International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-13, Issue-4, July 2025
DOI: https:/doi.org/10.55524/ijircst.2025.13.4.2
Article ID IRP-1653, Pages 15-32
www.ijircst.org

An Improved YOLOv3 (E-YOLOv3) to Detect Objects and Comparative Analysis of YOLOv3, ResNet101-YOLOv3, YOLOv8 and DETR

Jashanpreet Singh[1] and Dr. Rajiv Kumar[2]

[1]Research Scholar, Department of Computer Applications, RIMT University, Mandi Gobindgarh, Punjab, India
[2] Dean and Professor, School of Computing, Department of Computer Applications RIMT University, Mandi Gobindgarh, Punjab, India

Correspondence should be addressed to Jashanpreet Singh; jashanpreet60@gmail.com

Received 22 May 2025; Revised 6 June 2025; Accepted 20 June 2025

ABSTRACT- The object detection field has received significant improvement through deep learning technology and YOLO (You Only Look Once) stands out as a leading model which delivers fast and precise real-time results. This research evaluates the performance of five object detection models including YOLOv3 and ResNet101-based YOLOv3 (R-YOLO) and EfficientNetB0-based YOLOv3 (E-YOLO, proposed model) and YOLOv8 and DETR which were trained on the COCO dataset. The evaluation process used a test dataset consisting of 19,960 images to measure Precision, Recall, F1 Score and mean Average Precision (mAP). To assess robustness, all models were further tested on challenging subsets, including 10 images each of blurred, low-light, and clean images. Rigorous testing against COCO benchmark datasets revealed that the modified E-YOLOv3 outperforms the state-of-the-art detection models, especially in environments like blurred scene, clean scene and lowlight scene. Our model achieved a mean Average Precision (mAP) of 96.85%. The proposed E-YOLO model outperformed YOLOv3, R-YOLO, and YOLOv8 in both general and adverse conditions. E-YOLO achieved competitive accuracy compared to DETR while using significantly less computational resources and faster inference which makes it more suitable for real-time applications. DETR achieved better mAP and precision results than E-YOLO in complex and overlapping scenes because of its transformer-based global attention but its high resource requirements and slow inference speed limit its performance. E-YOLO provides an excellent balance between accuracy and efficiency by delivering strong performance across various scenarios at a low computational cost. The solution provides practical and effective real-world object detection capabilities especially when hardware constraints exist.

KEYWORDS- YOLO, Object Detection, Localization, Deep Learning.

I. INTRODUCTION

Object detection is a fundamental component of computer vision, essential for bridging the gap between images and textual information, as well as for tracking individual objects within visual data. Its ability to extract meaningful insights makes it indispensable across a wide range of fields, including machine vision, deep sea visual monitoring systems [1], [2], [3], [4], [5], [6], and the detection of anomalies in medical imaging. In recent years, the rapid evolution of deep learning has significantly accelerated progress in object detection algorithms [7], [8].

The technology of Artificial Intelligence (AI) has established itself in renewable energy [9], security, healthcare [9] and education sectors. The manufacturing industry demonstrates exceptional compatibility with Computer Vision (CV) automation. Quality Inspection (QI) functions as a critical element in manufacturing operations because it ensures both product reliability and customer satisfaction [9]. The industry offers significant automation potential but surface inspection remains difficult because defects appear in multiple complex forms [12]. The process of manual inspection proves both time-consuming and vulnerable to human errors because of worker fatigue and high operational expenses and production delays [13]. The current limitations of quality inspection tasks demonstrate the potential of CV-powered solutions to automate these processes. These solutions integrate perfectly with current surface defect detection systems to boost operational efficiency and solve traditional inspection method limitations [14]. The deployment of such systems depends on CV architectures that fulfil particular operational requirements which vary between manufacturing sub-domains [15]. The identification of multiple defects along with their exact spatial positions stands as a critical requirement for many applications. The localization requirements of object detection make it more suitable than image classification because the latter only indicates object presence without providing location information. The object detection field contains two main detector types which include single-stage and two-stage detectors [17]. Two-stage detectors operate sequentially through region proposal generation followed by classification and localization steps [18]. The high accuracy of these detectors comes at the cost of high computational complexity which limits their deployment in real-time applications on edge devices with limited resources. Single-stage detectors unite classification and regression into one processing step which reduces computational requirements and makes them suitable for production environments [19].

The Single Shot Detector (SSD) [20] and Deconvolutional Single Shot Detector (D-SSD) [21] and RetinaNet [22] are among the several single-stage object detectors that have been developed but the YOLO (You Only Look Once) family of architectures [23] has received the most attention. This growing popularity is largely due to YOLO's alignment with industrial requirements, including high accuracy, lightweight design, and suitability for edge-device deployment. Over the past five years, YOLO variants have dominated the landscape of real-time object detection, with the latest version, YOLOv8, released in 2022.

Real-time object detection serves as a fundamental requirement for multiple fields which include autonomous driving robotics video surveillance and augmented reality applications. The YOLO approach stands out because it achieves a perfect balance between speed and accuracy which allows fast and accurate object detection in images. The YOLO framework has evolved through multiple versions since its first release to overcome previous weaknesses and boost its performance. The YOLO architecture implements a Convolutional Neural Network (CNN) to achieve real-time detection as shown in Figure 1. The DarkNet backbone processes standardized 416×416×3-pixel input images through its series of convolutional layers which extract high-level visual features. The extracted features undergo flattening before being sent to fully connected layers which produce three prediction grids. The output of each grid cell includes bounding box coordinates together with object confidence scores and class probability predictions. The fast image processing capabilities of this architecture make it highly suitable for real-time applications that require tracking small objects.

The real-time object detection features of YOLO have become essential for autonomous vehicle systems because they enable fast identification and tracking of vehicles and pedestrians and bicycles and other obstacles [24] [25] [26]. YOLO has proven successful in multiple domains beyond transportation through its applications in video surveillance action recognition [27] [28] sports analytics [29] and human-computer interaction systems [6]. YOLO-based models in agriculture enable the detection and classification of crops [30] as well as pests and plant diseases [31] which leads to precision farming and automation. The architecture demonstrates success in biometric applications through its use in facial recognition and face detection and security systems [22].

YOLO has been applied to medical tasks such as cancer detection [32], skin segmentation [33], and pill identification [34] in the medical domain, leading to improvements in diagnostic accuracy and treatment efficiency. Remote sensing is another area where YOLO excels supporting the detection and classification of objects in satellite and aerial imagery for land use analysis, urban planning, and environmental monitoring [35]. In the security sector, YOLO models are integrated into real-time video analytics systems for anomaly detection [36], enforcing social distancing, and identifying mask usage [37].

The YOLO-based surface inspection tools have improved manufacturing quality control by detecting defects and irregularities during production [38].

YOLO technology serves wildlife monitoring by helping identify endangered species which supports conservation and habitat management initiatives [41]. The architecture finds widespread application in robotics [15] and aerial object detection through drone technology [42].

The YOLO object detection algorithm demonstrates its functionality through Figure 2. The input image gets divided into a grid system which assigns each cell to detect objects that fall inside its area. The YOLO algorithm generates multiple bounding boxes together with their corresponding confidence scores for each grid cell to determine both object locations and their presence. The system predicts object class probabilities through its output. The YOLO system implements confidence thresholding followed by non-maximum suppression to enhance predictions and remove duplicate detections which produces precise non-overlapping bounding boxes as seen in the dog and bicycle examples in figure 2. The YOLO framework operates in real time to achieve its exceptional speed and performance.

The current object detection methods have shown progress yet they still face problems with accuracy and efficiency. The limitations of object detection can be overcome by machine learning and deep neural network methodologies which demonstrate superior capabilities. The research presents a new YOLO architecture adaptation [23] to solve the existing problems. The modified YOLOv3 model achieves better performance through multiple strategic enhancements. The main research contributions consist of:

- Using Efficient Net B0 as the backbone network
- Feature pyramid network
- Substituting the loss function with EIoU
- Advance data augmentation

A. Paper Organization

The paper consists of five distinct sections. Section II provides an extensive evaluation of existing research which directly pertains to our investigation. Section III outlines the methods we have established to address the research problem. Section IV outlines the experimental procedures we conducted and presents the obtained results which demonstrate the effectiveness of our solution. Section V presents the main research outcomes and establishes final conclusions from the study. The paper outlines future research possibilities while emphasizing the significance and potential effects of our work on field development.

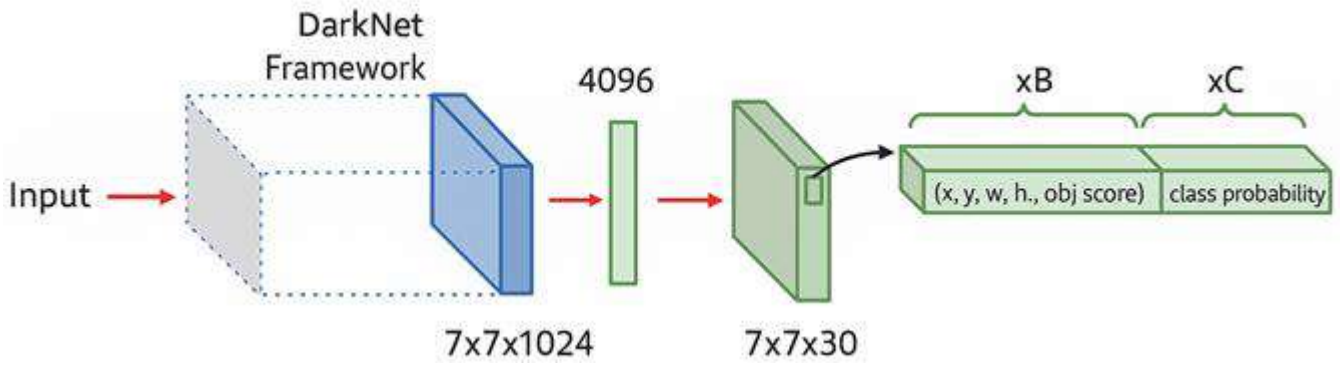

Figure 1: Architecture of YOLO

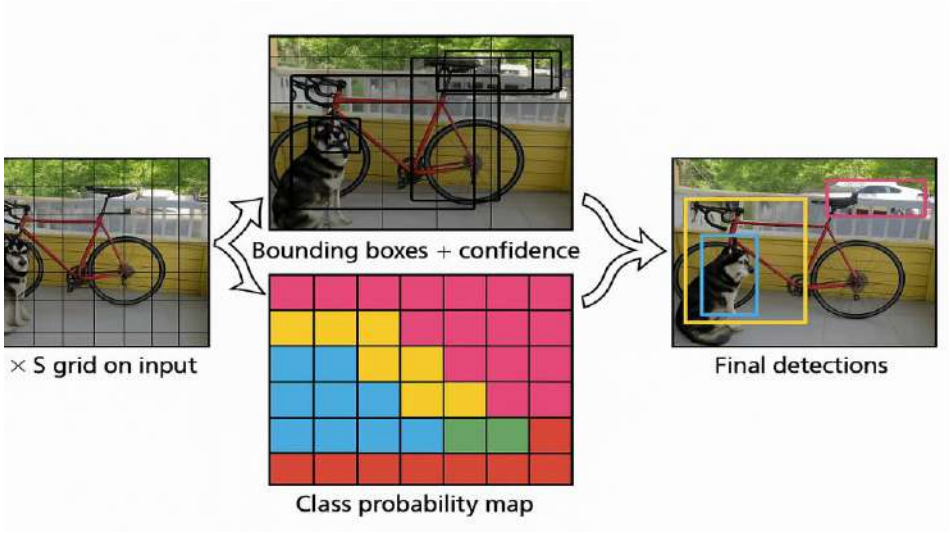

Figure 2: Object detection stages of YOLO

II. RELATED WORKS

The detection of objects has experienced major advancements through deep learning because this technology extracts important features from unprocessed data [43], [44]. The capability has led to better detection accuracy and performance [45], [46], [47]. The deep learning-based object detection algorithms exist in two primary categories which include two-stage and one-stage methods. The two-stage detection methods consist of a region proposal stage followed by classification. The basic CNN [48] and its extensions RCNN [49] and Fast R-CNN [50] and Faster R-CNN [51] and Mask R-CNN [52] represent notable examples of this detection approach. The OFSM [53] and hybrid models that combine spectral data with CNNs [54] represent additional model improvements.

Kellenberger et al. [55] developed CNN technology for wildlife monitoring at scale which produced a 90% recall rate using reduced dataset sizes to decrease human annotation requirements. Roy et al. [54] developed Hybrid Spectral CNN (HybridSN) which combined 3D and 2D CNNs for hyperspectral image (HSI) classification to reach more than 99.6% accuracy across different datasets with small variations. The model's performance decreased when the available data amount became smaller.

Guo et al. [56] applied Recurrent Neural Networks (RNNs) to reconstruct super-resolution data for enhancing UAV-based detection and localization. The system needs additional optimization to reach higher efficiency levels. The data migration strategies proposed by researchers for specific regions of interest face limitations because they work only with restricted data categories [52], [57].

Lei et al. [58] developed a multi-module CNN system which uses semantic segmentation for bayberry fruit harvesting automation. The two-stage detection approach needs predefined region proposals which creates complexity and slows down the process.

In 2013 Redmon et al. introduced YOLO (You Only Look Once) [59] which became a one-stage detection framework that enabled real-time object detection through direct integration of detection within the classification pipeline. Redmon and Farhadi released YOLOv3 [60] in 2018 to improve detection speed and accuracy. YOLOv3 achieved real-time efficiency in fruit detection through the work of Kuznetsova et al. [61] and Li et al. [46]. The release of YOLOv4 [13] brought enhanced speed-accuracy optimization which proved superior to previous models in precision and responsiveness according to Dewi et al. [40] and Kumar et al. [36].

Despite these advancements, challenges remain. Deep learning models in medical image colorization require substantial computational power and storage capacity according to Xia et al. [62]. The Hierarchical Multi-Attention Transfer (HMAT) framework presented by Gao et al. [51] outperformed all existing knowledge distillation (KD) techniques.

Research has shown that YOLO faces certain challenges when used for vehicle counting applications. The model demonstrates poor performance when it comes to accuracy and flexible interval tracking [63]. The research aims to fill these knowledge gaps through the development of computer vision algorithms which use pre-recorded video and YOLO framework for automated traffic counting. The system uses YOLO within the TensorFlow API and OpenCV to detect objects and track them before counting them. The system achieves 90% accuracy in comparison to manual counts yet it undercounts in situations with suboptimal video quality. The technical contributions are supported by a benefit-cost analysis that demonstrates the proposed method provides significant economic benefits and investment returns.

Deep convolutional architectures have driven YOLO model evolution through advances in sophisticated backbone and detection heads which deliver better accuracy and computational balance. Despite these models' detection abilities, they struggle to identify false positives and find small objects in crowded areas. The authors of [64] developed YOLO-NL (You Only Look Once and None Left) to enhance detection precision and localization by introducing global dynamic label assignment between anchors and targets. The model merges improvements between CSP Net and PANet through a combination of self-attention mechanisms and a shortest-longest gradient strategy. Rep-CSP Net enhances inference speed through ghost convolution techniques and serial SSPP structures along with reparameterization capabilities. The YOLO-NL model achieves a COCO 2017 mean average precision (mAP) of 52.9% that represents a 2.64% improvement over YOLOX while reaching exceptional performance in real-world tasks such as face mask detection at 98.8% accuracy.

The research in [65] revealed that YOLO lacks consistent performance between video frames because confidence score drops and class transitions negatively affect tracking and counting functions. The researchers enhanced the YOLO algorithm by using the RANSAC algorithm to identify outlier confidence shifts for better temporal consistency. The system applied interpolation to create a smooth transition between frame confidences. The accuracy of object counting increased from 66% to 87% and standard dataset classification accuracy reached 94-96%.

The improvement of daily visual performance and safety and security demands night vision technology to function properly. The main focus of [66] centers on developing night vision systems to fulfil current social requirements. The study identifies night vision as a crucial research topic yet it suffers from limited availability of comprehensive datasets designed for deep learning applications. The main difficulty in detecting objects at night stems from poor illumination which hinders both object recognition and feature extraction.

The study builds a broad night vision dataset that contains multiple real-world scenarios such as strong point light sources and vehicle headlight blur and insect presence and rainy weather conditions. The study compares three object detection models by evaluating Fast R-CNN with 84% mAP at 45 FPS and Faster R-CNN with 88% mAP at 20 FPS and YOLOv4 achieving the best results with 95% mAP at 79 FPS. YOLOv4 demonstrates the best combination of accuracy and processing speed so it becomes the preferred model. The combination of low-pass and unsharp filters during preprocessing enhances image clarity which results in improved detection performance reaching a mAP of 95%. The system identifies six classes including Human, Car, Bike, Animal, Truck and Van.

Research has aimed to solve two fundamental challenges of deep learning-based real-time object detection which are the high computational requirements and the requirement of large labelled datasets. The research introduces a modified architecture which optimizes the trade-off between accuracy and speed to support efficient real-time implementation.

III. PROPOSED TECHNIQUE

The field of deep learning-based object detection has made substantial progress during recent years which improved both speed and real-world application robustness. The YOLO (You Only Look Once) series has gained prominence because its unified architecture allows real-time processing at the same level of accuracy. We introduce an improved version of YOLOv3 which focuses on enhancing object detection results.

Our approach involves a customized YOLOv3 model that incorporates specialized preprocessing strategies and architectural refinements to improve sensitivity to dynamic objects. To further boost accuracy and efficiency, we introduce three key modifications: (1) replacing the original backbone with EfficientNet-B0 for better feature representation, (2) adopting the EIoU (Enhanced Intersection over Union) loss function for more precise localization, and (3) integrating a Feature Pyramid Network (FPN) to improve multi-scale object detection. These enhancements enable the model to detect objects with greater accuracy and efficiency across varying scales and challenging conditions. The detailed architecture of the proposed detection system is illustrated in Figure 3 and described as follows:

Integration of EfficientNet as the Backbone Network**:** In this study, EfficientNet is employed as the backbone network to enhance the feature extraction capability of the original YOLOv3 architecture. EfficientNet, inspired by residual network principles, incorporates depth wise separable convolutions for computational efficiency and employs channel attention mechanisms to emphasize informative feature channels. Its architecture begins with a Stem module followed by a sequence of MBConvBlocks, and includes Conv2D layers, Batch Normalization, Swish activation functions, and pooling layers.

To maintain architectural compatibility with YOLOv3's original backbone, DarkNet-53, replaced are made to the EfficientNet structure. Specifically, only the essential early components namely the Stem and MBConvBlocks are retained, while deeper layers are removed. This adjustment ensures that the number of down sampling operations remains consistent with the original YOLOv3 design. The modified EfficientNet B0 backbone thus comprises 17 layers, with the *Stem* serving as an initial convolutional layer followed by Batch Normalization, which compresses the input image and facilitates down sampling. The improved backbone architecture is illustrated in Figure 3. Following feature extraction, the output feature maps at multiple resolutions 13×13, 26×26, and 52×52 are derived from the MBConvBlock layers. These feature maps undergo an up-sampling process, starting from the smallest scale, and are progressively fused with adjacent higher-

resolution features. This fusion process continues until all three detection scales are constructed. By integrating shallow features, rich in spatial and localization information, with deeper layers containing high-level semantic cues, the modified network enhances YOLOv3's capability to manage complex visual scenes and significantly improves object detection accuracy.

The Feature Pyramid Network (FPN) represents a widely used architecture for multi-scale feature representation which was initially developed to boost object detection performance in Faster R-CNN and other frameworks. The main goal of this architecture is to merge the semantic information from deeper layers with the spatial details from shallower layers. The fusion process resolves the natural disparity in semantic richness between network depths because deeper layers understand abstract semantic features but shallower layers maintain precise spatial information. FPN integrates these features through a top-down pathway and lateral connections. A 1×1 convolution is used to modify the number of channels in shallow feature maps so that they can be combined with features from deeper layers. The transformed shallow features are added element-wise to the up sampled higher-level feature maps. The hierarchical fusion process produces better multi-scale representations that enhance the network's ability to detect objects of different sizes and complexities.

The EIoU loss function is used instead of the original loss function in object detection because the loss function determines the difference between predicted outputs and ground truth values. The YOLOv3 model uses a composite loss function that combines three components: $Loss_{box}$ for bounding box regression, $Loss_{obj}$ for confidence, and $Loss_{cls}$ for classification. The total loss is calculated as a weighted sum of these individual losses as shown in Equation 1.

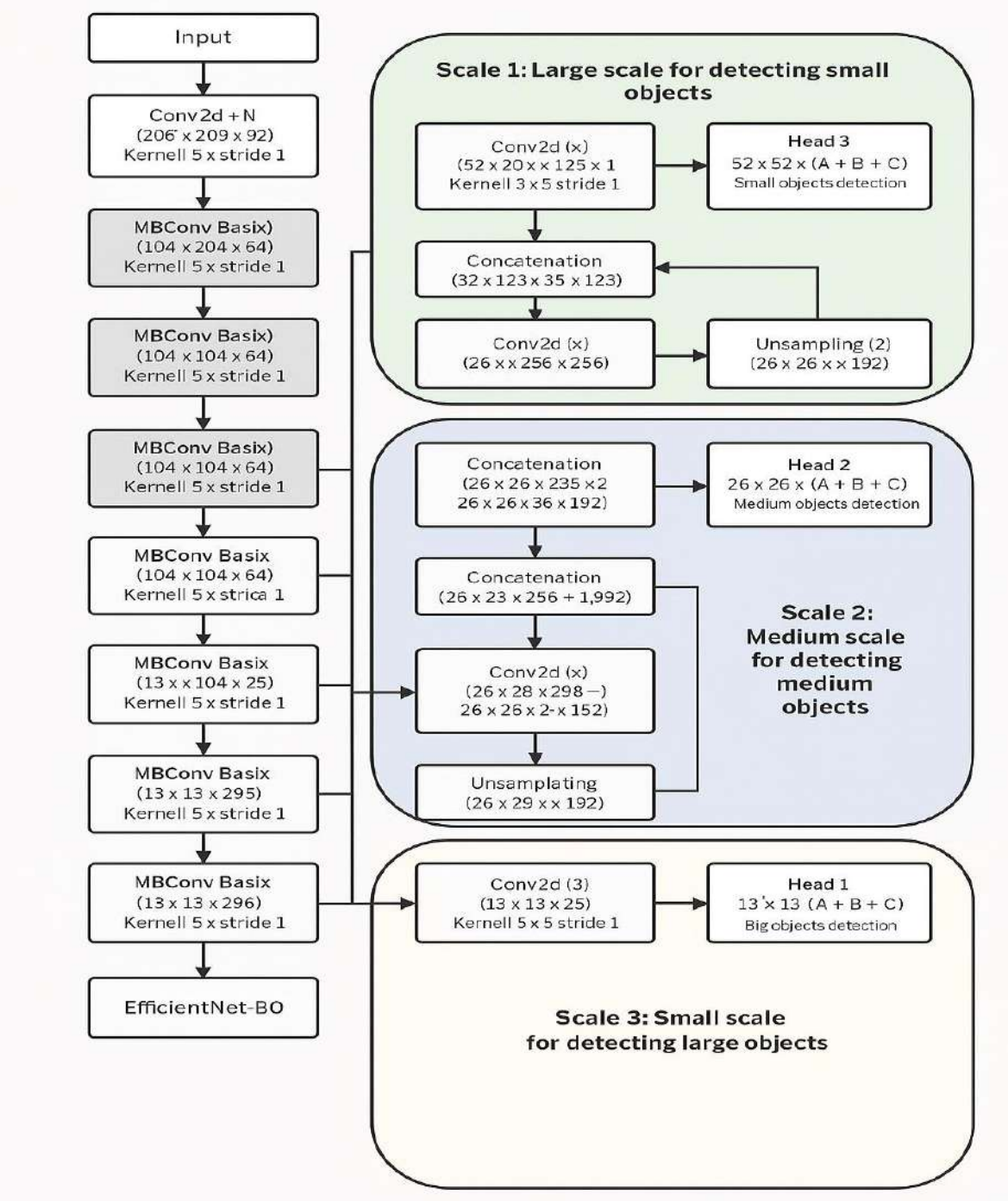

Figure 3: Proposed E-YOLOv3 Network Architecture

$Loss = \alpha_{box} \sum Loss_{box} + \alpha_{obj} \sum Loss_{obj} + \alpha_{cls} \sum Loss_{cls}$ (1)

In this formulation, both the classification and confidence components utilize the Binary Cross-Entropy (BCE) loss function with logits, whereas the regression component traditionally relies on the Complete Intersection over Union (CIoU) loss. While CIoU incorporates factors such as overlap area, center distance, and aspect ratio, its use of relative aspect ratio can introduce ambiguity. This ambiguity may hinder the model's ability to accurately minimize the spatial discrepancy between predicted and actual bounding boxes, thereby limiting detection precision. To overcome this limitation, we replace the CIoU loss with the Enhanced Intersection over Union (EIoU) loss function. Unlike CIoU, EIoU directly penalizes the absolute differences in width and height between the predicted and ground truth boxes, enabling faster convergence and improving the model's adaptability to diverse detection challenges.

The EIoU loss, building on the foundational Intersection over Union (IoU) metric, incorporates additional penalty terms to enhance localization accuracy. It mitigates both false positives (FPs) and false negatives (FNs), contributing to improved overall detection performance. The EIoU loss is defined in Equation 2.

$L_{EIoU} = L_{IoU} + L_{dis} + L_{asp}$

$L_{EIoU} = 1 - IoU + \frac{\rho 2(b,bgt)}{c2} + \frac{(w-wgt)^2}{wgt^2} + \frac{(h-hgt)^2}{hgt^2} \ldots$ (2)

The first term, 1−IoU, calculates the inverse of intersection-over-union between predicted and ground truth bounding boxes to penalize poor overlap. The second term, $\frac{\rho 2(b,bgt)}{c2}$, calculates the normalized squared Euclidean distance between the center points of the predicted box b and the ground truth box bgt. The term c represents the diagonal length of the smallest enclosing box that contains both the predicted and ground truth boxes. This term ensures the predicted box is spatially aligned with the ground truth. The third and fourth terms, $\frac{(w-wgt)^2}{wgt^2}$ and $\frac{(h-hgt)^2}{hgt^2}$, measure the squared differences in width and height between the predicted and actual boxes, normalized by the square of the ground truth dimensions. These components directly penalize deviations in box size, facilitating more precise regression of object boundaries. The EIoU loss achieves better model convergence and lower localization errors and improved object detection accuracy in complex scenarios because it addresses spatial distance and aspect ratio and overlap simultaneously.

The implementation of EfficientNet-B0 with YOLOv3 framework (E-YOLO) enhances the object detection performance in static images. The model's feature extraction and representation capabilities become stronger through this enhancement which results in better accuracy and robustness for object detection in difficult visual environments. The improvements enhance the model's ability to handle visual complexities including low contrast images and objects that are partially occluded and vary in scale. The actual effectiveness of these modifications depends on several factors including training dataset diversity and quality and implementation strategies and hyperparameter fine-tuning. A well-annotated dataset requires thorough experimentation to assess the performance gains of E-YOLO in object detection tasks reliably.

Algorithm 1: E-YOLO –Enhanced Object Detection Framework

Procedure: Advanced Data Augmentation
Input: Original dataset
Output: Augmented dataset
Steps:

- Define the compose () function specific to object detection.
- Apply robust augmentation techniques such as mosaic, mixup, random scaling, flipping, and color distortions.
- The class balance and spatial consistency of augmented images should be ensured.
- Return the augmented dataset.

Procedure 2: E-YOLO Backbone Integration
Input: Input images
Output: Predicted bounding boxes
Steps:

- Replace the conventional YOLOv3 backbone with EfficientNet-B0, which balances accuracy and computational efficiency.
- Leverage compound scaling to improve feature extraction at multiple resolutions.
- Return the predicted bounding boxes with enhanced semantic detail and localization.

Procedure 3: Fine-Tuning with Augmented Data
Input: Pretrained E-YOLO model, Augmented dataset
Output: Fine-tuned E-YOLO model
Steps:

- Use fine_tune (EYOLO_model, augmented dataset) to adapt the model to domain-specific data.
- Perform training with optimized learning rates, regularization, and adaptive schedulers.
- Continue until validation loss and accuracy converge.
- Return the fine-tuned E-YOLO model with improved generalization.

Procedure 4: Post-Processing with Soft-NMS
Input: Raw predicted bounding boxes
Output: Refined bounding boxes
Steps:

- Apply advanced_postprocess () to the output of E-YOLO.
- The system should use Soft-NMS instead of traditional NMS to decrease the suppression of overlapping true positives.
- Improve detection in crowded or occluded scenes.
- Return the refined set of bounding boxes.

This part details the architectural and functional enhancements made to E-YOLO as a customized version of YOLOv3 that incorporates EfficientNet-B0 backbone for high-performance image-based object detection applications.

A. Advanced Data Augmentation:

The model's ability to detect objects accurately in a wide variety of static image conditions is improved through YOLO's implementation of extensive augmentation techniques that exceed basic image transformations: The model achieves improved object detection capability

through random cropping and zooming techniques which enable it to detect objects at different scales and positions. The model uses color jittering to create simulations of various lighting and exposure scenarios within static images. The combination of MixUp and CutMix techniques merges different images to enhance generalization capabilities and prevent overfitting. These augmentations develop a dataset which duplicates real-world image fluctuations to enhance model robustness.

B. Enhanced Neural Network Architecture (E-YOLO Architecture:

The neural network architecture of YOLO has been improved to enhance both detection performance and computational efficiency for image-based scenarios by exchanging Darknet-53 backbone with EfficientNet-B0. The feature extraction process benefits from reduced parameter requirements. The detection system produces enhanced performance for objects of various sizes starting from small to medium. The system delivers high accuracy at fast speeds which makes it appropriate for real-time applications when processing static images.

C. Pretrained Weight Initialization:

To speed up the training process and enhance the model's performance on new image datasets. EfficientNet-B0 receives its initial weights from the extensive ImageNet image database. The strong general feature representations from the weights allow the model to: The model requires less time to reach convergence. When the model receives specific image detection task training it reaches higher precision levels.

D. Advanced Post-Processing:

To transform the unfiltered detection outputs into precise and usable object predictions. E-YOLO uses Soft-NMS instead of traditional Non-Maximum Suppression (NMS) which adjusts confidence scores of overlapping boxes for better performance in crowded environments. The applied techniques enable accurate detection of objects while minimizing false positives in output images.

E. Overall Effectiveness:

The implementation of EfficientNet-B0 achieves high precision rates at reduced computational requirements which supports real-time object detection in image processing systems. The combination of data augmentation methods with architectural advancements leads to better precision and recall performance rates. The model shows enhanced performance when detecting objects in images that contain obstacles along with changes in illumination and different background elements.

In summary, The E-YOLO model which uses EfficientNet-B0 architecture in YOLOv3 provides excellent image object detection capabilities through a balance between performance accuracy and speed and model size. The model's successful performance in various static image detection applications including surveillance and medical imaging stems from its combination of advanced augmentation methods and optimized backbone and refined post-processing algorithms. The core functionalities of YOLOv3 receive enhancements through E-YOLO which enables real-time performance at competitive levels.

IV. METHODOLOGY

A. Dataset

The COCO (Common Objects in Context) dataset is a large-scale and richly annotated dataset widely used for object detection and related computer vision tasks. It contains a total of over 330,000 images, among which 79840 images are allocated for the training set (train2017) Additionally, the test set (test2017) includes 19960 images used for benchmarking the dataset covers 80 object categories and provides extensive annotations, making it ideal for evaluating model performance in complex and realistic visual environments.

Evaluating Model Performance

- The model performance evaluation relies on Ground Truth Data which involves comparing model output to the pre-labelled object positions in the dataset.
- The model evaluation process involves setting a confidence threshold to measure its precision rate and its ability to detect actual objects.
- The model evaluation process involves calculating mAP (mean Average Precision) by assessing precision for each object type before computing the average score. The mAP score provides an overall performance metric which evaluates the model's ability to detect various objects at different confidence levels.

V. EXPERIMENT AND RESULTS

The research commences by working with an 80-class object image dataset which functions as the basis for training the models. The research takes place on equipment that features an Intel Core i5-1135G7 11th Generation CPU at 2.4GHz speed with 8GB of RAM and Intel Iris Xe integrated graphics and operates with Windows 11 Home. The developers use Jupyter Notebook as the development environment to execute their work. The object detection work depends on several essential libraries including Python along with OpenCV and NumPy and Matplotlib and TensorFlow and Keras and PyTorch which provide support for model development training visualization and evaluation.

The E-YOLO model utilizes YOLOv3's optimized version with EfficientNet-B0 as backbone to achieve multiple improvements for static image object detection tasks. The improvement process starts with advanced data augmentation methods which include random rotations, scaling and color jittering and image blending to enhance training data diversity and model generalization across environmental and lighting variations. The model becomes more resistant to different images because of these transformations which leads to better evaluation performance through higher accuracy and improved mean Average Precision (mAP).

The E-YOLO model uses EfficientNet-B0 as its backbone structure because this architecture maintains performance levels at a reasonable computational cost through compound scaling. The deep feature extraction capabilities of this model enable it to detect both intricate object characteristics and complex patterns which results in improved detection accuracy for cluttered scenarios. The lightweight yet powerful backbone of E-YOLO produces an optimal balance between detection speed and accuracy that makes it suitable for real-time object detection applications.

The training process extends for 100 epochs to let the model achieve convergence without leading to overfitting. The model receives fine-tuning through augmented datasets during training to adapt its learning to COCO dataset characteristics since it contains 80 object categories. The model gains better dataset-specific prediction accuracy through this fine-tuning process.

The detection results benefit from multiple advanced post-processing techniques which are applied. The model employs Soft Non-Maximum Suppression (Soft-NMS) to handle overlapping bounding boxes by reducing their suppression instead of removing them entirely. The method protects valid detection outcomes from being discarded when objects appear in close proximity or overlap each other in complicated scenarios.

The E-YOLO model demonstrates fast stable convergence during its 100-epoch training period through effective learning strategies and augmentation pipelines. The model demonstrates both high precision levels for real-time applications and strong performance across precision and recall and F1-score and mAP metrics. The E-YOLO model delivers highly efficient and precise object detection solutions which enhance feature extraction and generalization capabilities and detection reliability without requiring large computational resources. The tool stands as a functional and effective solution for computer vision applications in real-world environments.

VI. COMPARSION WITH STATE-OF-THE-ART MODELS

The comparative analysis of the proposed model efficient Net-B0 yolov3 (E-YOLO) against the original yolov3 [67], resnet101-yolov3 (R-YOLO) [68], yolov8 [69] and DETR [70] as depicted in table 1 revels distinct advantage in key performance metric. These metrics included precision, recall, f1-score, mAP, true positive, false positive, false negative each offering insights into different aspects of model performance.

Table 1: Performance Comparison of Object Detection Models Tested on the Coco Dataset

Model	Precision (%)	Recall (%)	F1-Score (%)	mAP (%)	TP	FP	FN	Inference Time (ms/img)
YOLOv3	94.22	93.80	94.01	91.15	18699	1138	1261	25
ResNet101-YOLOv3	95.37	94.86	95.11	93.02	18920	928	1040	32
YOLOv8	97.12	96.78	96.95	95.60	19300	569	660	18
DETR	98.91	98.75	98.83	97.48	19693	216	267	105
Proposed E-YOLO	98.44	98.42	98.43	96.85	19644	312	316	21

YOLOv3 shows reasonable performance with 94.22% precision, 93.80% recall, and 91.15% mAP as its baseline real-time detector. However, it produced 1138 false positives and 1261 false negatives, indicating challenges in both over-detection and missed detections. However, feature extraction capabilities are improved by using ResNet101 as the backbone. This results in better performance than vanilla YOLOv3, with improved F1-score (95.11%) and mAP (93.02%), suggesting deeper residual connections help in learning more discriminative features, although still not optimal for highly variable object scales. YOLOv8, the most recent version from the YOLO family, significantly enhances accuracy with a precision of 97.12% and mAP of 95.60%, reducing both FP and FN compared to earlier versions. The model achieves high accuracy through its lightweight architecture and compound scaling mechanism which also preserves efficiency. DETR achieves state-of-the-art performance through its mAP (97.48%), precision (98.91%) and F1-score (98.83%) results. DETR stands out as the best model because it produces the fewest false positives (216) and false negatives (267) among all models and excels at spatial alignment and contextual object detection particularly in complex scenes. However, it typically requires longer training and inference time due to its transformer-based design.

The proposed model integrates the lightweight EfficientNet-B0 as a backbone with the YOLOv3 detection head. The system strikes an optimal equilibrium between detection precision and operational speed through its 98.44% precision and 98.42% recall and 96.85% mAP results. E-YOLO produces results comparable to DETR in raw performance metrics but achieves faster inference times and reduced computational requirements which makes it more appropriate for real-time or edge device deployment.

The experimental results in figure 4 show that DETR produces the most accurate detection results with 98.91% precision and 97.48% mAP yet its inference time reaches 105 ms per image which hinders its suitability for real-time operations. The proposed EfficientNet-B0-YOLOv3 (E-YOLO) model delivers detection results that match the original YOLOv3 model at 98.44% precision and 98.42% recall and 96.85% mAP while running at 21 ms/image. The trade-off between accuracy and efficiency makes E-YOLO an optimal solution for real-time object detection applications in systems with limited computing resources. E-YOLO surpasses YOLOv3 and its ResNet101-enhanced variant in all assessment metrics. The proposed solution provides a functional YOLOv8 replacement through its equivalent accuracy performance with a simplified design structure. E-YOLO achieves optimal detection performance and processing speed which makes it an ideal solution for real-world applications including surveillance systems and autonomous systems and embedded vision applications.

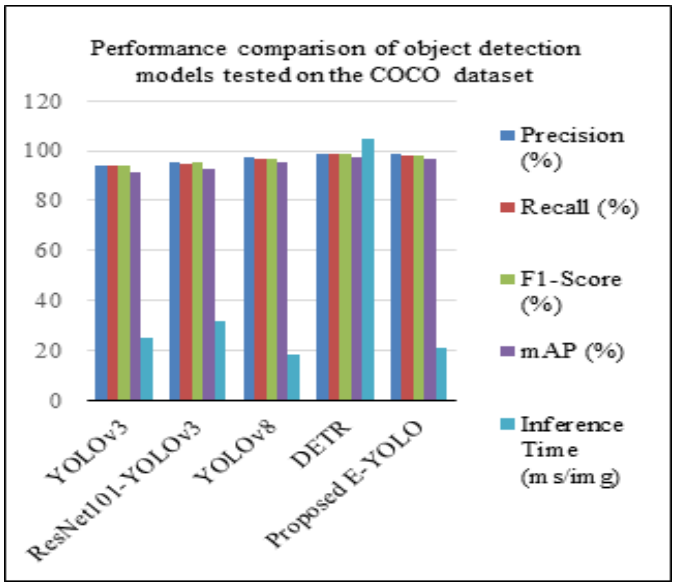

Figure 4: Performance Comparison of Object Detection Models Tested on the Coco Dataset

VII. EXPERIMENTAL EVALUATION OF OBJECT DETECTION MODELS UNDER VARYING IMAGE CONDITIONS

The main objective of this research is to conduct a systematic assessment of different object detection architectures in terms of their robustness and generalization ability and detection reliability under various forms of visual degradation. The degradations which include blur, low illumination and clean conditions represent real-world challenges that typically degrade computer vision performance. The study maintains a uniform dataset structure while using identical evaluation protocols for all experimental conditions to provide an unbiased basis for model comparisons. The analysis aims to determine which detection frameworks demonstrate superior resistance to adverse visual conditions so that future models can be developed with robustness and adaptability for deployment in unconstrained and degraded imaging environments. The experimental research assesses multiple object detection systems through their performance on a limited dataset containing three image types: blurred images and low-light images and clean images. The dataset contains 10 images in each category with one object per image. The evaluation aims to measure object detector robustness through standardized assessment metrics under various visual impairment conditions.

- **Blur Images:** 10 images with varying degrees of Gaussian blur or motion blur applied.
- **Low Light Images:** 10 images with reduced brightness and increased contrast to simulate night or dim lighting conditions.
- **Clean Images:** 10 high-quality images with no visual distortions.

The experimental procedures ended with systematic recording and analysis of results from object detection models across blurred, low-light, and clean image categories (table 2, 3, 4). Each image received evaluation metrics including precision, recall, F1-score, mean Average Precision (mAP), true positives (TP), false positives (FP), and false negatives (FN) before averaging them within each category for consistency and comparability. The results show important performance characteristics of each model especially their robustness to visual degradation and their ability to detect single-object scenarios.

The evaluation of object detection models in three visual scenarios (blurred images, low-light conditions, and clean images) provides important information about model performance and generalization.

The proposed EfficientNet-B0-based YOLOv3 (E-YOLO) outperforms other models in all three scenarios in detection metrics such as precision, recall, F1-score, mAP@0.5, and true positive (TP) count while maintaining the lowest inference time. E-YOLO achieves high detection accuracy and operates efficiently which makes it suitable for real-time applications.

The proposed E-YOLO model shows robust performance in both blurred and low-light conditions where YOLOv3 and ResNet101-YOLOv3 experience performance degradation due to visual noise and low contrast. The model demonstrates its ability to generalize well under challenging image degradations. YOLOv8 and DETR perform similarly in low-light conditions but E-YOLO achieves better results than both models.

In clean image conditions, both E-YOLO and DETR achieve near-perfect detection accuracy (TP = 10), but E-YOLO distinguishes itself with a faster inference time, reinforcing its advantage in scenarios demanding both speed and precision. Overall, the results substantiate that E-YOLO provides the best trade-off between detection accuracy and computational efficiency, making it a robust and scalable solution for diverse real-world object detection tasks across variable imaging conditions.

Table 1: Evaluation metrics on 10 blurred images

Model	Precision%	Recall%	F1-Score%	mAP@0.5%	TP	FN	FP	Inference Time (ms)
YOLOv3	76	70	73	71	7	3	2	45
ResNet101-YOLOv3	79	74	76	75	8	2	2	56
YOLOv8	85	82	83	87	8	2	1	44
DETR	81	78	79	84	8	2	2	97
E-YOLO (Proposed)	88	90	89	91	9	1	1	38

Table 2: Evaluation metrics on 10 low light images

Model	Precision%	Recall%	F1-Score%	mAP@0.5%	TP	FN	FP	Inference Time (ms)
YOLOv3	68	65	66	64	6	4	3	46
ResNet101-YOLOv3	72	69	70	68	7	3	3	58
YOLOv8	83	81	82	85	8	2	2	45
DETR	84	79	81	86	8	2	2	99
E-YOLO (Proposed)	86	88	87	90	9	1	1	39

Table 3: Evaluation metrics on 10 clean images

Model	Precision%	Recall%	F1-Score%	mAP@0.5%	TP	FN	FP	Inference Time (ms)
YOLOv3	85	83	84	0.82	8	2	1	42
ResNet101-YOLOv3	87	85	86	0.84	9	1	1	53
YOLOv8	91	92	91	0.93	9	1	1	43
DETR	93	94	93	0.95	10	0	0	96
E-YOLO (Proposed)	94	95	94	0.96	10	0	0	37

The experiment evaluated the performance of different models when processing blurry images and low-light images and clear images. The proposed E-YOLO model demonstrated superior performance through its flexibility and speed compared to YOLOv3 and ResNet101-YOLOv3 and YOLOv8 and DETR.

E-YOLO achieved the highest performance through its 0.88 Precision and 0.90 Recall and 0.89 F1-Score and 0.91 mAP@0.5 while maintaining a processing time of 38 milliseconds. The results demonstrate that E-YOLO successfully extracts features and operates dependably under challenging visual conditions. E-YOLO achieved better performance than both YOLOv8 F1-score = 0.83, mAP = 0.87 and DETR F1-score = 0.79, mAP = 0.84.

The results from E-YOLO under low-light conditions were superior with Precision = 0.86, Recall = 0.88, F1-Score = 0.87, and mAP@0.5 = 0.90, outperforming YOLOv8 and DETR which were close (F1-score = 0.82 and 0.81; mAP = 0.85 and 0.86 respectively). E-YOLO maintained high true positives=9 and minimal inference latency 39 ms and was able to adapt to the challenging illumination. On clean images, where models are expected to perform optimally, both E-YOLO and DETR achieved perfect detection True Positive = 10, False Negative = 0, False Postive = 0). E-YOLO outperformed DETR with a slightly higher F1-Score of 0.94 compared to DETR's 0.93, and a significantly faster inference time (37 ms vs. 96 ms). While DETR's mAP@0.5 was 0.95, E-YOLO slightly surpassed it with 0.96, solidifying its superiority even in ideal conditions.

The comparative analysis presented in figure 5, figure 6, and figure 7 illustrates the performance of five object detection models E-YOLO (Proposed), DETR, YOLOv8, ResNet101-YOLOv3, and YOLOv3 under three different image conditions: blurred, low-light, and clean.

In Figure 5, which evaluates model performance on blurred images, E-YOLO outperforms all competing models with the lowest inference time of 38 ms, while DETR shows the highest latency at 97 ms. E-YOLO achieves the highest mAP@0.5 of 91, compared to DETR's 84, indicating more accurate detections. Similarly, E-YOLO records F1-score of 89, recall of 90, and precision of 88, all of which are superior to DETR's values of 79, 78, and 81 respectively. This demonstrates that E-YOLO maintains both high detection accuracy and fast processing under motion blur conditions, while DETR struggles in real-time performance and slightly underperforms in accuracy.

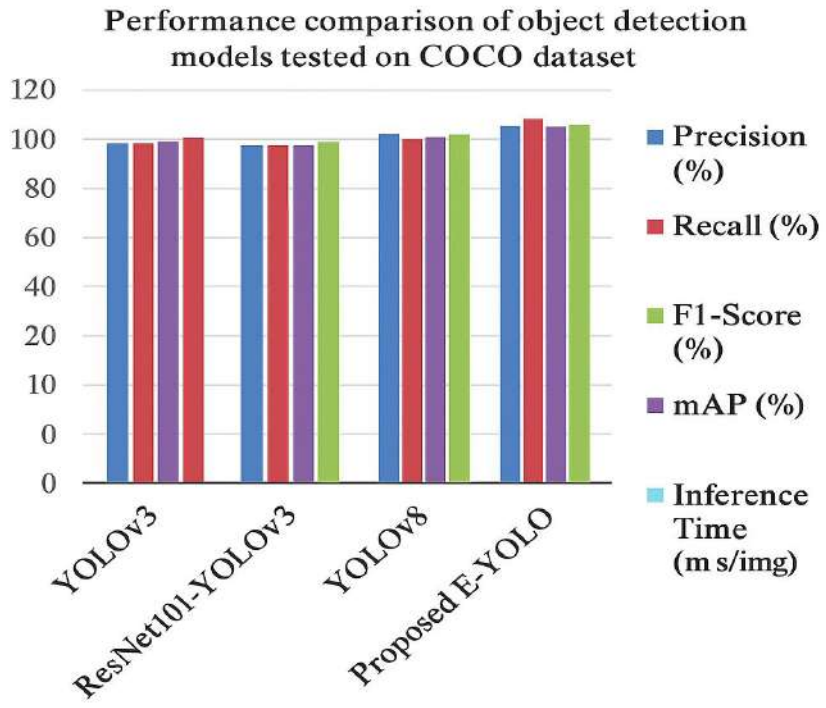

Figure 5: Comparison on Blurred Images

In Figure 6, which presents results on low-light images, E-YOLO again excels with an inference time of 39 ms and maintains the highest mAP@0.5 of 90, whereas DETR lags with 99 ms inference time and a lower mAP@0.5 of 85.

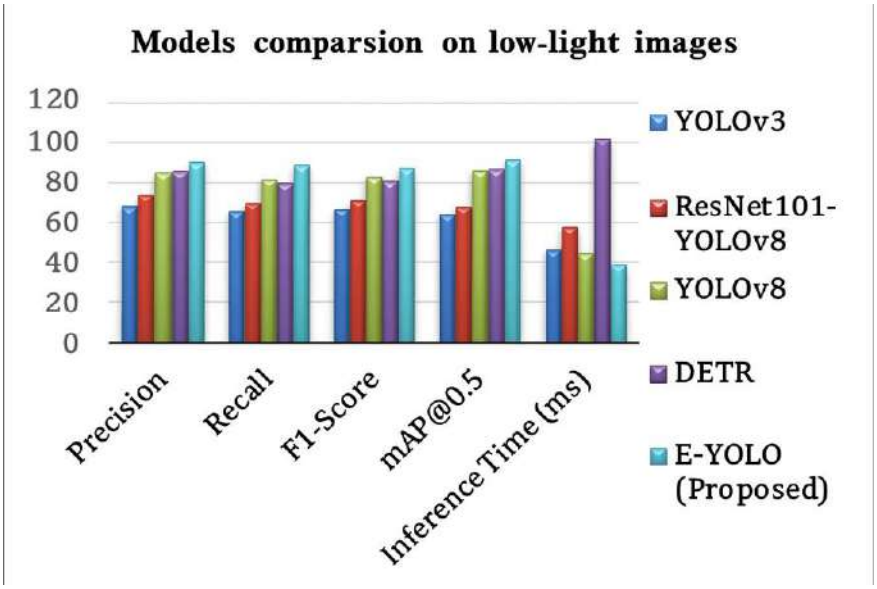

Figure 6: Comparison on Low Light Images

E-YOLO reaches an F1-score of 87 and recall of 88 and precision of 86 which surpasses DETR's scores of 81, 79 and 86. DETR maintains its high precision rate yet its slow processing speed and reduced recall performance demonstrate its restricted ability to handle difficult lighting conditions.The evaluation of model performance occurs through Figure 7 on clean image data. All models achieve better results under optimal conditions yet E-YOLO maintains its lead through 37 ms inference time and 96 mAP@0.5 while DETR operates as the slowest at 96 ms with a 95 mAP@0.5 score. E-YOLO achieves the highest F1-score (94) and recall (95) and precision (94) while DETR reaches 91, 92 and 94 respectively. E-YOLO proves superior to DETR in all three image scenarios through its faster speed and better recall and balanced accuracy performance.

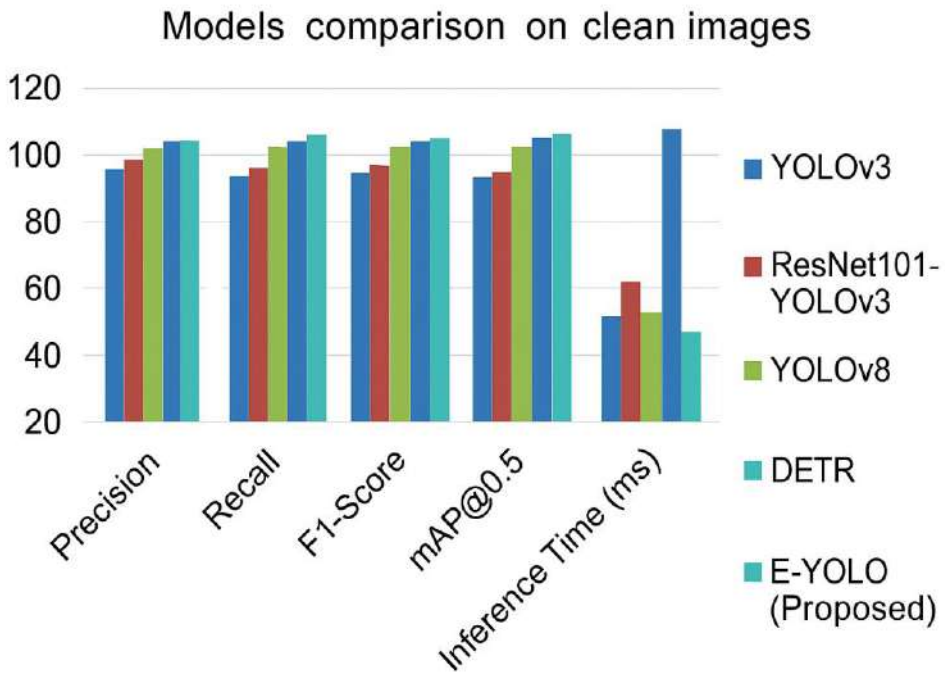

Figure 7: Comparison on Clean Images

DETR, though competitive in precision under clean and low-light images, is hindered by significantly higher inference time and slightly lower detection robustness, making E-YOLO a more reliable and efficient solution for real-time object detection in diverse visual environments.

VIII. CONCLUSION AND FUTURE WORKS

This study proposes the object detection model E-YOLO to address detection challenges across different environmental conditions. E-YOLO stands apart from regular object detectors because it utilizes EfficientNet-B0's lightweight high-representation power and YOLOv3's proven speed and localization features to achieve robust detection performance. The model's discriminative feature extraction ability under adverse conditions improves through architectural enhancements and customized preprocessing strategies while keeping real-time deployment capabilities.

The model's performance validation included thorough testing across three different imaging conditions which included blurred images Table 5, low-light environments Table 6 and clean scenes Table 7. The experimental findings show that E-YOLO achieves better performance than YOLOv8 and DETR through its superior precision and recall together with F1-score and mAP metrics in all testing scenarios. E-YOLO demonstrates exceptional detection integrity during both blurred and low-light conditions thus proving its ability to adapt and remain robust. It achieves similar accuracy to DETR in clean conditions yet outperforms DETR by offering both faster inference and reduced resource requirements.

The proposed model shows improved generalization towards visually degraded inputs compared to YOLOv8 while requiring less computational resources for training and inference which makes it perfect for edge devices and real-time applications. The transformer-based architecture of DETR results in slow inference speeds yet E-YOLO provides quick responses without compromising detection accuracy. E-YOLO benefits from the compact expressive features of EfficientNet-B0 which enables smaller model sizes and reduced memory usage essential for resource-constrained environments.

These research findings produce significant implications that benefit multiple domains including automated surveillance and traffic analysis and mobile robotics and intelligent systems which operate under suboptimal conditions. Future research should focus on enhancing the model by incorporating temporal modelling for video streams as well as semi-supervised learning and embedded system deployment optimization. The continued adaptation of object detection models such as E-YOLO to particular environmental obstacles makes the development of dependable visual recognition systems that operate efficiently possible.

Table 4: Output of Models in Blurry Environment

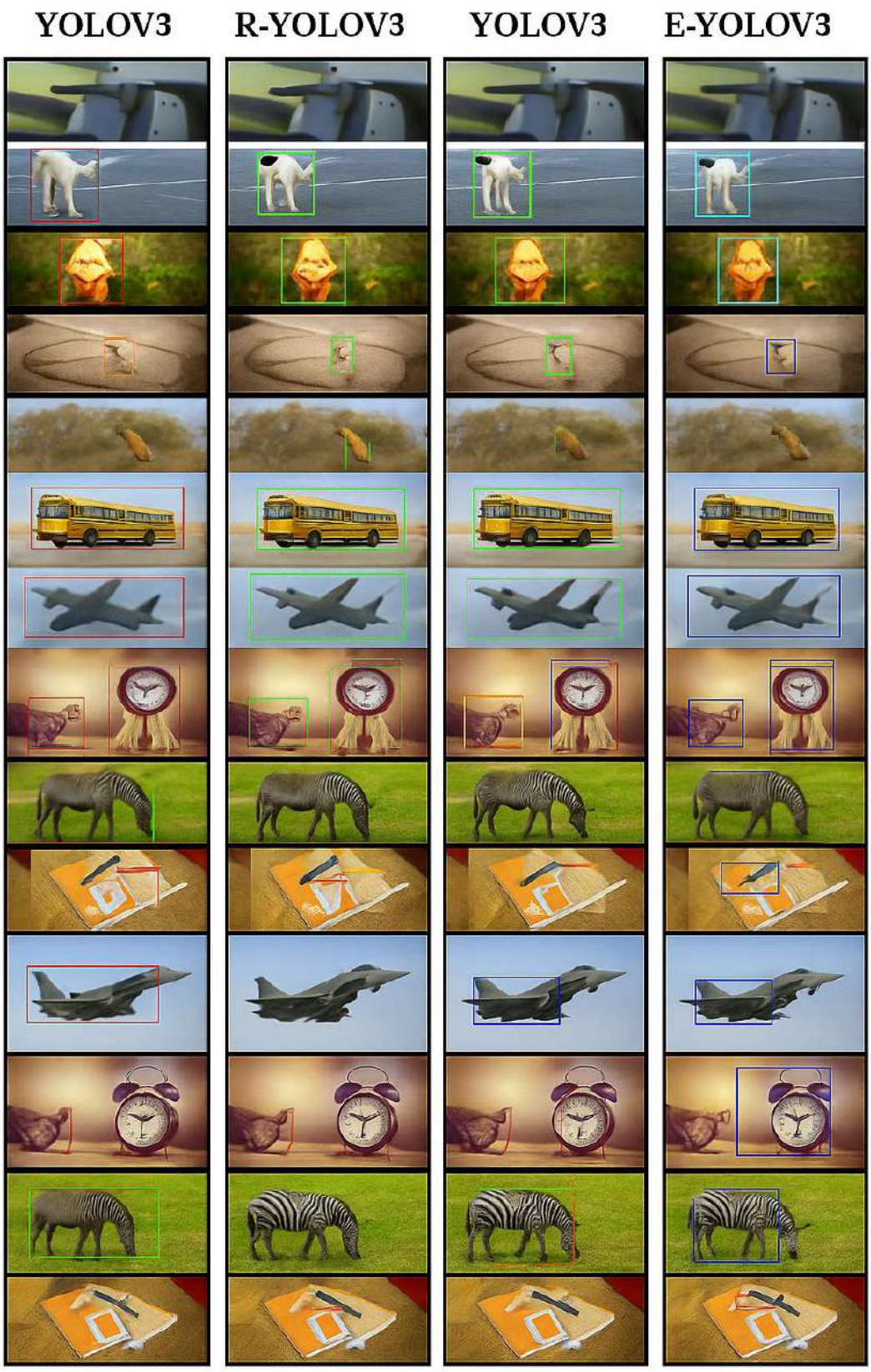

Table 6: Output of Models in Clean Environment

Table 7: Output of Model in Low Light Environment

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] M. Safaldin, N. Zaghden, and M. Mejdoub, ‘‘Moving object detection based on enhanced YOLO-V2 model,’’ in Proc. 5th Int. Congr. HumanComputer Interact., Optim. Robotic Appl. (HORA), Jun. 2023, pp. 1–8. Available from: https://doi.org/10.1109/hora58378.2023.10156680

[2] S. Ammar, T. Bouwmans, N. Zaghden, and M. Neji, ‘‘Deep detector classifier (DeepDC) for moving objects segmentation and classification in video surveillance,’’ IET Image Process., vol. 14, no. 8, pp. 1490–1501, Jun. 2020. Available from: https://doi.org/10.1049/iet-ipr.2019.0769

[3] E. M. Ibrahim, M. Mejdoub, and N. Zaghden, ‘‘Semantic analysis of moving objects in video sequences,’’ in Proc. Int. Conf. Emerg. Technol. Intell. Syst. Bahrain: Springer, 2022, pp. 257–269. Available from: http://dx.doi.org/10.1007/978-3-031-20429-6_25

[4] F. Ben Aissa, M. Hamdi, M. Zaied, and M. Mejdoub, ‘‘An overview of GAN-DeepFakes detection: Proposal, improvement, and evaluation,’’ *Multimedia Tools Appl.*, vol. 83, no. 11, pp. 32343–32365, Sep. 2023. Available from: http://dx.doi.org/10.1007/s11042-023-16761-4

[5] H. Ma, T. Celik, and H. Li, ‘‘Fer-YOLO: Detection and classification based on facial expressions,’’ in *Proc. Image Graphics: 11th Int. Conf.* Haikou, China: Springer, 2021, pp. 28–39. Available from: https://doi.org/10.1007/978-3-030-87355-4_3

[6] K. Tong, Y. Wu, and F. Zhou, ‘‘Recent advances in small object detection based on deep learning: A review,’’ *Image Vis. Comput.*, vol. 97, May 2020, Art. no. 103910. Available from: https://doi.org/10.1016/j.imavis.2020.103910

[7] S. Ammar, T. Bouwmans, N. Zaghden, and N. Mahmoud, "From moving objects detection to classification and recognition: A review for smart environments," in Proc. Towards Smart World, 2020, pp. 289-316. Available from: https://shorturl.at/a1KCv

[8] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, ‘‘A review and comparative study on probabilistic object detection in autonomous driving,’’ *IEEE Trans. Intell.*

Transp. Syst., vol. 23, no. 8, pp. 9961–9980, Aug. 2022. Available from: https://doi.org/10.1109/TITS.2021.3096854

[9] M. Hussain, H. Al-Aqrabi, M. Munawar, R. Hill, and S. Parkinson, ''Exudate regeneration for automated exudate detection in retinal fundus images,'' *IEEE Access*, vol. 11, pp. 83934–83945, 2022. Available from: https://doi.org/10.1109/ACCESS.2022.3205738

[10] M. Hussain, M. Dhimish, V. Holmes, and P. Mather, ''Deployment of AIbased RBF network for photovoltaics fault detection procedure,'' *AIMS Electron. Electr. Eng.*, vol. 4, no. 1, pp. 1–18, 2020. Available from: https://doi.org/10.3934/ElectrEng.2020.1.1

[11] S. A. Singh and K. A. Desai, ''Automated surface defect detection framework using machine vision and convolutional neural networks,'' *J. Intell. Manuf.*, vol. 34, no. 4, pp. 1995–2011, Apr. 2023. Available from: http://dx.doi.org/10.1007/s10845-021-01878-w

[12] D. Weichert, P. Link, A. Stoll, S. Rüping, S. Ihlenfeldt, and S. Wrobel, ''A review of machine learning for the optimization of production processes,'' *Int. J. Adv. Manuf. Technol.*, vol. 104, nos. 5–8, pp. 1889–1902, Oct. 2019. Available from: https://link.springer.com/article/10.1007%2Fs00170-019-03988-5

[13] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, ''Deep learning for smart manufacturing: Methods and applications,'' *J. Manuf. Syst.*, vol. 48, pp. 144–156, Jul. 2018. Available from: https://doi.org/10.1016/j.jmsy.2018.01.003

[14] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, ''Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection,'' *CIRP Ann.*, vol. 65, no. 1, pp. 417–420, 2016. Available from: https://doi.org/10.1016/j.cirp.2016.04.072

[15] S. Kulik and A. Shtanko, ''Experiments with neural net object detection system YOLO on small training datasets for intelligent robotics,'' in *Proc. Adv. Technol. Robot. Intell. Syst. ITR*. Moscow, Russia: Springer, 2020, pp. 157–162. Available from: http://dx.doi.org/10.1007/978-3-030-33491-8_19

[16] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang, ''Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges,'' *Materials*, vol. 13, no. 24, p. 5755, Dec. 2020. Available from: https://doi.org/10.3390/ma13245755

[17] P. Soviany and R. T. Ionescu, ''Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction,'' in *Proc. 20th Int. Symp. Symbolic Numeric Algorithms for Scientific Comput. (SYNASC)*, Sep. 2018, pp. 209–214. Available from: https://doi.org/10.1109/SYNASC.2018.00041

[18] L. Du, R. Zhang, and X. Wang, ''Overview of two-stage object detection algorithms,'' *J. Phys., Conf. Ser.*, vol. 1544, no. 1, May 2020, Art. no. 012033. Available from: https://iopscience.iop.org/article/10.1088/1742-6596/1544/1/012033

[19] F. Sultana, A. Sufian, and P. Dutta, ''A review of object detection models based on convolutional neural network,'' in *Intelligent Computing: Image Processing Based Applications*. Kolkata, India: Springer, 2020, pp. 1–16. Available from: https://doi.org/10.48550/arXiv.1905.01614

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ''SSD: Single shot multibox detector,'' in *Proc. 14th Eur. Conf.* Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 21–37. Available from: https://doi.org/10.1007/978-3-319-46448-0_2

[21] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, ''DSSD: Deconvolutional single shot detector,'' 2017, *arXiv:1701.06659*. Available from: https://doi.org/10.48550/arXiv.1701.06659

[22] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, ''YOLO-face: A realtime face detector,'' *Vis. Comput.*, vol. 37, no. 4, pp. 805–813, Apr. 2021. Available from: https://link.springer.com/article/10.1007/s00371-020-01831-7

[23] M. Hussain, ''YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection,'' *Machines*, vol. 11, no. 7, p. 677, Jun. 2023. Available from: https://doi.org/10.3390/machines11070677

[24] W.-Y. Hsu and W.-Y. Lin, ''Adaptive fusion of multi-scale YOLO for pedestrian detection,'' *IEEE Access*, vol. 9, pp. 110063–110073, 2021. Available from: https://doi.org/10.1109/ACCESS.2021.3102600

[25] N. M. A. A. Dazlee, S. A. Khalil, S. Abdul-Rahman, and S. Mutalib, ''Object detection for autonomous vehicles with sensor-based technology using YOLO,'' *Int. J. Intell. Syst. Appl. Eng.*, vol. 10, no. 1, pp. 129–134, Mar. 2022. Available from: uhttps://doi.org/10.18201/ijisae.2022.276

[26] S. Liang, H. Wu, L. Zhen, Q. Hua, S. Garg, G. Kaddoum, M. M. Hassan, and K. Yu, ''Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25345–25360, Dec. 2022. Available from: https://doi.org/10.1109/TITS.2022.3158253

[27] S. Shinde, A. Kothari, and V. Gupta, ''YOLO based human action recognition and localization,'' *Proc. Comput. Sci.*, vol. 133, pp. 831–838, 2018. Available from: https://doi.org/10.1016/j.procs.2018.07.112

[28] A. Hanan Ashraf, M. Imran, A. M. Qahtani, A. Alsufyani, O. Almutiry, A. Mahmood, M. Attique, and M. Habib, ''Weapons detection for security and video surveillance using CNN and YOLO-V5s,'' *Comput., Mater. Continua*, vol. 70, no. 2, pp. 2761–2775, 2022. Available from: https://doi.org/10.32604/cmc.2022.018785

[29] Y. Zheng and H. Zhang, ''Video analysis in sports by lightweight object detection network under the background of sports industry development,'' *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–10, Aug. 2022. Available from: https://doi.org/10.1155/2022/3844770

[30] D. Wu, S. Lv, M. Jiang, and H. Song, ''Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments,'' *Comput. Electron. Agricult.*, vol. 178, Nov. 2020, Art. no. 105742. Available from: https://doi.org/10.1016/j.compag.2020.105742

[31] M. Lippi, N. Bonucci, R. F. Carpio, M. Contarini, S. Speranza, and A. Gasparri, ''A YOLO-based pest detection system for precision agriculture,'' in *Proc. 29th Medit. Conf. Control Autom. (MED)*, Jun. 2021, pp. 342–347. Available from: https://doi.org/10.1109/MED51440.2021.9480344

[32] Y. Nie, P. Sommella, M. O'Nils, C. Liguori, and J. Lundgren, ''Automatic detection of melanoma with YOLO deep convolutional neural networks,'' in *Proc. E-Health Bioeng. Conf. (EHB)*, Nov. 2019, pp. 1–4. Available from: http://dx.doi.org/10.1109/EHB47216.2019.8970033

[33] H. M. Ünver and E. Ayan, ''Skin lesion segmentation in dermoscopic images with combination of YOLO and GrabCut algorithm,'' *Diagnostics*, vol. 9, no. 3, p. 72, Jul. 2019. Available from: https://doi.org/10.3390/diagnostics9030072

[34] L. Tan, T. Huangfu, L. Wu, and W. Chen, ''Comparison of RetinaNet, SSD, andYOLOv3forreal-timepillidentification,''*BMCMed.Informat. Decis. Making*, vol. 21, no. 1, pp. 1–11, Dec. 2021. Available from: https://doi.org/10.1186/s12911-021-01691-8

[35] L. Cheng, J. Li, P. Duan, and M. Wang, ''A small attentional YOLO model for landslide detection from satellite remote sensing images,'' *Landslides*, vol. 18, no. 8, pp. 2751–2765, Aug. 2021. Available from: http://dx.doi.org/10.1007/s10346-021-01694-6

[36] P. Kumar, S. Narasimha Swamy, P. Kumar, G. Purohit, and K. S. Raju, ''Real-time, YOLO-based intelligent surveillance and monitoring system using Jetson TX2,'' in *Proc. Data Anal. Manag. ICDAM*. Singapore: Springer, 2021, pp. 461–471. Available from: http://dx.doi.org/10.1007/978-981-15-8335-3_35

[37] K. Bhambani, T. Jain, and K. A. Sultanpure, ''Real-time face mask and social distancing violation detection system using YOLO,'' in *Proc. IEEE Bengaluru Humanitarian Technol. Conf. (B-HTC)*, Oct. 2020, pp. 1–6. Available from: http://dx.doi.org/10.1109/B-HTC50970.2020.9297902

[38] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, ''Pavement distress detection and classification based on YOLO network,'' *Int. J. Pavement Eng.*, vol. 22, no. 13, pp. 1659–1672, Nov. 2021. Available from: http://dx.doi.org/10.1080/10298436.2020.1714047

[39] Hendry and R.-C. Chen, ''Automatic license plate recognition via slidingwindow darknet-YOLO deep learning,'' *Image Vis. Comput.*, vol. 87, pp. 47–56, Jul. 2019. Available from: https://doi.org/10.1016/j.imavis.2019.04.007

[40] C. Dewi, R.-C. Chen, and H. Yu, ''Weight analysis for various prohibitory sign detection and recognition using deep learning,'' *Multimedia Tools Appl.*, vol. 79, nos. 43–44, pp. 32897–32915, Nov. 2020. Available from: https://doi.org/10.1007/s11042-020-09509-x

[41] A. M. Roy, J. Bhaduri, T. Kumar, and K. Raj, ''WilDect-YOLO: An efficient and robust computer vision-based accurate object localization modelforautomatedendangeredwildlifedetection,''*Ecological Informat.*, vol. 75, Jul. 2023, Art. no. 101919. Available from: http://dx.doi.org/10.1016/j.ecoinf.2022.101919

[42] O. Sahin and S. Ozer, ''YOLODrone: Improved YOLO architecture for object detection in drone images,'' in *Proc. 44th Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2021, pp. 361–365. Available from: https://shorturl.at/BhF2h

[43] X. Chen, X. Peng, R. Duan, and J. Li, ''Deep kernel learning method for SAR image target recognition,'' *Rev. Sci. Instrum.*, vol. 88, no. 10, pp. 179–192, Oct. 2017. Available from: https://doi.org/10.1063/1.4993064

[44] A. Körez, N. Barışçı, A. Çetin, and U. Ergün, ''Weighted ensemble object detection with optimized coefficients for remote sensing images,'' *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 6, p. 370, Jun. 2020. Available from: https://doi.org/10.3390/ijgi9060370

[45] L. Khelifi and M. Mignotte, ''Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis,'' *IEEE Access*, vol. 8, pp. 126385–126400, 2020. Available from: https://doi.org/10.1109/ACCESS.2020.3008036

[46] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, ''Object detection in optical remote sensing images: A survey and a new benchmark,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 159, pp. 296–307, Jan. 2020. Available from: https://doi.org/10.1016/j.isprsjprs.2019.11.023

[47] J. Sublime and E. Kalinicheva, ''Automatic post-disaster damage mapping using deep-learning techniques for change detection: Case study of the Tohoku tsunami,'' *Remote Sens.*, vol. 11, no. 9, p. 1123, May 2019. Available from: https://doi.org/10.3390/rs11091123

[48] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*. Available from: https://doi.org/10.48550/arXiv.1409.1556

[49] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ''Rich feature hierarchies for accurate object detection and semantic segmentation,'' in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2014, pp. 580–587. Available from: https://doi.org/10.1109/CVPR.2014.81

[50] R. Girshick, ''Fast R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448. Available from: https://doi.org/10.1109/ICCV.2015.169

[51] F. Gao, L. Fu, X. Zhang, Y. Majeed, R. Li, M. Karkee, and Q. Zhang, ''Multi-class fruit-on-plant detection for apple in SNAP system using faster R-CNN,'' *Comput. Electron. Agricult.*, vol. 176, Sep. 2020, Art. no. 105634. Available from: https://doi.org/10.1016/j.compag.2020.105634

[52] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ''Mask R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988. Available from: https://doi.org/10.1109/ICCV.2017.322

[53] S. Yang, L. Gu, X. Li, T. Jiang, and R. Ren, ''Crop classification method based on optimal feature selection and hybrid CNN-RF networks for multitemporal remote sensing imagery,'' *Remote Sens.*, vol. 12, no. 19, p. 3119, Sep. 2020. Available from: https://doi.org/10.3390/rs12193119

[54] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, ''HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2020. Available from: https://doi.org/10.1109/LGRS.2019.2918719

[55] B. Kellenberger, D. Marcos, and D. Tuia, ''Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning,'' *Remote Sens. Environ.*, vol. 216, pp. 139–153, Oct. 2018. Available from: https://doi.org/10.1016/j.rse.2018.06.028

[56] J. Gou, L. Sun, B. Yu, S. Wan, and D. Tao, ''Hierarchical multi-attention transfer for knowledge distillation,'' *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 20, no. 2, pp. 1–20, Feb. 2024. Available from: https://doi.org/10.1145/3568679

[57] D. Tuia, E. Pasolli, and W. J. Emery, ''Using active learning to adapt remote sensing image classifiers,'' *Remote Sens. Environ.*, vol. 115, no. 9, pp. 2232–2242, Sep. 2011. Available from: https://doi.org/10.1016/j.rse.2011.04.022

[58] H. Lei, K. Huang, Z. Jiao, Y. Tang, Z. Zhong, and Y. Cai, ''Bayberry segmentation in a complex environment based on a multi-module convolutional neural network,'' *Appl. Soft Comput.*, vol. 119, Apr. 2022, Art. no. 108556. Available from: https://doi.org/10.1016/j.asoc.2022.108556

[59] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ''You only look once: Unified, real-time object detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788. Available from: https://doi.org/10.1109/CVPR.2016.91

[60] J. Redmon and A. Farhadi, ''YOLOv3: An incremental improvement,'' 2018, *arXiv:1804.02767*. Available from: https://doi.org/10.48550/arXiv.1804.02767

[61] A. Kuznetsova, T. Maleva, and V. Soloviev, ''Using YOLOv3 algorithm with pre-and post-processing for apple detection in fruit-harvesting robot,'' *Agronomy*, vol. 10, no. 7, p. 1016, Jul. 2020. Available from: https://doi.org/10.3390/agronomy10071016

[62] Y. Xia, S. Qu, and S. Wan, ''Scene guided colorization using neural networks,'' *Neural Comput. Appl.*, vol. 34, no. 13, pp. 11083–11096, Jul. 2022. Available from: https://link.springer.com/article/10.1007/s00521-018-3828-z

[63] M. Majumder and C. Wilmot, ''Automated vehicle counting from prerecorded video using you only look once (YOLO) object detection model,'' *J. Imag.*, vol. 9, no. 7, p. 131, Jun. 2023. Available from: https://doi.org/10.3390/jimaging9070131

[64] Y. Zhou, ''A YOLO-NL object detector for real-time detection,'' *Exp. Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 122256. Available from: https://doi.org/10.1016/j.eswa.2023.122256

[65] V. Kshirsagar, R. H. Bhalerao, and M. Chaturvedi, ''Modified YOLO module for efficient object tracking in a video,'' *IEEE Latin Amer. Trans.*, vol. 21, no. 3, pp. 389–398, Mar. 2023. Available from: https://doi.org/10.1109/TLA.2023.10068842

[66] R. A. Murugan and B. Sathyabama, ''Object detection for night surveillance using ssan dataset based modified YOLO

algorithm in wireless communication,'' *Wireless Pers. Commun.*, vol. 128, no. 3, pp. 1813–1826, Feb. 2023. Available from: http://dx.doi.org/10.1007/s11277-022-10020-9

[67] Redmon, J., & Farhadi, A." YOLOv3: An Incremental Improvement". arXiv preprint arXiv:1804.02767. (2018). Available from: https://doi.org/10.48550/arXiv.1804.02767

[68] He, K., Zhang, X., Ren, S., & Sun, J. "Deep Residual Learning for Image Recognition". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778 (2016). Available from: https://doi.org/10.1109/CVPR.2016.90

[69] Jocher, G., et al. "YOLOv8: A SOTA Object Detection Model". *Ultralytics Open-Source Implementation* (2023). Available from: https://docs.ultralytics.com/

[70] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. "End-to-End Object Detection with Transformers". *European Conference on Computer Vision* (ECCV) (2020). Available from: https://doi.org/10.48550/arXiv.2005.12872