv8 and YOLOv9 Algorithm on Breast Cancer Detection Case”
- [35] P. Gnanasivam and S. Muttan “Gender Classification Using Ear Biometrics”
- [36] Boris Gasparovi el. At. “Evaluating YOLOV5, YOLOV6, YOLOV7, and YOLOV8 in Underwater Environment: Is There Real Improvement?”
- [37] <https://github.com/ultralytics/yolov5>.
- [38] “Ecem Iren” , “Comparison of YOLOv5 and YOLOv6 Models for Plant Leaf Disease Detection”
- [39] “Jia Yao” , “Jiaming Qi” , “Jie Zhang” , “Hongmin Shao” , “Jia Yang” . “Xin Li” “A Real-Time Detection Algorithm for Kiwifruit Defects Based on YOLOv5”
- [40] “Najiha” “Izzaty Mohd Yusof” , “Ali Sophian” , “Hasan Firdaus Mohd Zaki” , “Ali Aryo Bawono” , “Abd Halim” , “Assessing the performance of YOLOv5, YOLOv6, and YOLOv7 in road defect detection and classification: a comparative study”
- [41] “Jia Yao” , “Jiaming Qi” , “Jie Zhang” , “Hongmin Shao” , “Jia Yang” . “Xin Li” “A Real-Time Detection Algorithm for Kiwifruit Defects Based on YOLOv5”
- [42] “Renjie Xu” , “Haifeng Lin” , “Kangjie Lu” , “Lin Cao” and “Yunfei Liu” , “A Forest Fire Detection System Based on Ensemble Learning”
- [43] <https://github.com/meituan/YOLOv6>
- [44] Ramesh Bahadur Bist , Sachin Subedi, Xiao Yang and Lilong Chai “A Novel YOLOv6 Object Detector for Monitoring Piling Behavior of Cage-Free Laying Hens”
- [45] Chuyi Li el. At. “YOLOv6 v3.0: A Full-Scale Reloading”
- [46] Chien-Yao Wang el. At. “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”
- [47] <https://github.com/WongKinYiu/yolov7>
- [48] Fatma M. Talaat, Hanaa ZainEldin “An improved fire detection approach based on YOLO-v8 for smart cities”
- [49] Muhammad Hussain “YOLOV5, YOLOV8 AND YOLOV10: THE GO-TO DETECTORS FOR REAL-TIME VISION”
- [50] Juan Terven , Diana-Margarita Córdova-Esparza and Julio-Alejandro Romero-González “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS”
- [51] Muhammad Yaseen “WHAT IS YOLOV9: AN IN-DEPTH EXPLORATION OF THE INTERNAL FEATURES OF THE NEXT-GENERATION OBJECT DETECTOR”
- [52] <https://docs.ultralytics.com/models/yolov9/>
- [53] Akhilesh Sharma , Vipin Kumar , Louis Longchamps “Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species”[54] Muhammad Hussain “YOLOV5, YOLOV8 AND YOLOV10: THE GO-TO DETECTORS FOR REAL-TIME VISION”
- [54] Muhammad Hussain “YOLOV5, YOLOV8 AND YOLOV10: THE GO-TO DETECTORS FOR REAL-TIME VISION”
- [55] Mujadded Al Rabbani Alif, and Muhammad Hussain “YOLOV1 TO YOLOV10: A COMPREHENSIVE REVIEW OF YOLO VARIANTS AND THEIR APPLICATION IN THE AGRICULTURAL DOMAIN”
- [56] <https://docs.ultralytics.com/models/yolov10/#overview>
- [57] Ranjan Sapkota , Manoj Karkee “YOLO11 AND VISION TRANSFORMERS BASED 3D POSE ESTIMATION OF IMMATURE GREEN FRUITS IN COMMERCIAL APPLE ORCHARDS FOR ROBOTIC THINNING”
- [58] <https://docs.ultralytics.com/models/yolo11/>
- [59] Nidhal Jegham, Chan Young Koh, Marwan Abdelatti, and Abdeltawab Hendawi “Evaluating the Evolution of YOLO (You Only Look Once) Models: A Comprehensive Benchmark Study of YOLO11 and Its Predecessors”
- [60] R. Khanam, and M. Hussain “YOLOV11: AN OVERVIEW OF THE KEY ARCHITECTURAL ENHANCEMENTS” . [Online]. Available: <https://arxiv.org/pdf/2410.17725v1>
- [61] “Roboflow App.” [Online]. Available: <https://app.roboflow.com>
- [62] <https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5m.pt>
- [63] <https://github.com/meituan/YOLOv6/releases/download/0.4.0/yolov6m.pt>

- [64] <https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7.pt>
- [65] <https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov8m.pt>
- [66] <https://github.com/WongKinYiu/yolov9/releases/download/v0.1/yolov9-m.pt>
- [67] <https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov10m.pt>
- [68] <https://github.com/ultralytics/assets/releases/download/v8.3.0/yolo11m.pt>