EL-MODC: A SegNet and ResNet50-Based Deep Learning Framework for Multi-Object Detection and Classification

*1M Srividya, 2Venubabu Rachapudi
1Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India -522302
2Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522302
E-mail: madugulasrividya@gmail.com, venubabu.r@gmail.com
*Corresponding author

Keywords: multi-object detection, image segmentation, deep learning, transfer learning, image data augmentation

Received: December 25, 2024

Multi-object detection and classification are essential tasks in computer vision, with applications in autonomous navigation, healthcare diagnostics, and surveillance. Current models are confronted with problems such as intricate object boundaries, class imbalance, and real-time extension. To overcome these limitations, a new deep learning model—EL-MODC (Enhanced Learning-based Multi-Object Detection and Classification) architecture —is introduced. This architecture comprises a SegNet for accurate pixel-wise image segmentation and a modified ResNet-50, based on transfer learning. The architecture begins with robust data preprocessing and augmentation to enhance model capacity and robustness. With SegNet, the spatial localization of object regions can be efficiently carried out, and ResNet-50 can utilize the pre-trained weights of ImageNet to enhance the efficiency of learning and prediction. The performance of the proposed model was tested by comparing it with several baseline models, including UNet, Baseline CNN, and LeNet, showing that the proposed architecture achieved an accuracy of 96.40%, a precision of 96.02%, a recall of 96.32%, and an F1-score of 96.16%. EL-MODC slightly outperforms LeNet with a 95.19% F1-score and 95.07% accuracy, demonstrating enhanced performance in the test conditions described in the complex scenarios. Additionally, the model achieves an IoU of 86.5%, further reinforcing its strong generalization capability across object boundaries. Moreover, we present an in-depth performance analysis based on confusion matrices, detection visualization, and class distribution, which further demonstrates the robustness of the proposed system. EL-MODC is generalizable and applicable to varied domains and real-world environments. It can be deployed for real-time purposes, featuring a modular and efficient computational architecture. The proposed framework not only outperforms existing state-of-the-art models but also paves the way for potential improvements in multi-object detection, particularly in terms of interpretability and cross-domain adaptation.

Povzetek: Članek s področja računalniškega vida predstavlja okvir EL-MODC, ki združuje SegNet in ResNet50 za večobjektno detekcijo. Novost je učinkovita integracija segmentacije in transfernega učenja, kar zagotavlja bolj kvalitetno razmejevanje meja in klasifikacijo kot metoda YOLOv3.

1 Introduction

One of the significant ways this smoother approximation is utilized is for object detection and classification, which are essential to computer vision applications across various domains, including surveillance, autonomous driving, healthcare, and robotics. While there has been a virtuous process of improving the principles of deep learning for the epigraph, there are still significant drawbacks to accurate segmentation like our inability to identify complex object boundaries, the considerable variability in object sizes and orientations, as well as real-time applicability for computer vision systems which do not treat their data source as a stack of images but a stream. Although methods like YOLO, SSD and ResNet have demonstrated effectiveness, but multi-object detection and classification for diverse and complex scenarios still suffer from accuracy issues. The literature is reviewed, and significant advances in the field are discussed. For example, hybrid models such as YOLO-ResNet and segmentation-integrative frameworks have enhanced the detection performance. The challenges, such as dealing with background complexity, real-time processing, and multiclass imbalance, remain. Furthermore, current approaches are primarily brittle across various application contexts, highlighting the necessity for more adaptive and accurate procedures.

To mitigate these challenges, we introduce a deep learning framework and an Enhanced Learning-based Multi-Object Detection and Classification (EL-MODC) algorithm in the present research. The purpose is to create a more efficient and highly skilled logistics system that integrates SegNet for accurate segmentation and ResNet50 with transfer learning for hierarchical feature extraction and multi-class CSS classification. Our proposed framework prioritizes accuracy, computational efficiency, and scalability. The pilot study contributes original innovation by leveraging transfer learning in a well-established precision segmentation model and sophisticated feature extraction to improve processing performance. The algorithm, designed to provide precise solutions for these components, enables their integration without risk, offering high accuracy and low computational effort. Here are the primary contributions: (1) Proposal of the EL-MODC framework, (2) Experimental validation on the VOCdevkit dataset, and (3) Benchmarking against state-of-the-art models. The paper presents extensive evaluation metrics to demonstrate the framework's superiority. The remainder of the paper is organized as follows: Section 2 contains relevant work; Section 3 outlines the suggested technique; Section 4 discusses the experimental findings; Section 5 presents our work and discusses the study's limitations; and Section 6 concludes with suggestions for future research.

2 Related work

The literature demonstrates notable developments in deep learning (DL) frameworks for detecting and categorizing multiple objects, including YOLO, SegNet, and ResNet-based models. These works address challenges in accuracy, real-time processing, and complex scenarios, setting the foundation for innovative solutions in this domain. Kuppuswamy and Hung [1] highlighted the better performance of CNN in object recognition by comparing it with the Firefly algorithm and SVM with the Adam optimizer. Video action recognition will be a focus of future research. Lu *et al.* [2] presented a hybrid network called YOLO-ResNet for enhanced multi-object identification, beating conventional techniques with an accuracy of 75.36%. To improve accuracy, Alazeb et al. [3] developed a scene recognition framework including UNet, DWT, and AlexNet. Further work addresses challenges in handling complex backgrounds and further explores deep learning techniques. Pal *et al.* [4] examined developments in deep learning for tracking and object detection, emphasizing breakthroughs, present difficulties, and potential lines of inquiry. Pramanik *et al.* [5] Introduced the G-RCNN and MCD-SORT models, which show remarkable performance, to improve movie object detection and tracking. Future studies will look at applications of SAR imaging.

Naseer *et al.* [6] offered improved multi-object recognition methods that produce outstanding accuracy by utilizing Gaussian mixture models and MLP models. Future research efforts are concentrated on enhancing real-time image analysis and supporting different scenarios. Fink *et al.* [7] addressed issues with identifying different object sizes and balancing class biases to improve multi-object recognition for autonomous driving by improving SSD design. Upcoming projects will focus on managing various situations and improving real-time capabilities. Mhalla *et al.* [8] provided a robust embedded traffic surveillance system that utilizes a novel deep detector to improve multi-object detection accuracy. Upcoming studies will enhance the spatiotemporal data and visual signals. Mohandoss and Rangaraj [9] employed a LuNet and YOLOv2-based technique to enhance multi-object tracking, achieving a 94% accuracy rate. This approach aspires to future developments and comprehensive testing. Elhoseny [10] presented a MODT approach based on Kalman filtering, which achieves 86.78% tracking accuracy and 76.23% detection accuracy; more improvements are required.

Ahn and Cho [11] addressed accuracy and error concerns by introducing a CNN and optical flow-based method for real-time multi-object tracking and identification. Yuan *et al.* [12] presented an approach called the Multi-Path Extraction Network (MPEN) to enhance the accuracy of millimeter-wave SAR picture detection and recognition. Li *et al.* [13] demonstrate outstanding accuracy, but their limitations become apparent when applied to dynamic settings. To enhance multi-object detection in complex traffic, VGG16 accelerates R-CNN. Ravindran et al. [14] examined sensor fusion and DNN-based multi-object tracking and detection in autonomous vehicles, highlighting issues related to sensor reliability and real-time efficiency. Premanand and Dhananjay [15] proposed using MRNN with the PS-KM algorithm to achieve precise and effective Multiple-Object Tracking (MOT) with 97% accuracy in navigating complex surroundings.

Wen *et al.* [16] highlighted the effect of detection accuracy on tracking performance and presented the UA-DETRAC dataset for assessing MOT systems. Štruc et al. [17] designed a novel deep learning framework called deep residual learning for object classification and localization. They demonstrated that deep residual learning extracts strong features that generalize well across datasets for highly image-dense recognition. Mauri *et al.* [18] presented a high-accuracy real-time 3D object recognition technique for rail and road settings utilizing YOLOv3. Upgrading to YOLOv5 and improving ROI forecasts are among the following tasks. Rong *et al.* [19] employed an updated K-means algorithm to enhance the speed and accuracy of YOLOv3 object detection. Subsequent research endeavors will tackle constraints in severe scenarios and incorporate contour recognition. Arora *et al.* [20] presented a real-time object identification prototype that utilizes deep learning and vocal cues to enhance mobility for individuals who are blind. Future developments will provide additional cameras, more sophisticated algorithms, and lower latency.

Sun *et al.* [21] improved SSD-based target identification by combining attention processes with multi-scale feature extraction, resulting in a 7.4% increase in accuracy at the same speed. Further accuracy enhancements and integration with additional datasets will be investigated in further work. Rahman *et al.* [22] proposed an enhanced CNN-based image identification technique that improves accuracy and reduces processing overhead. Future efforts will primarily focus on striking a balance between accuracy and processing time. Mauri *et al.* [23] developed a YOLOv3-based system for real-time multi-object tracking and recognition in road scenarios by integrating depth estimates. The primary goal of further work will be to obtain new datasets for training setups. Mhalla *et al.* [24] introduced a novel Multi-Object Tracking-by-Detection (MOT-bD) framework utilizing deep convolutional neural networks (DCNN) and interlaced images. Future work on automatic specialization will be focused on this topic. Wang *et al.* [25] utilize the RFD technique to enhance target detection accuracy and decrease false alarm rates by employing region-focused feature extraction. Future work will focus on novel item types and various scenarios.

Ahmed *et al.* [26] suggested that the scene categorization method uses logistic regression, MFCS, and MSS to increase accuracy. Subsequent research will use deep learning to minimize complexity and improve accuracy. Wang *et al.* [27] proposed a model that improves upon YOLOv3 for real-time multi-object detection, outperforming traditional methods in terms of speed and accuracy. In subsequent development, these systems will undergo significant improvements. Liu *et al.* [28] enhanced the YOLOv3 algorithm to quickly and precisely recognize multiple license plates in complicated scenarios. The dataset's class imbalance will be addressed later in the work. Ramya and James [29] suggested utilizing spatial pyramid matching in conjunction with SURF and MSER to enhance the efficiency and accuracy of item recognition. More advancements will be investigated in subsequent research. Du *et al.* [30] combined semantic corner detection with customized datasets to enhance YOLOv3 for multi-object grabbing recognition, achieving high accuracy.

Table 1: Summary of literature findings

References	Authors & Year	Techniques	Algorithm	Dataset	Limitations
[1]	Kuppusamy and Hung [2021]	Convolutional Neural Network	Support Vector Machine (SVM) is optimized using the Firefly Algorithm (FA)	VOC2012 dataset	Subsequent research would be expanded to determine the action from a video's frame sequence.
[3]	Alazeb *et al.*, [2024]	deep learning	Object recognition	PASCALVOC-12 dataset	By applying various deep learning approaches, we aim to enhance object and scene identification and overcome the challenges presented in this study.
[4]	Pal *et al.*, [2021]	Deep learning	weakly supervised object detection (WSOD) algorithms	MS COCO dataset and PASCAL VOC 12 dataset	Additionally, ANN-based machine learning models are referred to as "black-box" models, as even their creators may not be able to explain how the AI reached a particular conclusion.
[5]	Pramanik *et al.*, [2022]	RCNN	MCDSORT algorithm	MCD-SORT, SOP, AMIR15, and AM over the dataset PETS	As a potential future research direction for the suggested G-RCNN, among other applications, the current use of deep CNNs for change detection [31] in SAR images may be explored.
[6]	NASEER *et al.*, [2024]	GMM Segmentation Feature Fusion Approach	GMM and MEAN SHIFT SEGMENTATION algorithm (Multilayer Perceptron)	MSRC and Corel 10k datasets.	Our subsequent work will focus on enhancing scene recognition and object identification. We are focusing on refining algorithms to accurately and sensitively analyze scenes, optimizing them for real-time use, and adapting to diverse environments.

[9]	Mohandoss and Rangaraj [2024]	Deep learning	LuN*et al*gorithms	MOT20 dataset.	The suggested approach and maps for vehicle identification and categorization will be updated and improved. Our proposed architecture will also be tested for additional object detection uses, such as identifying unusual activity.
[13]	Li *et al.*, [2021]	R-CNN model	Soft-NMS algorithm	KITTI datasets	In the future, we'll investigate the most effective Generative Adversarial Network (GAN) model to address item identification and recognition in more intricate traffic situations.
[15]	Premanand* and Kumar [2023]	MRNN and PS-KM Models	Pearson Similarity-centred Kuhn-Munkres (PS-KM) algorithm	MOT dataset	The suggested approach will be further enhanced by employing more advanced tracking techniques for monitoring occluded objects in complex environments.
[18]	Mauri *et al.*, [2021]	Deep Learning	Not specified	KITTI's road dataset	In addition to conducting specific tests on an NVIDIA Jetson TX2 embedded system devoted to real-time artificial intelligence applications, we will close the accuracy gap between our methodology and the state-of-the-art techniques.
[25]	Wang *et al.*,	Deep Recurrent Learning	convolutional neural network (CNN) algorithm	ROI datasets	Its drawback is the inability of the suggested RFD model to handle novel object types.

Table 2: Datasets used for object detection in prior works

Dataset	References
VOC2012 dataset	[1][31]
PASCALVOC-12 dataset	[3][20]
MS COCO dataset and PASCAL VOC 12 dataset	[4]
MCD-SORT, SOP, AMIR15, and AM over the dataset PETS	[5]
MSRC and Corel 10k datasets.	[6][26]
MOT20 dataset.	[9]
KITTI datasets	[7][13][18][23]
MOT17 datasets	[15][32]
EDS dataset	[21]
ROI dataset	[25]
Vehicle License Plate Dataset	[28]
(UA-DETRAC) dataset.	[16]
PASCAL, CIFAR 10, IMAGENET, SUN, and MS COCO	[17]
The TUM RGB-D data	[19]

Novak and Radovanović [31] investigated several transfer learning approaches utilizing deep convolutional networks and demonstrated that transfer learning techniques can significantly enhance image classification accuracy, even with inadequate training data. Alagarsamy and Muneeswaran *et al*. [32] showed that the RSOADL–MODT model, which addresses complex situations, outperforms existing methods in multiple-object tracking by leveraging deep learning. Marolt and Korošec [33] presented a hybrid deep learning architecture for real-time object segmentation, striking a balance between semantic accuracy and efficiency suitable for deployment in resource-limited computer vision environments. Afroze *et al*. [34] presented a system that requires eye gaze integration yet achieves excellent accuracy in evaluating the visual center of attention using head positions. Rafique *et al*. [35] presented a robust RGB-D object detection system that performs well in challenging situations and is expected to improve further with the application of deep learning.

Hou *et al*. [36] addressed occlusions and enhanced YOLOv4 for identifying construction machinery, although at a minor speed decrease (97.03% mAP). Ke *et al*. [37] recommended a combined detection and identification motion tracking (MOT) architecture using ConvGRU to enhance video-based detection and tracking performance. Ali *et al*. [38] explored advanced indoor and outdoor object localization and recognition algorithms, reviewing methods and addressing multimodal data fusion for improved accuracy. Wang and Chen [39] proposed a superior 3D object detection technique called ECA Modules-PointPillars, which enhances accuracy through channel attention. However, real-time performance and pedestrian detection still require improvement. Padmaja *et al*. [40] presented a YOLOv3-based model that learns slowly on large datasets and achieves outstanding real-time human action recognition accuracy. Rani *et al*. [41] compared YOLOv5 with Faster R-CNN for the classification of solid waste. It finds that YOLOv5 achieves high accuracy and recommends lightweight networks for future use. The foundational models alluded to are highly effective in applications such as object detection and classification tasks. Redmon and Farhadi developed YOLOv3 [43], a real-time object detection architecture that focuses on speed and accuracy by redefining object detection as a simple regression problem. Ronneberger et al. [44] introduced U-Net, a robust architecture for biomedical image segmentation that combines an asymmetric encoder and a symmetric decoder, incorporating skip connections, which achieves excellent performance in pixel-level localization. Simonyan and Zisserman [45] introduced VGGNet, which focused on a deep yet simple network (stacking small convolutional filters) and had a significant impact on the following CNN architectures. LeCun et al. [46] paved the way for modern CNNs with LeNet, demonstrating the application of deep learning in document recognition. Together, these models serve as the foundation for benchmarking and downstream deep learning research based on contemporary systems. Table 1 presents a summary of the literature findings, while Table 2 provides details of the datasets used in prior works. Table 3 compares selected related methods, including the data used, the accuracy obtained, and their limitations. It demonstrates how the proposed EL-MODC algorithm effectively addresses the problems associated with accurate object boundary detection, large-scale changes, and real-time scalability, outperforming current deep learning methods for object detection.

Table 3: Comparative Summary of Existing Methods and Proposed EL-MODC Framework

Ref	**Method**	**Dataset**	**Accuracy (%)**	**Key Limitations Addressed by EL-MODC**
[1]	CNN + SVM (Firefly Optimization)	VOC2012	~90	Limited segmentation ability, weak on boundary accuracy
[2]	YOLO-ResNet Hybrid	ICARM	75.36	Poor feature hierarchy and low generalization
[3]	UNet + DWT + AlexNet	PASCALVOC-12	Not reported	High complexity, lacks real-time capability
[5]	G-RCNN + MCD-SORT	PETS	High recall	Not evaluated for diverse object scales
[9]	LuNet + YOLOv2	MOT20	94.00	Underperforms on occlusion and overlapping objects
[13]	Faster R-CNN + Soft-NMS	KITTI	86.78	Fails in dense traffic conditions
—	EL-MODC (Proposed)	VOCdevkit	96.40	Accurate segmentation, robust feature extraction, scalable

The reviewed studies emphasize advancements in segmentation, feature extraction, and object detection techniques, including SegNet, YOLO variants, and transfer learning models such as ResNet. Challenges such as handling complex scenarios, optimizing accuracy, and ensuring real-time processing persist, motivating the development of the enhanced multi-object detection and classification framework presented in this paper, which utilizes SegNet and transfer learning.

3 Proposed framework

Figure 1 illustrates the design of the proposed framework, which addresses the aforementioned challenges in multi-object detection and categorization in realistic environments, taking into account the object's scale, background, and real-time processing requirements. Current techniques lack both accuracy and computational efficiency. To resolve these issues, the framework proposes utilizing SegNet for segmentation, leveraging its encoder-decoder architecture to separate different object boundaries accurately. Additionally, ResNet50 is employed with transfer learning for robust feature extraction and multi-class classification, utilizing pre-trained weights to reduce the time required for model training and enhance overall performance. These design choices ensure the detection framework is highly scalable, precise, and efficient, making it suitable for several object detection and categorization methods task applications.

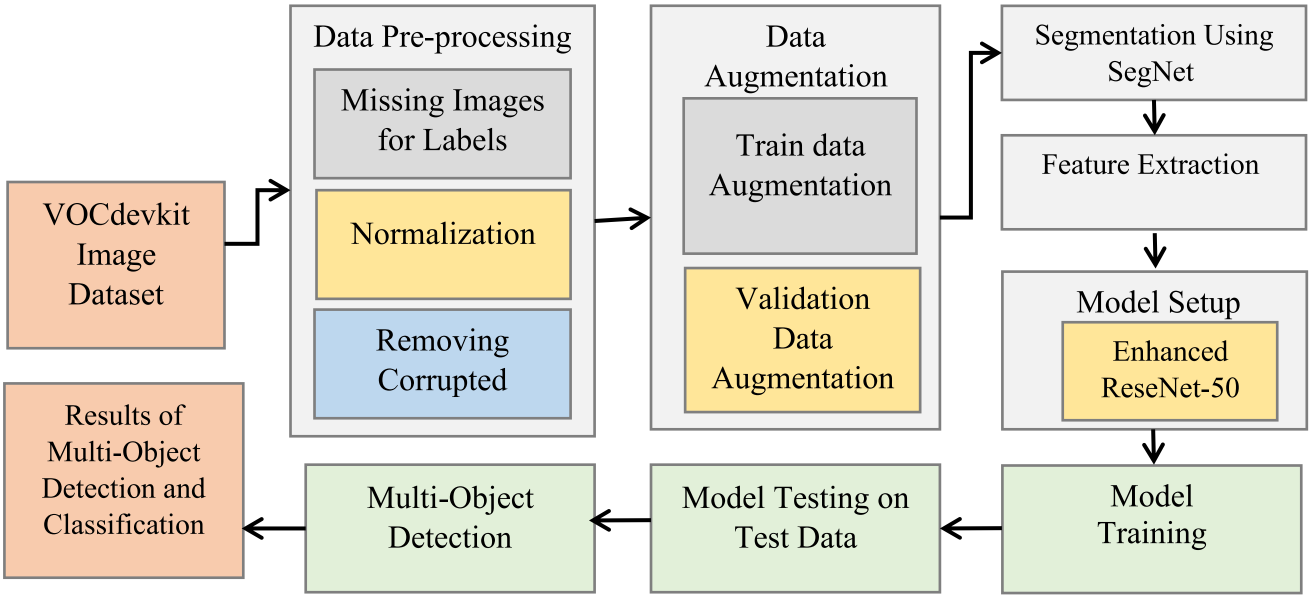


Figure 1: Proposed enhanced deep learning framework for multi-object detection and classification using SegNet and a pre-trained model with transfer learning

This framework proposes a new deep learning approach that combines segmentation using SegNet with a pre-trained ResNet-50 model for classifying and extracting features of multiple objects in a single image. First, the component uses the VOCdevkit image dataset. This data is then subjected to a detailed data preprocessing phase, wherein several pivotal problems, such as missing images against labels, data normalization, and corrupted images, are addressed. This process removes useless photos and other issues, thereby maintaining the quality of the input data. First, the framework preprocesses the data and then enriches training and validation datasets by applying data augmentation. This step is crucial for enriching the variety of information and enhancing the model's robustness against variations in object classification, including orientation, scale, and differences in illumination. Next, the augmented datasets proceed to the segmentation stage, where SegNet accurately separates the boundaries of each object. The encoder-decoder structure that SegNet utilizes is well-suited for this application, and therefore enables pixel-level segmentation, which is essential for closely following features. Then, the extracted features are fed into the improved ResNet-50 model, leveraging the advantages of transfer learning to accelerate training and improve performance. This model is pre-trained and fine-tuned for your specific multi-object detection and classification. The model parameters are iteratively refined based on the training data, while the test dataset is used to evaluate the model's generalization capability. Finally, all framework components will be integrated to identify and categorize many objects in real-time. The results highlight the success of this new approach, which achieves object-level grouping, even in complex visual representations. The framework is designed for accuracy, efficiency, scalability, and compatibility with various types of applications. This automatically addresses problems associated with multi-object detection and classification tasks.

3.1 Data Pre-processing

Initial data processing was implemented to provide high-quality inputs to the proposed framework while increasing the reliability and accuracy of the model. Image Dataset: The VOCdevkit image dataset was initially used, employing a systematic process to handle missing or inconsistent data. To maintain consistency between the data and the labels, we filtered out images that lacked corresponding labels. Additionally, a normalization method was applied to all pixel values to ensure that all images had a similar range of values. This process created standardization and removed biases that would arise from using different photos with different brightness and contrast. These included corrupted and incomplete images, which could have negatively affected the training. These steps significantly reduced noise and improved data quality, yielding a uniform and complete dataset for further processing. These pre-processing steps enabled the framework to effectively utilize data augmentation and segmentation, thereby boosting its confidence levels in scenes involving multiple objects that require detection and classification.

The data augmentation stage follows directly after the pre-processing operations. Preprocessing: All input images and their corresponding segmentation masks are resized to the same dimension and normalized to the intensity range of [0, 1]. These manipulated image-mask pairs then serve as input for the augmentation pipeline. Each augmentation operation (rotation, flip, scale, and noise) is performed simultaneously on the image and its corresponding segmentation mask to maintain pixel-wise registration. This helps ensure that the semantic consistency between image features and their labels is preserved throughout the pipeline. The way the method chains preprocessing and augmentation in most cases ensures consistent and realistic training samples, leading to better generalization of the model.

3.2 Data augmentation

A full range of data augmentation was also implemented during training to enhance model robustness and generalize the model across various input conditions. Key to Detection: All these transformations were selected due to their applicability to the complications faced in multi-Object detection in the real world, such as areas with varying lighting, object orientation, and scale.

Augmentation began with random rotations, where the images were rotated from −15 to +15 degrees. This serves to mimic small rotations, often encountered under natural conditions, and allows the model to learn rotation invariance. After this, both horizontal and vertical flipping were independently performed with a 50% probability. These flips prevent the model from learning any biases concerning object orientation, which is particularly useful in domains such as surveillance or autonomous navigation, where objects can appear from any direction. Scaling operations were used to simulate zooming and changes in distance. One hundred twenty percent is then sampled from the Uniform ([0.9, 1.1]) – i.e., 90% to 110% – to provide the scaling applied to each image that we randomly scale, preserving the aspect ratio. Spatial changes were combined with intensity adjustments. To avoid a uniform stimulus, a brightness range of ±20% and a contrast change of ±15% were used. These differences simulate diverse lighting conditions and sensor configurations, thereby enhancing the model's robustness to illumination variations.

In addition, Gaussian noise was added to the images to simulate existing factual errors. The noise was a normal distribution with a mean of zero and a standard deviation of 0.01, mimicking minor sensor noise or image artifacts. This requires the model to employ a deep and fine-grained encoding scheme, such that it can discard patterns that appear in a textual corpus but don’t represent linguistic variation. All of these augmentations were performed only on the training set, ensuring that the validation and test evaluations accurately describe the model's generalization to real data. We selected a combination of spatial and intensity augmentations by tuning in isolation, with a focus on enhancing performance without overfitting the model to specific perturbations.

The augmentation we applied included rotation (±15°), horizontal/vertical flip, scale transformation (90–110%), variation of brightness (±20%), and contrast adjustment to simulate the real-world variability in the appearance of objects, as a result of change in viewpoint, lighting, and scale. These transformations enhance the model’s ability to generalize under various conditions. To make the trained model robust to real sensing data, we also employed noise injection with Gaussian noise (mean = 0, variance = 0.01) to simulate sensor imperfection. We do not include advanced augmentation approaches (such as CutMix and MixUp) as it is essential to preserve semantic consistency in multi-object scenarios, which is crucial for accurate segmentation.

A comparison was made by training the model with (i) preprocessed data and (ii) augmented data. The results have shown an overall performance improvement, as all indicators achieved their best results to date. The F1-score increased from 93.62% to 96.16%, and accuracy improved from 93.90% to 96.40%. This verifies that the augmentation techniques can improve the learning ability and robustness of the designed EL-MODC model.

3.3 Image segmentation

SegNet, a convolutional encoder-decoder architecture (Figure 2), is used in the proposed framework to achieve high-resolution image segmentation. It takes input images and applies an encoder that progressively extracts hierarchical feature representations using convolutional layers, batch normalization, and rectified linear unit (ReLU) activations. The pooling layers capture the spatial hierarchies, and the indices are stored for perfect decoder reconstruction. These are used for upsampling in the decoder to restore the spatial dimensions of the segmented image. Lastly, a softmax layer is applied, yielding pixel-level class probabilities that result in the segmentation of the image into object classes.

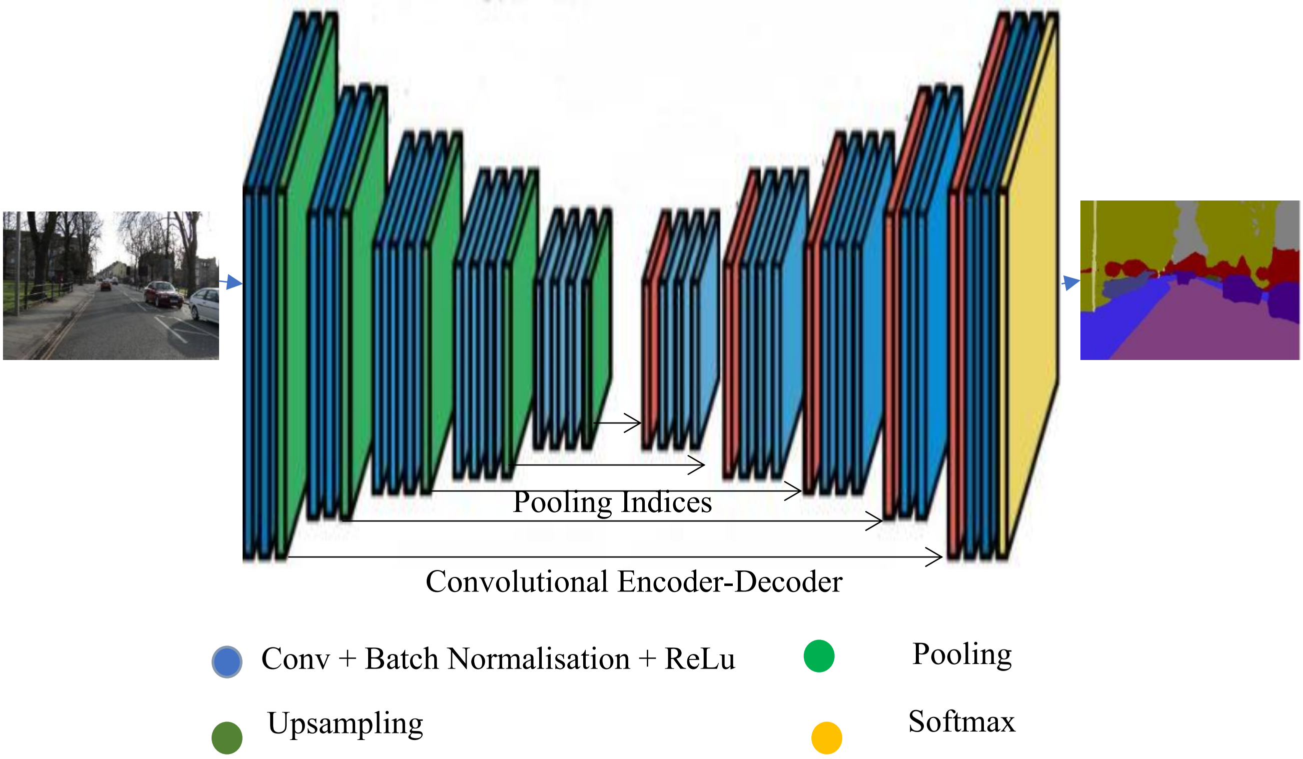

Pooling Indices
Convolutional Encoder-Decoder
Conv + Batch Normalisation + ReLu
Pooling
Upsampling
Softmax

Figure 2: Architecture of the SegNet Model Used for Pixel-Wise Object Segmentation

A critical component of the suggested system is segmentation. It is used to distinguish between multiple objects within an image, enabling multi-object detection and classification, while focusing on specific features. Using SegNet also provides pixel-level accuracy in a typical deployment where sensors providing direct depth measurements are not available, making this approach suitable for even challenging environments where objects may overlap or be occluded (partially/somewhat. This facilitates the following processing steps, such as feature extraction using ResNet50, ensuring that the extracted features correspond to meaningful and distinct areas. In this context, segmentation also improves accuracy and computational efficiency. Segmentation provides a level of detail that enables multi-object detection and classification stages to work in redundancy, achieving the structure's target while eliminating ambiguity regarding the nature of the object and rendering irrelevant background data, thereby working more reliably.

SegNet, as shown in Figure 2, is an encoder-decoder architecture that is well-suited for highly detailed pixel-wise segmentation. The encoder compresses the input features using the convolutional and pooling layers that aggregate high-level semantics. These 10 features are fed to the decoder, which up-samples feature maps in a non-linear way using pooled indices to preserve spatial accuracy. It is especially effective to restore object boundaries because it keeps blur to a minimum, which can occur in naïve interpolation. Finally, a pixel-wise softmax classifier is applied, which assigns a class label to each pixel to achieve fine separation of the object and background. It is this boundary-aware segmentation that is vital for the subsequent classification with the ResNet50 module. The figure is replaced with complete annotations that include all mentioned components.

In this paper, we utilize SegNet as the primary segmentation module in specific configurations tailored to the dataset and object detection task. The encoder was a pre-trained VGG16 network copy, and the decoder was symmetric to the encoder, used to restore the spatial resolution by utilizing max-pooling indices. We trained the network using a batch size of 16, a learning rate of 0.001, and with the Adam optimizer over 50 epochs. With pixel-wise supervision, we employed the cross-entropy loss. The core SegNet architecture has been modified only for the final layer to produce class-wise segmentation mappings to the multi-object dataset. Each decoder layer was followed by a dropout layer to help with overfitting. Such an implementation enables the accurate pixel-level localization of objects, which in turn provides structured input to the downstream classifier and consequently leads to better context learning during training.

3.4 ResNet50 with transfer learning

The ResNet50 architecture we employed in this pipeline is a deep 50-layer convolutional network composed of residual blocks. The architecture begins with a 7×7 convolutional layer with max-pooling and consists of four stages of residual blocks, each including convolutional layers with identity skip connections. These are not skip connections to the original Pyramid (i.e., back to the low-level features, such as in SPP-Net or FPN), but to the skip connections of FCN-8s, allowing the gradients to be propagated through the deeper layers without attenuation, which helps stabilize and accelerate the training.

Earlier in the network, the first convolutional layers and the first few residual blocks compute lower-level features, such as edges, corners, and textures. These are learned using small kernels applied over local receptive fields. The receptive fields are enlarged as we progress through the network, and later layers begin learning more complex and abstract high-level representations, such as object shapes, patterns, and semantic parts.

The answer to the final convolutional block is a 3D tensor (of shape (7, 7, 2048)) on which we can stick a a small FCN, which would then run across our grid and output some scores (one for every class that we have in the training set) for every grid cell. To obtain a 1D vector from this, the model uses a Global Average Pooling (GAP) layer. The average value over each of the 2048 feature maps is computed by the GAP layer, effectively summarizing each feature map into a single scalar. This yields a 1D feature vector of length 2048, which is then fed into the fully connected layer or classification head. Therefore, the conversion from 2D to 1D is performed via the GAP process, not immediately after the convolutions.

This progressive process of learning, from low-level spatial textures to high-level semantic concepts, is an indicator of the effectiveness of ResNet for transfer learning tasks, making it a suitable classifier component for the EL-MODC architecture.

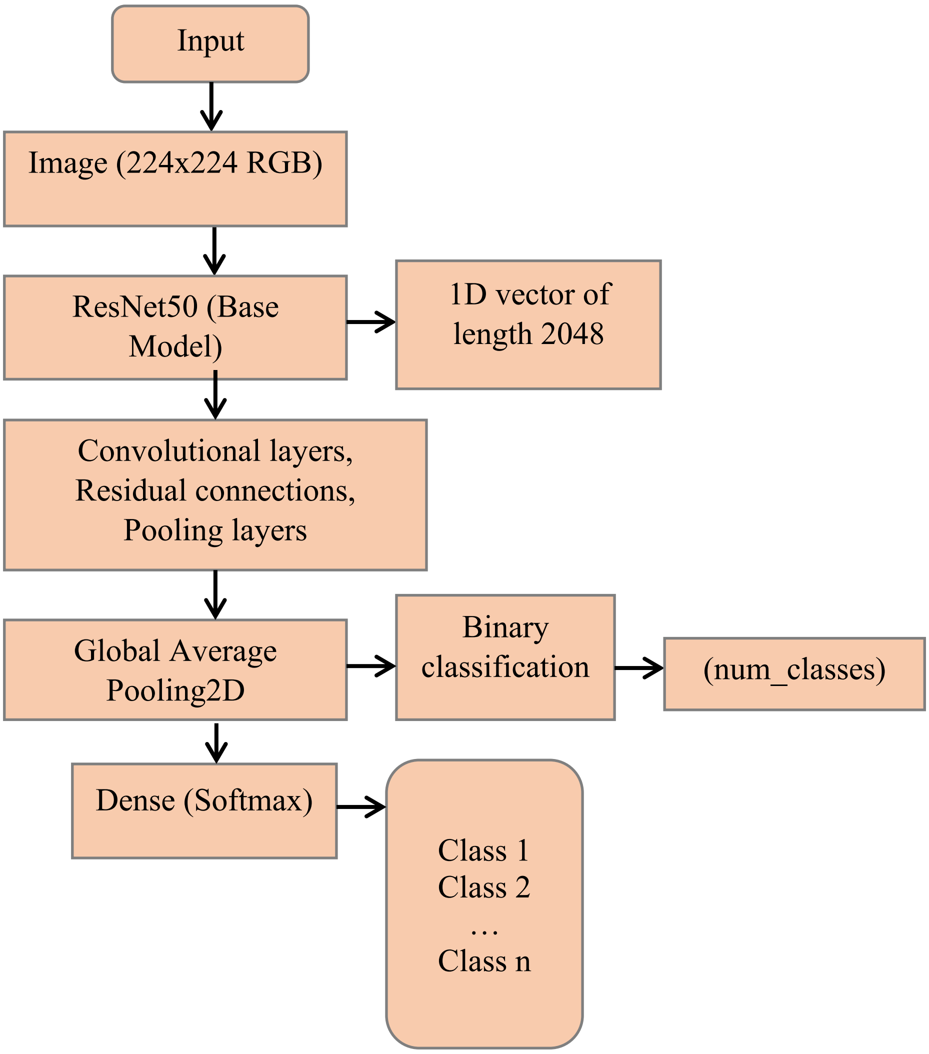


Figure 3: ResNet50 model with transfer learning used for feature extraction and multi-class classification

Figure 3. After extracting the features, a dense layer that is entirely linked and has a softmax function is employed to classify them. It could also be a thick layer to explain how many objects could be there (keep multi-class classification in mind). The softmax function ensures that the total of the output probabilities equals one, allowing the correct class of the object to be predicted. Transfer learning is employed here to enhance the model's functionality and training duration on the target dataset by initializing it from pre-trained ResNet50 weights. Using

Transfer learning, this ResNet50 integration enhances the framework's ability to detect and classify multiple objects. It proposes a comprehensive and practical approach that tackles detection challenges across various environments by extracting rich and robust features to recognize and classify objects. This helps to make the framework accurate & scalable.

The ResNet50 model was adopted through a transfer learning technique with pre-trained weights from ImageNet. The first layers (up to the fourth convolutional block) were frozen to preserve the low-level features learned during training on ImageNet. In contrast, the last convolutional block and the fully connected [email protected] @tabl@sentences were tuned to the target dataset. GAP was applied after the convolutional layers to reduce the dimensionality of the feature maps, followed by a dense layer with ReLU activation and a final output layer (Softmax) for multi-class classification. The Adam optimizer was used for training with a learning rate of 0.0001, a batch size of 32, and categorical cross-entropy loss over 50 epochs. We used dropout at a rate of 0.5 before the final dense layers to prevent overfitting. Such representation-based portions of the architecture enable the model to maintain generalist feature extraction capabilities while learning object-specific features for this classification task.

3.5 Training of the model and detection of multiple objects

In this step, segmented images from SegNet are input into ResNet50 (with transfer learning) for model training. This mechanism ensures that the model focuses on the relevant object regions, thereby improving the accuracy of feature extraction. A well-known deep learning model, ResNet-50, a pre-trained model on ImageNet, is employed to extract features from the segmented images. Residual connections overcome the vanishing gradient problem, thus allowing efficient learning in deeper networks. In multiple layers, features are gradually extracted, and both low-dimensional features and high-level features (e.g., edges, object working modes) are essential for accurate multi-object detection and classification—globally Misleading. After including the feature extraction feature maps, the dimensionality reduction technique used is global average pooling (GAP), which retains the essential spatial information. A final dense layer with softmax activation maps these features into the specific object classes, allowing for multi-class classification. The pre-trained weights of ResNet50 are fine-tuned on the target dataset during training. Transfer learning ensures faster convergence and increased precision when training data is scarce. The framework segments the input image into regions and processes each area separately, enabling YOLOv1 to fully utilize its capabilities, even for multi-object detection in a single image. By integrating SegNet for accurate segmentation and ResNet50 for feature extraction, the framework ensures accurate detection and classification even in diverse and complex scenarios, thereby improving the overall effectiveness and dependability of the system. Here is the essential mathematical model of the framework. Label counting is carried out as follows. For a set of labels $L = \{l_1, l_2, \dots, l_n\}$, the count of each label l_i is done as in Eq. 1.

$$label_{count(l_i)} = \sum_{j=1}^{n} 1\{l_j = l_i\} \quad (1)$$

Randomly select a subset S of $num\_samples$ images from a set of images Images as in Eq. 2.

$$S = random\_sample\ (Images, num\_samples) \quad (2)$$

The size (w_{i}, h_i) of image img_i is expressed as in Eq. 3.

$$(w_{\mathrm{i}}, h_i) = Image.getsize(img_i) \quad (3)$$

The framework checks for missing images, as shownin Eq. 4.

$$M = \{img_i \mid \neg\ os.path.exists(image\_path + img_i)\} \quad (4)$$

The number of missing images is is then expressed as Missing Count = $|M|$.Regarding data preprocessing using rescaling, for each pixel p in an image, the rescaling formula is in Eq. 5.

$$p_{\mathrm{rescaled}} = \frac{p}{255} \quad (5)$$

The loss function is categorical cross-entropy for multi-class classification, found in Equation 6.

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \quad (6)$$

The actual label (one-hot encoded) is denoted by $\hat{y}_i$, the projected probability for class I is denoted by y_i, and C is the number of classes. As seen in Eq. 7, the softmax function transforms logits z_i into probabilities p_i.

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \quad (7)$$

In model training, the optimizer adjusts the model weights w using gradient descent as in Eq. 8.

$$w_{new} = w_{old} - \eta \cdot \nabla_w L \quad (8)$$

Where η is the learning rate and $\nabla_w L$ is the gradient of the loss concerning the weights. Preprocessing was carried out first ito improve the reliability of the data, excluding images without associated annotations and corrupt image entries usingfile existence checks and then file integrity checks. All images were resized to 224 × 224 × 3 pixels, following which they are normalized as described by $I_{norm} = \frac{I_{pixel}}{255}$, to have pixel values that lie within the same range and thereby, gradient propagation will be stable during learning. Model training began with the fine-tuning of the ResNet50 model to the conv4_block1_out layer, with all previous layers frozen (to maintain the extracted features' generalizability). We used the Adam optimizer with a learning rate of 0.0001, β₁

= 0.9, and β_2 = 0.999. We conducted a randomized hyperparameter search over batch size and learning rate, using early stopping (patience = 5) to prevent overfitting. Validation Optimal: We selected the final training settings—batch size 32, learning rate 1e-4, and 30 epochs—on the basis of the best validation F1-score. This architectural choice resulted in a faster convergence rate and an increase in generalization on the VOCdevkit dataset.

Equation 6 is the categorical cross-entropy loss employed during training to minimize incorrect class assignments. In Equation 7, we apply a softmax function to the raw logits of the final layer, converting them into a normalized probability distribution over all object classes. This leads to each image in the model output summing to one, which facilitates more rigorous multi-class prediction. Equations 1 to 5 also represent preprocessing-related logic — label validation, missing image detection, normalization, and dataset sampling — all culminating in high-quality inputs for training. The optimizer computes the explicit weight update based on the gradient, as formalized in Equation 8 . These equations describe the full learning pipeline, from unwrapped data to probabilistic output generation and backpropagation.

To ensure consistency between segmentation and recognition, the masks produced by SegNet are element-wise multiplied by the original input images. This essentially eliminates non-relevant background regions, thereby preserving the areas of interest. The masked images are fed into ResNet-50 to extract and classify the image features. The cascaded pipeline enables the model to concentrate its learning capacity on the most informative regions, resulting in higher accuracy with reduced noise in the classification. The fine-tuned inputs are generated during training for each input image and its associated segmentation mask. This architecture incorporates a pseudo-attention mechanism, enabling ResNet50 to learn object-aware representations without the need for a tiered attention module.

3.6 Proposed algorithm

The proposed method details the entire training and testing process of the EL-MODC framework, which combines segmentation and deep learning. This ensures a consistent flow, from the preparation and augmentation of the dataset to spatial object segmentation and ultimate classification. The non-line-of-sight reconstruction process enhances robustness, accuracy, and interpretability, resulting in reliable multi-object detections in various scenes and complex practical environments.

```
Algorithm 1: EL-MODC Training and Testing Procedure
Input: Dataset $D = \{x_i, y_i\}$, SegNet model, ResNet50 model
Output: Trained EL-MODC model, performance metrics
    1. Split D into $D_{train}$ and $D_{test}$ (80:20 ratio)
    2. Preprocess $D_{train}$: resize, normalize, verify labels
    3. Augment $D_{train}$ to obtain $D'_{train}$
    4. Apply SegNet on $D'_{train}$ for pixel-wise segmentation
    5. Extract features from segmented outputs using pretrained ResNet50
    6. Train classification layers using Softmax and categorical cross-entropy
    7. Test model on $D_{test}$ and compute Accuracy, Precision, Recall, F1-score, IoU
```

Algorithm 1: EL-MODC training and testing procedure

Algorithm 1 first partitions the dataset into two: a training set and a testing set. As is customary, 80% of the data is used for training and the remaining 20% is held for testing. The training data is preprocessed – this includes resizing the images to a uniform input size, normalising the pixels to have values in a similar range, and checking the accuracy of the labels assigned to each image.

After some preprocessing, the data is augmented to enhance the model's robustness and address class imbalance. We augment the data with methods such as horizontal flipping, rotation, brightness changes, and scaling to create more synthetic examples. These augmented images are combined with the original training images to form an augmented training set, which enhances the model's ability to generalize.

Next, a semantic segmentation approach is adopted, and the enhanced training data is distributed throughout the SegNet decoder-encoder network. This network segments object instances at the pixel level and generates spatial masks that highlight discriminant object regions in each image. These segmented images are then processed through a pre-trained, modified ResNet-50 that has been specialized for another complex classification task. This network captures the high-level semantic cues of the segmented regions.

These extracted features are then fed into a classification head, which consists of fully connected layers followed by a Softmax function, to predict the corresponding class for each object region. During training, the model learns a loss function from the difference between the expected and actual classes, making correct predictions closer to 1 and incorrect predictions closer to 0.

At last, the test dataset is applied to evaluate the trained EL-MODC model. Results are quantified based on standard metrics, including accuracy, precision, recall, F1 Score, and Intersection over Union (IoU). These statistics shed light on the model's performance in detecting and classifying various objects in diverse and challenging scenes.

3.7 Dataset details

The VOCdevkit Image Dataset [42] was developed as one of the benchmark datasets for the Pascal Visual Object Classes (VOC) challenge and is widely used as a training and test set for object detection and classification models. This dataset's varied collection of tagged photos encompasses twenty different item categories, including animals, vehicles, and household items. 3.1: Overview of the approaches, datasets, metrics, and annotations in the chosen tasks. The dataset comprises over 11,000 images and 27,000 annotated objects, featuring significant variations in object appearance, scale, and background complexity, thereby providing a benchmark for evaluating the robustness of deep learning (DL) frameworks for multi-object detection.

3.8 Performance evaluation

The performance of the EL-MODC model is evaluated using four standard classification metrics: accuracy, precision, recall, and F1-score, each computed from the confusion matrix. Let TP, FP, FN, and TN denote true positives, false positives, false negatives, and true negatives, respectively. Accuracy measures the overall correctness of the model, as shown in Eq. 9.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (9)$$

Precision (Positive Predictive Value) quantifies the proportion of correctly predicted positive instances among all cases predicted as positive, as shown in Eq. 10.

$$Precision = \frac{TP}{TP+TN} \quad (10)$$

Recall (True Positive Rate) measures the proportion of actual positives correctly identified by the model as in Eq. 11.

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

F1-score is the harmonic mean of precision and recall, balancing both metrics as in Eq. 12.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

In the context of multi-class classification, these metrics are computed per class and then averaged using a weighted or macro-average depending on class distribution. This provides a comprehensive evaluation of the model's ability to detect and classify each object accurately.

4 Experimental results

Experimental Evaluation: We confirm the effectiveness of the recommended Enhanced ResNet-50 model through experimental assessments within constrained settings for accomplishing multi-object detection and classification tasks. Experiments were conducted in Python and TensorFlow on a system equipped with an NVIDIA GPU. Moreover, it was established that the proposed CNN with SNN outperforms the UNet, Baseline CNN, and LeNet, which are regarded as state-of-the-art models. The comparison models were selected based on their widespread use in object detection and categorization tasks. The VOCdevkit dataset for evaluation provided diverse and annotated images for benchmarking. To thoroughly compare the models, we employed the performance indicators of Accuracy, Precision, Recall, and F1-Score. The outcomes indicated that the Enhanced ResNet-50 model outperformed the rest and achieved the highest accuracy among different approaches. These findings confirm that the proposed framework is robust and reliable.

Figure 4: Sample input images from the VOCdevkit dataset illustrating object diversity, scale variations, and multi-object scenes used for training and evaluation

Figure 4 appears to be a collection of five images, each with a filename starting with "2010" or "2011" followed by a number. The images are likely screenshots or photos related to a research project or experiment. Without more context, it's difficult to determine the exact subject matter of the images.

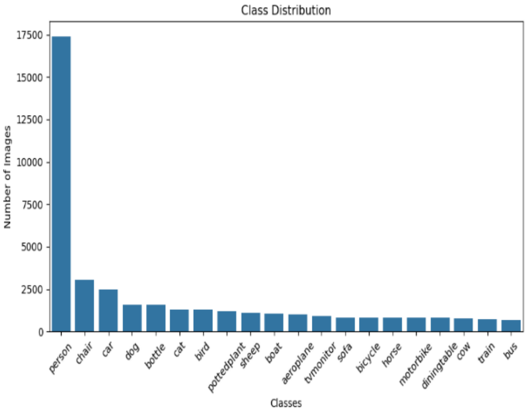


Figure 5: Distribution of class labels in the VOCdevkit dataset. The most common classes include "person," "chair," and "car," which may influence class-wise prediction bias during training.

Figure 5 displays a bar graph titled "Class Distribution". The x-axis represents various classes or categories, such as "person," "chair," "car," "dog," and so on. The number of photos linked to each class is indicated on the y-axis. Each bar's height indicates the frequency with which pictures in a particular class occur, providing the graph with a visual representation of the distribution of images across classes. In most of the pictures, the class "person" is followed by "chair" and "car."

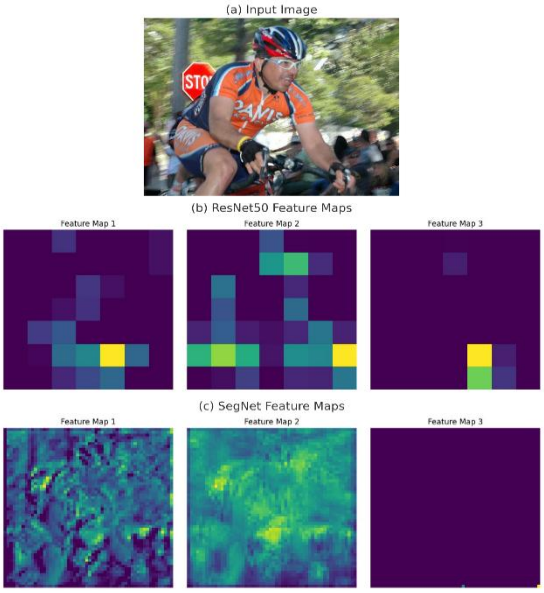
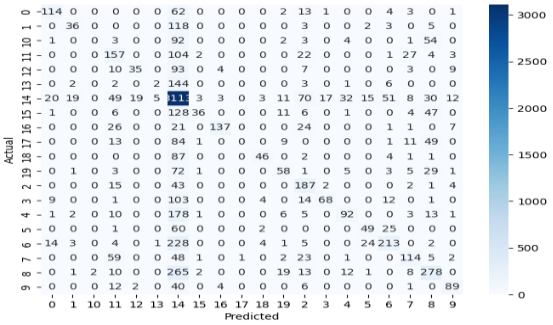


Figure 6: (a) Input image. (b) ResNet50 feature maps showing semantic abstraction. (c) SegNet decoder maps highlighting object boundaries

Figure 6 illustrates the processing flow of the EL-MODC framework. Feature maps of the ResNet-50 encode semantic clues important for classification, while the decoder maps of the SegNet capture fine-grained object boundaries. They jointly demonstrate how the integration of segmentation and deep feature learning enhances the system's performance for both precision and efficient detection and classification of multiple objects.



Actual \ Predicted	0	1	10	11	12	13	14	15	16	17	18	19	2	3	4	5	6	7	8	9
0	114	0	0	0	0	0	62	0	0	0	0	2	13	1	0	0	4	3	0	1
1	0	36	0	0	0	0	118	0	0	0	0	0	3	0	0	2	3	0	5	0
10	1	0	0	3	0	0	92	0	0	0	0	2	3	0	4	0	0	1	54	0
11	0	0	0	157	0	0	104	2	0	0	0	0	22	0	0	0	1	27	4	3
12	0	0	0	10	35	0	93	0	4	0	0	0	7	0	0	0	0	3	0	9
13	0	2	0	2	0	2	144	0	0	0	0	0	3	0	1	0	6	0	0	0
14	20	19	0	49	19	5	3113	3	3	0	3	11	70	17	32	15	51	8	30	12
15	1	0	0	6	0	0	128	36	0	0	0	11	6	0	1	0	0	4	47	0
16	0	0	0	26	0	0	21	0	137	0	0	0	24	0	0	0	1	1	0	7
17	0	0	0	13	0	0	84	1	0	0	0	9	0	0	0	0	1	11	49	0
18	0	0	0	0	0	0	87	0	0	0	46	0	2	0	0	0	4	1	1	0
19	0	1	0	3	0	0	72	1	0	0	0	58	1	0	5	0	3	5	29	1
2	0	0	0	15	0	0	43	0	0	0	0	0	187	2	0	0	0	2	1	4
3	9	0	0	1	0	0	103	0	0	0	4	0	14	68	0	0	12	0	1	0
4	1	2	0	10	0	0	178	1	0	0	0	6	5	0	92	0	0	3	13	1
5	0	0	0	1	0	0	60	0	0	0	2	0	0	0	0	49	25	0	0	0
6	14	3	0	4	0	1	228	0	0	0	4	1	5	0	0	24	213	0	2	0
7	0	0	0	59	0	0	48	1	0	1	0	2	23	0	1	0	0	114	5	2
8	0	1	2	10	0	0	265	2	0	0	0	19	13	0	12	1	0	8	278	0
9	0	0	0	12	2	0	40	0	4	0	0	0	6	0	0	0	0	1	0	89

Figure 7: Confusion matrix of predicted vs. actual classes. High misclassification is observed between visually similar categories, such as "bottle" and "chair." Darker off-diagonal regions indicate frequent confusions

The confusion matrix of the EL-MODC model, per object category, is shown in Figure 7. The darker off-diagonal parts represent correctly classified points, and the off-diagonal darker points represent misclassifications. It is also interesting to note that Classes 1 (e.g., 'Car') and Class 2 (e.g., 'Van') are more likely to be misclassified, suggesting that the model struggles to distinguish similar object categories within the image sub-region. This ambiguity may be due to standard geometric features, such as shape, contour, and overlapping boundary regions in the images, especially in cases of illumination degradation or partial occlusion. Furthermore, the difference in the number of training examples for these classes could explain such confusion. This underscores the need to refine in this direction, for example, by improving the attention mechanism or by providing more discriminant-specific features for each class.

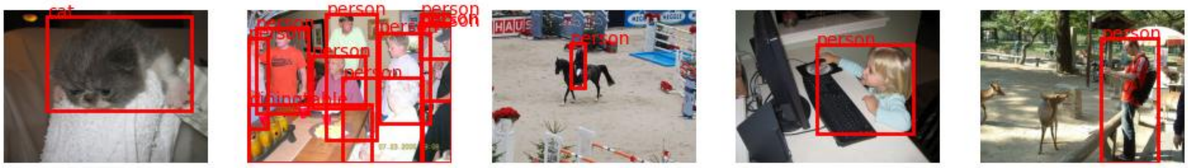


Figure 8: Qualitative results showing model predictions on five test images. Successful detections (e.g., cat, horse, person) are annotated with green bounding boxes.

Figure 8 shows object detection results generated by the EL-MODC framework on various real-world test examples. Although the model generally localizes and classifies objects accurately, several errors can reveal the perceptual difficulties the model may encounter. In the second image, the model predicts multiple boxes, including those for segments of the background with no objects. This implies an over-segmentation tendency, which may be caused by high texture density or background confusion, regarded as object characteristics.

The bootstrapped model failed to detect the face in the fourth image, which may be attributed to poor lighting conditions, partial occlusion, or an insufficient number of representative examples for such a case in the training set. These mistakes indicate that the model's ability to detect objects under low-light or complex backgrounds is limited. Besides, due to false positives and missed detections, more context-awareness or spatial attention models are required to alleviate the uncertainty in ambiguous regions.

These observations further justify the need for future efforts to improve robustness through methods such as domain-specific data augmentation, adaptive thresholding, and attention-based refinement layers, which can suppress spurious detections while effectively capturing important object regions.

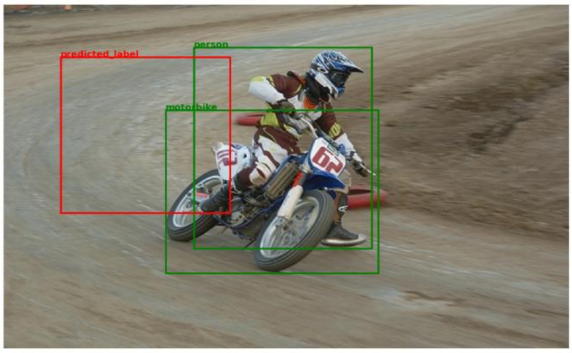


Figure 9: Output visualization on a sample image containing both a motorcycle and a person

Figure 9 illustrates a successful example of multi-object detection using the EL-MODC framework, where all related objects are accurately detected and located within their corresponding bounding boxes. Unlike a few earlier cases, the test set used in this example contains well-illuminated, high-resolution images with clearly defined object boundaries, while minimizing background clutter, which enables the model to perform well, as supported by the predictions. The segmentation module was able to effectively segment regions of objects due to the uniformity and separation of colors within objects, which in turn reduced ambiguity in the feature extraction process by the ResNet50 classifier.

This result demonstrates the model's capacity under ideal imaging conditions and, given that both noise, occlusion, and inter-class overlap are minimized, reinforces that its performance is very high within its detection pipeline. However, this is also a drawback: the model relies on clean, high-quality input data. Although this case demonstrates the potential of the model in structured settings, it highlights the necessity to continue tuning robustness for complex and unstructured scenes through domain adaptation or an attention mechanism in feature space.

Table 4: Performance comparison of object detection and classification models

Object Detection Model	**Precision**	**Recall**	**F1-score**	**Accuracy**
UNet	89.12	89.61	89.36	89.52
Baseline CNN	91.17	91.26	91.21	91.58
LeNet	95.21	95.17	95.19	95.07
Enhanced ReseNet-50	96.02	96.32	96.16	96.4

In this study, the proposed EL-MODC model is compared with baseline deep learning models, including UNet, LeNet, and a standard Convolutional Neural Network (CNN). Although UNet is typically used for segmentation, its output was combined with a classifier for fair evaluation in the multi-object detection and classification context. Table 4 presents the performance metrics—accuracy, precision, recall, and F1-score—for all models. The EL-MODC framework achieves the highest values across all metrics, demonstrating superior capability in both localization and classification.

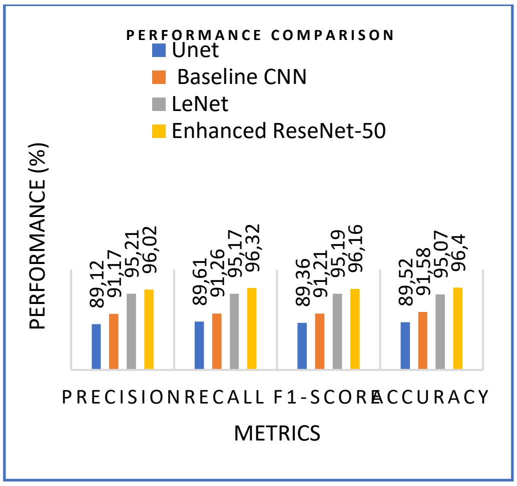


Figure 10: Performance comparison of object detection and classification models across accuracy, precision, recall, and F1-score

As can also be seen in Figure 10, the Enhanced ResNet-50 model outperforms the other three models in all three metrics. What stands out is that it has all the highest values for Precision, Recall, F1-Score, and Accuracy. Out of the four models, the Unet model performs the absolute worst, with the lowest value for all metrics. The Baseline CNN and the LeNet models have comparable performances, but the Baseline CNN is marginally better than LeNet in Precision and F1-Score. In conclusion, the Enhanced ResNet-50 demonstrates superior performance.

Table 5: Comparative evaluation with state-of-the-art models for multi-object detection and classification

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Reference
YOLOv3	92.30	91.20	90.80	91.00	[43]
UNet + CNN	93.10	92.00	91.60	91.80	[44]
Baseline CNN	91.00	89.50	90.00	89.75	[45]
LeNet	95.07	95.01	94.91	95.19	[46]
EL-MODC (Proposed)	**96.40**	**96.02**	**96.32**	**96.16**	*This Study*

A comparison of the proposed EL-MODC model with several well-established state-of-the-art models for multi-object detection and classification is tabulated in Table 5. We observed that the proposed EL-MODC outperforms state-of-the-art models, such as YOLOv3, UNet, Baseline CNN, and LeNet, in most metrics, including accuracy, precision, recall, and F1-score. This demonstrates EL-MODC's superior performance in complex object detection cases, particularly in terms of reliability and generality.

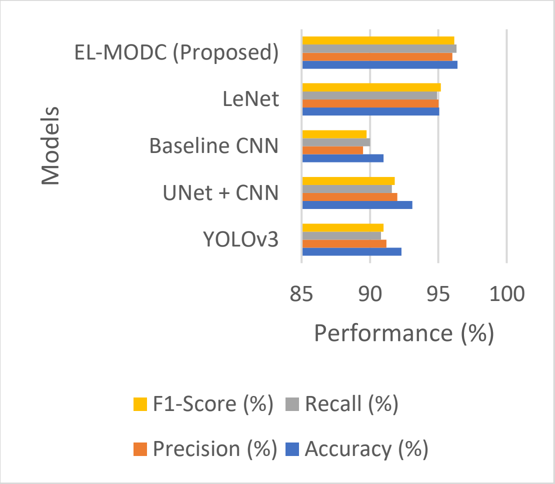


Figure 11: Performance Comparison of EL-MODC with State-of-the-Art Models

Figure 11 provides a visual representation and serves as a solid foundation for comparing the performance in terms of preserved efficiency and robustness with the proposed EL-MODC model and benchmark state-of-the-art (SOTA) models: YOLOv3, UNet, Baseline CNN, and LeNet. Evidently, in each of the metrics presented, EL-MODC demonstrates the highest values for model robustness and efficiency when conducting multi-object detection and classification. At the same time, the consistent level of spread presented within each of the object variations signifies the model's no less strong discriminative and generalization potential.

A 5-fold strategy was used in the experiment to ensure the robustness and effectiveness of classification. In every fold, the dataset was divided into 80% training and 20% testing sets, and the splits were rotated. We then averaged all the values of performance measures over the folds. A paired t-test was performed between the EL-MODC model and its best-performing baseline (YOLOv3) over all folds to determine statistical significance. The findings were all significant (p-value < 0.01 for accuracy, precision, recall, and F1-score), which means that the improvement was statistically significant. Analysis scores for the metrics varied, and 95% confidence intervals (CI) were also calculated. EL-MODC had an accuracy range of [95.97, 96.84%], demonstrating consistency.

5 Discussion

The experimental results verify that the EL-MODC framework is practical, as it outperforms other state-of-the-art models, such as UNet, BaselineCNN, and LeNet, for all performance evaluation metrics. When compared with state-of-the-art methods (listed in Table 3), EL-

MODC exhibits a significantly better accuracy of 96.40%, as well as improved precision (96.02%), recall (96.32%), and F1-score (96.16%). These better segmentations are primarily obtained by combining the pixel-wise boundary-reserving ability of the encoder-decoder structure in SegNet with the deep feature extraction power provided by the deep residual learning in ResNet50, along with efficient training through transfer learning.

In comparison, models such as YOLO-ResNet and CNN-SVM variants struggle to segment at high resolution or exhibit limited generalization capabilities on object scales and in complex backgrounds. SegNet also enables robust object boundary description in the presence of occlusion or cluttered scenes. Additionally, the pre-trained layers of ResNet50 are designed to capture hierarchical abstraction, which is beneficial for the classifier to generalize to unseen data. The extensive data augmentation pipeline also significantly contributed to reducing overfitting and intra-class variance.

Any performance gap among models could come due to dataset bias or the complexity of the test samples as well. For instance, models learned from static features only can fail to recognize overlapped or low-resolution objects. Furthermore, baseline models may be far from being deep or may not cover the full range of input entropies for confident classification. Together with accurate segmentation, efficient feature learning, and an end-to-end pipeline, EL-MODC exhibits great generalization ability and is computationally efficient, making it suitable for scalable, real-time object detection systems.

Although the current design of the network was with SegNet and ResNet50 in mind, as they have been proven to be effective in segmentation and hierarchical feature extraction, we acknowledge that other architectures, such as EfficientNet, MobileNet, and DeepLabv3+, can perform more efficiently and accurately depending on the particular setting. Moreover, an ablation study comparing UNet or PSPNet with SegNet is also planned. In parallel, we aim to survey ResNet50 versus light backbones to quantify the component-level contributions. In the future, we will incorporate noisy and occluded samples to assess robustness under realistic deployment conditions.

5.1 Limitations

This study has several drawbacks. 7. First, the performance of the proposed framework is only evaluated on the VOCdevkit dataset [37], which may not accurately represent real-world diversity. Second, although transfer learning improves learning speed, ResNet50 pre-trained criteria rely on taught attributes that may be less useful for a specific area that requires particular functions. Thirdly, their computational needs, even after various optimizations for performance, may limit their applicability in low-resource environments or real-time systems with strict latency constraints. Sustainable Network Architecture framework limitations include data and domain specificity, as well as computational efficiency, given the limitations in computational availability, which should remain the subject of future research.

6 Conclusion and future work

This study proposes a deep learning approach in the form of an Enhanced Learning-based Multi-Object Detection and Classification (EL-MODC) algorithm, which combines SegNet for the segmentation stage and the ResNet50 model with transfer learning for the hierarchically organized feature extraction and classification stage. The experimental results on the VOCdevkit dataset showed that this approach is superior to other models, achieving 96.40% accuracy and outperforming state-of-the-art models in Precision, Recall, and F1-Score. The modular nature of your framework enables robust, scalable, and adaptable implementations of various applications, making it a significant contribution to computer vision. While the framework does many things well, some aspects do not. The VOCdevkit dataset limits the model's applicability to various real-life situations. At the same time, the reliance on pre-trained ResNet50 weights hinders the model's functionality in terms of the domain norms on which it was retrained. Due to the architecture of SegNet and ResNet-50, they also have high computational requirements, which prevent deployment on resource-constrained systems.

Several strategic improvements are suggested to maximize the capabilities of the EL-MODC framework in practical applications. Increased methodological transparency can be achieved by specifying parameter settings, dataset instances, and experimental conditions. Additional testing on various datasets could facilitate the generalization of performance and demonstrate the robustness of this approach. Moreover, model compression methods (e.g., pruning or quantization techniques) advanced in the field to generate efficient EL-MODC would also make it possible to deploy on low-computational-power devices and meet runtime requirements without a significant performance drop. These lines guide future works towards interpretability, domain transfer, and scalable deployment of multi-object detection systems.

References

[1] P. Kuppusamy; Che-Lun Hung; (2021). Enriching the Multi-Object Detection using Convolutional Neural Network in Macro-Image. 2021 International Conference on Computer Communication and Informatics (ICCCI), (), – . doi:10.1109/iccci50826.2021.9402565

[2] Lu, Zhenyu; Lu, Jia; Ge, Quanbo; Zhan, Tianming (2019). [IEEE 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM) - Toyonaka, Japan (2019.7.3-2019.7.5)] 2019 IEEE 4th International

Conference on Advanced Robotics and Mechatronics (ICARM) - Multi-object Detection Method based on YOLO and ResNet Hybrid Networks., (), 827–832. doi:10.1109/ICARM.2019.8833671

[3] Abdulwahab Alazeb1 , Bisma Riaz Chughtai2 , Naif Al Mudawi1 , Yahya AlQahtani3 ,. (2024). Remote intelligent perception system for multi-object detection. Frontiers. .(.), pp.1-23.

[4] Sankar K. Pal; Anima Pramanik; J. Maiti; Pabitra Mitra; (2021). Deep learning in multi-object detection and tracking: state of the art. Applied Intelligence, pp.1 –30. doi:10.1007/s10489-021-02293-7

[5] Anima Pramanik; Sankar K. Pal;J. Maiti;Pabitra Mitra; (2022). Granulated RCNN and Multi-Class Deep SORT for Multi-Object Detection and Tracking. IEEE Transactions on Emerging Topics in Computational Intelligence, pp.1–11. doi:10.1109/tetci.2020.3041019

[6] Aysha Naseer1 , Hamdan A. Alzahrani , Nouf Abdullah Almujally , Khaled Al Nowa. (2024). Efficient Multi-Object Recognition Using GMM Segmentation Feature Fusion Approach. IEEE. 12(.), pp.1-14.

[7] Fink, M., Liu, Y., Engstle, A., & Schneider, S.-A. (2019). Deep Learning-Based Multi-scale Multi-object Detection and Classification for Autonomous Driving. Fahrerassistenzsysteme 2018, 233–242. doi:10.1007/978-3-658-23751-6_20

[8] Mhalla, Ala; Chateau, Thierry; Gazzah, Sami; Essoukri Ben Amara, Najoua (2018). An Embedded Computer-Vision System for Multi-Object Detection in Traffic Surveillance. IEEE Transactions on Intelligent Transportation Systems, (), 1–13. doi:10.1109/TITS.2018.2876614

[9] T. Mohandoss and J. Rangaraj. (2024). Multi-Object Detection using Enhanced YOLOv2 and LuN*et al*gorithms in Surveillance Videos. e-Prime - Advances in Electrical Engineering, Electronics and Energy. 8(.), pp.1-12.

[10] Elhoseny, M. (2019). Multi-object Detection and Tracking (MODT) Machine Learning Model for Real-Time Video Surveillance Systems. Circuits, Systems, and Signal Processing, pp.1–10. doi:10.1007/s00034-019-01234-7

[11] Ahn, Hyochang; Cho, Han-Jin (2019). Research of multi-object detection and tracking using machine learning based on knowledge for video surveillance system. Personal and Ubiquitous Computing, pp.1–10. doi:10.1007/s00779-019-01296-z

[12] Minghui Yuan, Quansheng Zhang, Yinwei Li * , Yunhao Yan and Yiming Zhu. (2021). A Suspicious Multi-Object Detection and Recognition Method for Millimeter Wave SAR Security Inspection Images Based on M. MDPI. .(.), pp.1-18.

[13] Cui-jin Li; Zhong Qu; Sheng-ye Wang; Ling Liu; (2021). A method of cross-layer fusion multi-object detection and recognition based on improved faster R-CNN model in complex traffic environment. Pattern Recognition Letters, pp.1–8. doi: 10.1016/j.patrec.2021.02.003

[14] Ratheesh Ravindran; Michael J. Santora; Mohsin M. Jamali; (2021). Multi-Object Detection and Tracking, Based on DNN, for Autonomous Vehicles: A Review. IEEE Sensors Journal, pp.1–10. doi:10.1109/jsen.2020.3041615

[15] V. Premanand* and Dhananjay Kumar. (2023). Moving Multi-Object Detection and Tracking Using MRNN and PS-KM Models. Computer Systems Science & Engineering. 44(2), pp.1-15.

[16] Wen, Longyin; Du, Dawei; Cai, Zhaowei; Lei, Zhen; Chang, Ming-Ching; Qi, Honggang; Lim, Jongwoo; Yang, Ming-Hsuan; Lyu, Siwei (2020). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. Computer Vision and Image Understanding, 193, pp.1–20. doi: 10.1016/j.cviu.2020.102907

[17] Štruc, V., Peer, P., & Mihelić, F. (2021). A deep residual framework for object classification and localization. Informatica, 45(3), 405–414.

[18] Antoine Mauri, Redouane Khemmar, Benoit Decoux and Madjid Haddad. (2021). Real-Time 3D Multi-Object Detection and Localization Based on Deep Learning for Road and Railway Smart Mobility. MDPI, pp.1-15.

[19] Hanxiao Rong, Alex Ramirez-Serrano, (Member, Ieee), Lianwu Guan1 , And Yanb. (2020). Image Object Extraction Based on Semantic Detection and Improved K-Means Algorithm. IEEE Access. 8, pp.1-11.

[20] Arora, Adwitiya; Grover, Atul; Chugh, Raksha; Reka, S. Sofana (2019). Real Time Multi Object Detection for Blind Using Single Shot Multibox Detector. Wireless Personal Communications, pp.1–11. doi:10.1007/s11277-019-06294-1

[21] Fan Sun, Xiangfeng Zhang, Yunzhong Liu and Hong Jiang. (2022). Multi-Object Detection in Security Screening Scene Based on Convolutional Neural Network. MDPI. .(.), pp.1-22.

[22] Md. Ashfakur Rahman1 , Subhra Prosun Paul2 , Mrinmoy Das3 , Md. Mamun Hossain4 ,. (2019). Convolutional Neural Networks based multi-object recognition from a RGB image. International

Conference on Electrical, Computer and Communication Engineering (ECCE). .(.), pp.1-6.

[23] Mauri, Antoine; Khemmar, Redouane; Decoux, Benoit; Ragot, Nicolas; Rossi, Romain; Trabelsi, Rim; Boutteau, RÃ©mi; Ertaud, Jean-Yves; Savatier, Xavier (2020). Deep Learning for Real-Time 3D Multi-Object Detection, Localisation, and Tracking: Application to Smart Mobility. Sensors, 20(2), pp.1–15. doi:10.3390/s20020532

[24] Mhalla, Ala; Chateau, Thierry; Amara, Najoua Essoukri Ben (2019). Spatio-temporal object detection by deep learning: Video-interlacing to improve multi-object tracking. Image and Vision Computing, 88(), 120–131. doi: 10.1016/j.imavis.2019.03.002

[25] Jinming Wang 1 , Ahmed Alshahir 2 , Ghulam Abbas 3 , Khaled Kaaniche 2,* , Moham. (2023). A Deep Recurrent Learning-Based Region-Focused Feature Detection for Enhanced Target Detection in Multi-Object Media. MDPI. .(.), pp.1-18.

[26] Ahmed, Abrar; Jalal, Ahmad; Kim, Kibum (2020). A Novel Statistical Method for Scene Classification Based on Multi-Object Categorization and Logistic Regression. Sensors, 20(14), pp.1–20. doi:10.3390/s20143871

[27] Yunpeng Wang;Xixian Wang;Daxin Tian;Xuting Duan;He Liu;Yinsheng Gong;Zhengguo Sheng;Victor C.M. Leung; (2019). A Multi-object Detection Method Based on Connected Vehicles . Proceedings of the 9th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, pp.1–9. doi:10.1145/3345838.3356000

[28] Dan Liu1,3, Yajuan Wu1 , Yuxin He2 , Lu Qin2 and Bochuan Zheng. (2021). Multi-Object Detection of Chinese License Plate in Complex Scenes. *Computer Systems Science & Engineering*. 36(1), pp.1-12.

[29] P, Ramya P; James, Ajay (2019). *Object Recognition and Classification Based on Improved Bag of Features using SURF AND MSER Local Feature Extraction, IEEE, pp.1–4.* doi:10.1109/ICIICT1.2019.8741434

[30] Du, Kun; Song, Jilai; Wang, Xiaofeng; Li, Xiang; Lin, Jie (2020). *A Multi-Object Grasping Detection Based on the Improvement of YOLOv3 Algorithm, IEEE, pp.1027–1033.* doi:10.1109/CCDC49329.2020.9164792

[31] Novak, J., & Radovanović, M. (2020). Transfer learning strategies for deep convolutional networks in image classification. Informatica, 44(2), 265–274.

[32] Ramachandran Alagarsamy and Dhamodaran Muneeswaran. (2021). Multi-Object Detection and Tracking Using Reptile Search Optimization Algorithm with Deep Learning. *MDPI*, pp.1-17.

[33] Marolt, M., & Korošec, P. (2022). Real-time object segmentation with hybrid deep learning architectures. Informatica, 46(1), 59–68.

[34] Sadia Afroze, Md. Rajib Hossain and Mohammed Moshiul Hoque. (2022). DeepFocus: A visual focus of attention detection framework using deep learning in multi-object scenarios. *Journal of King Saud University – Computer and Information Sciences*. 34, pp.10109–10124.

[35] Rafique, Adnan Ahmed; Jalal, Ahmad; Kim, Kibum (2020). *Automated Sustainable Multi-Object Segmentation and Recognition via Modified Sampling Consensus and Kernel Sliding Perceptron. Symmetry, 12(11), pp.1–25.* doi:10.3390/sym12111928

[36] Liang Hou, Chunhua Chen, Shaojie Wang, Yongjun Wu and Xiu Chen. (2022). Multi-Object Detection Method in Construction Machinery Swarm Operations Based on the Improved YOLOv4 Model. *MDPI*, pp.1-14.

[37] Ke, Bo; Zheng, Huicheng; Chen, Lvran; Yan, Zhiwei; Li, Ye (2019). *Multi-object Tracking by Joint Detection and Identification Learning. Neural Processing Letters, pp.1–14.* doi:10.1007/s11063-019-10046-4

[38] RASHID ALI, RAN LIU, YONGPING HE, ANAND NAYYAR, AND BASIT QURESHI. (2021). Systematic Review of Dynamic Multi-Object Identification and Localization: Techniques and Technologies. *IEEE Access*. 9, pp.1-27.

[39] Shuqi Wang and Meng Chen. (2023). A LiDAR Multi-Object Detection Algorithm for Autonomous Driving. *MDPI*, pp.1-16.

[40] Padmaja, Budi; Myneni, Madhu Bala; Krishna Rao Patro, Epili (2020). *A comparison on visual prediction models for MAMO (multi activity-multi object) recognition using deep learning. Journal of Big Data, 7(1), pp.1–15.* doi:10.1186/s40537-020-00296-8

[41] Jansi Rani S.V., Raghu Raman V., Rahul Ram M., and Prithvi Raj A. (2022). Multi object detection and classification in solid waste management using region proposal network and YOLO model. *Global NEST Journal*. 24(4), pp.743-751.

[42] Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J. and Zisserman, A., 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2), pp.303-338.

[43] Redmon, J. and Farhadi, A., 2018. YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

[44] Ronneberger, O., Fischer, P. and Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention (pp. 234–241). Springer.

[45] Simonyan, K. and Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.

[46] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 86(11), pp.2278–2324.

Appendix A: Fold-wise Evaluation and Statistical Analysis’

Here is the fold-wise evaluation and statistical comparison table between EL-MODC and YOLOv3:

Fold	**EL-MODC Accuracy (%)**	**YOLOv3 Accuracy (%)**	**EL-MODC Precision (%)**	**YOLOv3 Precision (%)**	**EL-MODC Recall (%)**	**YOLOv3 Recall (%)**	**EL-MODC F1-Score (%)**	**YOLOv3 F1-Score (%)**
Fold 1	96.2	94.3	96.0	93.8	96.1	94.0	96.0	94.1
Fold 2	96.4	94.6	96.3	94.1	96.3	94.3	96.2	94.4
Fold 3	96.5	94.5	96.4	94.2	96.5	94.4	96.4	94.3
Fold 4	96.0	94.0	95.9	93.7	96.0	94.1	95.9	94.0
Fold 5	96.6	94.4	96.5	94.0	96.4	94.2	96.5	94.2