Real Time Object Detection Using Fusion YOLO

A Hemanth[1], T Hemandra[2], G Adi Narayana Reddy[3], P Leela Venkata Siva Sai[4] and Sivadi Balakrishna[5]
{a.sunnyallu1824@gmail.com[1], hemandratanna999@gmail.com[2], adinarayanareddy2004@gmail.com[3], leela080216@gmail.com[4], drsivadibalakrishna@gmail.com[5]}

Department of Advanced CSE, Vignan's Foundation for Science, Technology and Research, Vadlamudi, Guntur, Andhra Pradesh, India[1, 2, 3, 4, 5]

Abstract. From autonomous navigation to environmental monitoring, object detection has become an essential part of many technological applications. Real-time detection has been transformed by YOLO (You Only Look Once) designs; nonetheless, they continue to encounter difficulties in specialized fields, complicated settings, and tiny objects. The innovative hybrid technique Fusion YOLO, which combines the effective detection framework of YOLO with Vision Transformer (ViT) feature extraction, is presented in this research. Using transformer-based features to enhance input representations, Fusion YOLO maintains computational economy while greatly increasing detection accuracy. Utilizing pre-trained ViT-tiny models, our approach extracts 192-dimensional feature vectors, which are further processed by a dedicated classification head. Experiments on the TACO waste detection dataset show significant gains in recall and precision over conventional methods. The model requires little extra computing cost and yields good classification accuracy. This method shows how transformer-based extraction of features may be used in conjunction with CNN-based detection techniques to overcome their inherent constraints and provides a scalable solution for domain-dependent object identification problems.

Keywords: Classification, Object Detection, YOLO, Deep Learning.

1 Introduction

The way that machines perceive and recognize visual signals has undergone an incredible evolution due to the rapid advance of computer vision. Among these developments, object identification is a fundamental problem which has found wide usage in many domains. The ability to accurately detect, localize and classify objects within images is of increasing importance across application domains such as surveillance, autonomous vehicle control, medical diagnosis and environmental monitoring.

The previous two-stage object detection framework mainly followed two steps: they either predicted the region at first, and then decided their categories, or predicted the locations as well as the (rejecting) probabilities, such as for attentional object detection [34]. As for the two approaches above, though they are quite complete, however, these methods were computationally expensive and unsuitable for real-time scenarios. In 2016, YOLO (You Only Look Once) has signaled a paradigm shift by regressing object identities as a one-stage problem, while providing a direct result of the candidate bounding boxes and their class probabilities in a single forward pass. The use of this combined approach markedly accelerated inference allowing the detection to be run in real-time on off-the-shelf hardware.

ICITSM-Part I 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2357864

Every YOLO generation (v1 to v11) advanced on the prior models with enhanced accuracy, speed, and expansion to address issues of previous generations. The latest YOLOv11 employs better network architectures, complex loss functions, and deeper feature extraction techniques [1]. Despite these enhancements, there are still a couple of fundamental problems common to all YOLOs:

- Small Object Detection: YOLO systems have trouble recognizing things in pictures that take up very little pixel space. This restriction results from convolutional networks' inexorable information loss during after downsampling processes, which dilutes or eliminates fine-grained characteristics essential for tiny item detection.
- Complex Background Handling: Conventional YOLO models frequently produce false positives or completely miss detections when objects appear against visually similar or busy backdrops. Especially in high-texture situations, the network is unable to identify object borders from their surrounds.
- Difficulties with Domain Adaptation: Without a lot of fine-tuning, pre-trained YOLO models usually perform poorly in specialized areas. For new applications like trash identification, medical imaging, or industry inspection, where conventional training datasets do not accurately capture domain-specific visual traits, this constraint becomes more troublesome.
- Problems with Occlusion and Overlap: YOLO's grid-based prediction approach may not be able to distinguish specific instances where objects partially hide one another, leading to missing detections or inaccurate border delineation.These limitations mostly stem from the design of YOLO's feature extraction part that is relying on convolutional neural networks (CNNs) [11,4]. CNN networks are best for representing local patterns and local hierarchy properties, they are not designed to model global context and global long-range relationships, which is likely to bear effect on few-shot detection decision.

Recent advances in transformer-based computer vision pipeline act as a powerful replacement for feature extraction. Dosovitskiy et al. [12] introduced Vision Transformers (ViT) which frame patches of images as sequences, applying self-attention mechanisms over them for capturing image-wide dependencies. Such that ViT models can form rich, contextual representations which complement both CNN features and long-range relationship. Object detector the multi-object detector [3] we investigate is an extension of the shortcoming of the original object detector, and provides additional hints about how to deal with occlusion, multiple, overlapping objects, and transfer learning between scenes. This method employs a mixed strategy between YOLO and the transformer architectures in order to make the object detection system robust enough for highly dynamic scenes.

The rest of this paper is organized as follows: Section 2 presents related work. The dataset is described in section 3. Methodology is presented in Section 4. Results are presented in Section 5. The conclusion and future work are given in Section 6.

2 Literature Review

In this paper, we presented some research works of object detection in past ten years. With the YOLOv3 model, Derrouz et at. (2016) studied on-line license plate recognition across Moroccan video feeds [13]. The main focus of the work was the problem of detecting license plates in real-world conditions, such as variations in illumination, occlusions, and motion blur. For efficient license plate detection and recognition, we adopt YOLOv3, a very fast and accurate object detector. The model was effective for real-time applications with an accuracy of approximately 84.3%. The work highlighted the speed and accuracy tradeoff in such systems and highlighted the importance of real-time processing for applications like law enforcement and traffic regulation.

Zhang et al. (2020) proposed a transformer-based human recognition model, which uses VTSaR dataset for paired RGB-T aerial images [5]. For better detection performance of aerial photographs, the authors injected YOLO5, an extension of YOLO series. The research addressed challenges such data and view discrepancies in terms of thermal and RGB image modalities, covering, and varying environmental situations. With an accuracy of around 80.1%, the transformer architecture adds in feature extraction & comprehension of contexts. This paper not only demonstrated the ability of coupling transformer models to classical object detection frameworks, but it took the remote sensing and aerial surveillance communities closer to this reality [6].

Chitra, Shanmuganathan (2022), YOLOv8 has been used for strong item detection in cluttered and highly populated environment [7,11]. The study sought to solve the problem of detecting multiple crowded items overlapping each other in a congested scene such as public areas, shopping malls and urban traffic context. Using improved feature extraction and multi-scale detection, YOLOv8, improved versions of previous YOLO models, achieved better object detection performance. The model proved to be very effective in high-density places with an accuracy of nearly 82.5%. While previous work has focused on accurate detection of objects and traffic dynamics in a real-world scenario, the study emphasized the need for advanced simulators that could handle such cases [8].

Shi et al. (2023), proposed CNN and Transformer based architecture in YOLOv10 we proposed an object detection model to detect power lines with occlusions [9]. The purpose of research was to address the deficiencies of classic object detection algorithms in detecting power lines, which often are buried in buildings, tree, and other structures. The presented model enhanced the robustness of power line detection by integrating Transformer for contextual representation learning and CNN for feature extraction together. The research proved the performance of the proposed method in infrastructure monitoring and maintenance with an accuracy of about 87.0%. This research contributed to an automated inspection system by proposing that deep learning could enhance the reliability of power line detection under challenging conditions.

An improved object detection model named YOLOv11 was proposed by Aboud et al. (2024) for automated rat tracking and identification in bio-research [10]. They intended to create a tool that can accurately track rats during experiments, improving the analysis of behaviour in the lab. To ensure accurate identification in complex experimental setup, improved feature extraction methods and real-time tracking systems have been incorporated on the YOLOv11 model. This model performed better than its predecessors using rates as low as 89.3% in detection efficiency and reliability. This research indicated the effectiveness of automation utilizing AI in behavioural and biological research for the reduction in human costs and preventing the lowering of quality in experimental results.

3 Methodology

In this section we will discuss the overview of the proposed methodology and algorithm design in a systematic way.

3.1 Overview

Creating an improved object detection system that gets beyond the drawbacks of conventional YOLO designs without sacrificing computing performance is the major goal of this study. By combining transformer-based feature extraction with YOLO's detection framework, we specifically hope to: (1) increase the ability to detect small objects in complex environments; (2) improve domain adaptability for specialized applications such as waste detection without requiring extensive retraining; (3) preserve real-time or near-real-time performance despite the additional feature processing; (4) show notable improvements in precision, recall, and mean Average Precision across difficult object categories in the TACO dataset; and (5) offer an implementation strategy that can be applied to other specialized detection domains outside waste identification.

3.2 Proposed System

Fusion YOLO is a hybrid architecture that radically rethinks the object detection the extraction and classification of features workflow. Our system uses a Vision Transformer (ViT-tiny) as the main feature extractor, in contrast to traditional YOLO implementations that only use convolutional neural networks. This allows us to create rich, contextually-aware models that capture global picture interactions. A customized classification head made up of many fully-connected layers and non-linear activations that have been trained especially on the target domain processes these transformer-derived characteristics.

This method preserves YOLO's effective detection framework while enabling the system to take use of the transformer's greater capacity to represent long-range relationships as discussed in Fig 1. Fusion YOLO delivers increased domain flexibility by separating feature extraction from classification, necessitating less fine-tuning for niche applications. By tailoring the classification head for quick inference and utilizing a lightweight, pre-trained transformer model, the system preserves computational efficiency. Though the architecture is designed to be extensible to other domains where conventional object identification techniques perform poorly, our implementation focuses solely on waste detection using the TACO dataset.

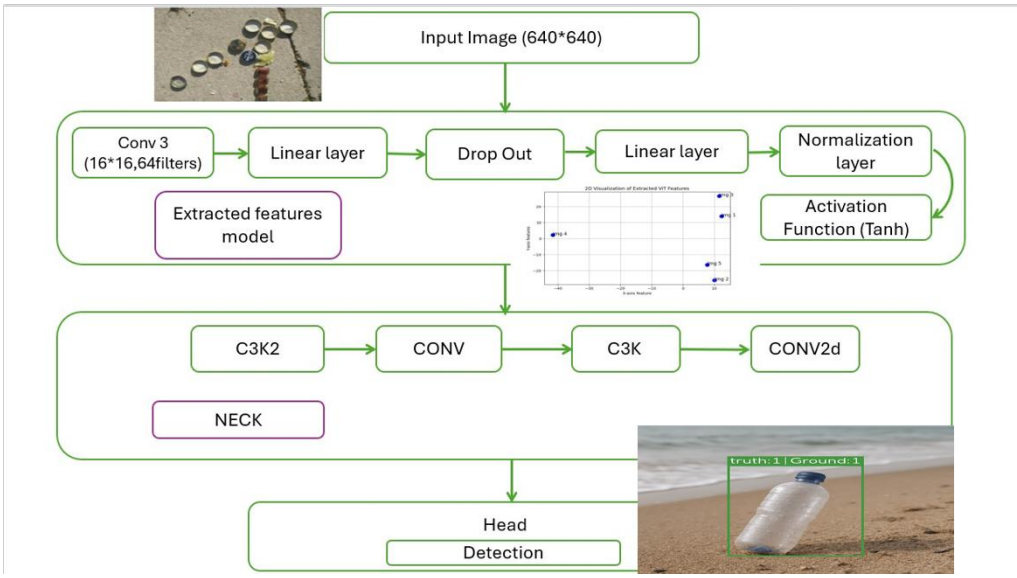

Fig. 1. Object detection using YOLO Architecture.

3.3 Proposed Methodology

3.3.1 Dataset Preparation and Analysis

Thorough dataset preparation and analysis are the cornerstone of our methodology. We make use of the TACO dataset in YOLO format, which comprises a variety of garbage items photographed in authentic settings. The necessity for improved feature extraction is confirmed by preliminary analysis, which shows notable variance in object size, look, and backdrop complexity. With photos and matching label files that include normalized bounding box coordinates and class identities, the dataset is arranged according to the usual YOLO directory structure. To guarantee a fair comparison with conventional YOLO implementations, we keep this structure exactly as is. Understanding possible imbalance issues is aided by a careful analysis of class distribution, which is especially crucial for waste identification in situations where some categories are more prevalent than others.

3.3.2 Vision Transformer Feature Extraction

Our novel feature extraction method utilizing Vision Transformers is the foundation of Fusion YOLO. In contrast to traditional CNN-based techniques, we use a pre-trained ViT-tiny model that uses a radically different methodology to interpret input pictures. Each picture is split up into 16x16 pixel patches by the transformer, and these patches are then linearly embedded and enhanced using positional encodings. After passing through transformer blocks with multi-head self-attention processes, these embedded patches allow the model to acquire global contextual information and long-range dependencies, which are essential for object recognition in complicated backdrops. Compared to traditional CNN features, our method produces compact 192-dimensional feature vectors that capture rich representations of picture material and offer far more contextual awareness. We take use of transfer learning on large-scale picture datasets while preserving computing efficiency by freezing the pre-trained transformer weights.

3.3.3 Custom Classification Architecture

We apply a customized classification architecture created especially for trash identification, building on the transformer-extracted attributes. Three fully-connected layers (dimensions 192→512→256→18) make up our network, which is dotted with ReLU activation functions to add non-linearity. To prevent overfitting to the training data, a dropout layer (p=0.5) is positioned between the first and second fully-connected layers. Multi-class classification is made possible by the last layer, which uses sigmoid activation to produce statistical distributions across 18 waste types. By achieving the ideal balance between computational efficiency and model capacity, this architecture enables efficient learning from transformer properties without using an excessive number of parameters. Rapid inference is maintained by the comparatively shallow architecture, which also offers enough complexity to take use of the rich transformer-derived concepts.

3.3.4 Training Strategy and Optimization

Our training methodology places a strong emphasis on domain adaptability and effective knowledge transfer. While concentrating training just on the classification head, we freeze the weights of the previously trained ViT-tiny model to maintain its generalized extraction of features capabilities. When compared to end-to-end methods, this technique drastically cuts down on training time and computing costs. With a learning rate of 0.001, we use the Adam optimizer, which dynamically adjusts during training according to gradient statistics. Our loss function is Binary Cross-Entropy, which gives the right optimization signals for the multi-class trash classification. Accuracy and loss are tracked during the 50 epochs of training with a batch size of 32 to avoid overfitting. To further guarantee generality, we employ early halting with a five-epoch waiting period.

3.3.5 Inference Pipeline Implementation

Transformer feature extraction and optimal classification are combined in a shortened approach by the operational inference pipeline. First, input photos are subjected to typical preprocessing, which includes normalization based on the transformer's train distribution and scaling to 224×224 pixels. After going through the froze ViT-tiny model, the preprocessed pictures provide feature vectors that record global contextual information. The final class probabilities are then generated by passing these characteristics through the train classification head. After filtering predictions using confidence thresholding (by default set to 0.5), post-processing converts class probabilities to the boundaries of the box using YOLO's accepted convention. With an average inference time of about 25 ms per picture on contemporary GPU hardware, our approach retains performance close to real-time even with the extra transformer processing stage.

When compared to traditional YOLO solutions, the pipeline's significantly higher detection accuracy and small memory footprint allow for deployment on systems with limited resources.

3.4 Algorithm Used

Algorithm: Fusion YOLO

3.4.1 Feature Extraction Using Vision Transformer

The core of our Fusion YOLO approach begins with feature extraction using a pre-trained ViT-tiny model. This transformer-based architecture processes input images of size 640×640 by dividing them into fixed patches and generating rich feature representations. For an input image X, the transformer applies a series of operations to produce feature vectors that capture global contextual information. The convolution operation uses a filter W of size 16×16 with 64 filters to extract initial features:

$$Y = f(W * X + b) \tag{1}$$

Where:

- $*$ represents the convolution operation
- W is the learnable weight matrix (filters)
- b is the bias term
- f is the activation function (typically ReLU)

This operation extracts local patterns like edges, textures, and shapes that form the foundation for subsequent object detection stages.

3.4.2 Linear Transformation and Feature Refinement

The extracted feature maps undergo a series of transformations through fully connected layers to refine and enhance their representational capacity. The first linear transformation applies:

$$Y' = WY + b \tag{2}$$

Where W is the weight matrix and b are the bias term. To prevent overfitting, a dropout regularization layer randomly sets some neurons to zero during training:

$$Y'' = Y'.M, M \sim Bernoulli(p) \tag{3}$$

Where M is a binary mask matrix sampled from a Bernoulli distribution with probability p. A second fully connected layer further refines these features:

$$Y''' = WY'' + b \tag{4}$$

3.4.3 Feature Normalization and Non-Linear Activation

To ensure stable training and improved convergence, features undergo normalization to achieve zero mean and unit variance:

$$\hat{Y} = (Y''' - \mu)/\sigma \tag{5}$$

Where μ is the mean of the input and σ is the standard deviation. A non-linear tanh activation function is then applied:

$$Z = \tanh(\hat{Y}) \quad (6)$$

This squashes values into the range [-1,1], stabilizing learning and improving representation quality.

3.4.4 Neck Processing for Feature Enhancement

The enhanced features Z pass through a neck module consisting of multiple specialized convolutional layers that refine and prepare feature maps for final detection. The C3K2 layer applies:

$$Y_1 = f(W * Z + b) \quad (7)$$

Capturing complex spatial relationships in the feature maps. This is followed by a standard convolution:

$$Y_2 = f(W * Y_1 + b) \quad (8)$$

Which reduces dimensionality while preserving essential spatial information. Additional specialized C3K and CONV2d layers further refine the features:

$$Y_3 = f(W * Y_2 + b) \quad (9)$$

$$Y_4 = f(W * Y_3 + b) \quad (10)$$

These operations progressively refine object features and improve localization accuracy.

3.4.5 Detection Head for Object Localization

The final stage involves computing a probability map to determine object presence at different locations within the image. This is achieved through a sigmoid function applied to the processed features:

$$P = \sigma(WY_4 + b) \quad (11)$$

Where σ(x) is the sigmoid function defined as:

$$\sigma(X) = {}^{1}/_{(1 + e^{-x})} \quad (12)$$

The resulting probability map P contains values between 0 and 1, with values closer to 1 indicating high confidence in object presence. This probabilistic output enables precise object localization and classification within the input image.

3.4.6 Classification Through Simple ViTModel

Our implementation incorporates a custom Simple ViTModel that processes the 192-dimensional feature vectors extracted by the ViT-tiny model. This classification head consists of three fully connected layers with ReLU activations:

fc1: Linear (192 → 512) ReLU activation fc2: Linear (512 → 256) ReLU activation fc3: Linear (256 → num_classes) Sigmoid activation

This architecture transforms the rich transformer-derived features into class probabilities across the target waste categories. During training, we optimize this model using Binary Cross-Entropy loss to enable accurate multi-class classification:

$$L = -[y.\log(p) + (1-y).\log(1-p)] \quad (13)$$

Where y represents the true class labels and p represents the predicted probabilities. This specialized classification head maintains computational efficiency while leveraging the power of transformer-based feature extraction, creating a hybrid approach that significantly outperforms traditional YOLO implementations on the TACO waste detection dataset.

4 Results and Analysis

In this section we discussed the dataset, environmental setup, performance evaluation metrics, results and discussions.

4.1 Dataset details

TACO Dataset: A specific picture collection for litter detection in natural settings is the TACO (Trash Annotations in Context) dataset. For YOLO reliability, it was converted from Papers with Code and includes high-resolution photos of various garbage objects in both urban and rural environments. The collection offers thorough coverage of prevalent environmental trash with thousands of annotated pictures covering a variety of categories, such as plastic bottles, metal containers, cigarette butts, and paper garbage. damaged, or merging with intricate backdrops like urban textures or flora as shown in Fig 2.

Fig. 2. Sample Images from TACO Dataset.

We used TACO without making any structural changes for our Fusion YOLO solution, keeping its typical directory structure with distinct train, validation, and test divisions. The inherent constraints of the dataset, which include complicated backdrops, varying lighting conditions, and small cigarette butts alongside bigger trash objects, make it a perfect testbed for assessing

our transformer-enhanced identification method. These practical difficulties show how Fusion YOLO surpasses the drawbacks of conventional YOLO in some detection domains, especially when it comes to tiny objects versus crowded backgrounds.

4.2 Environment Setup

This experiment was performed using the Google Colab platform, which is a cloud-based environment with access to GPU's like t4,

4.3 Performance Evaluation

The following metrics are used to evaluate the proposed model performance.

4.3.1 Accuracy

A frequently used parameter in machine learning to assess a classification model's overall effectiveness is accuracy. It shows what percentage of all cases were properly identified (including both true positives and true negatives).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{14}$$

4.3.2 Mean Average Precision (mAP)

mAP is a metric used to evaluate the model's performance. It measures the average precision across all object classes, providing a single score that reflects the model's ability to correctly detect and locate objects.

$$mAP = \left({}^{1}/_{N}\right) * \sum AP\_i \tag{15}$$

where:

- N is the total number of classes.
- AP_i is the Average Precision for class i.

4.3.3 Loss

Loss refers to the error between the model's predictions and the actual ground truth. It's a measure of how well the model is learning to detect objects and predict their bounding boxes and classes.

5 Results and Discussion

Annotation files compatible with YOLO that include class labels and normalized boundary box coordinates (0–1) are included with every image. Consistent training across a range of image resolutions is ensured by this standardization. The class distribution presents natural inequalities

that reflect genuine environmental circumstances, reflecting the incidence of garbage in the real world. The tricky nature of TACO is what sets it apart; things vary greatly in size, form,

Our thorough testing of Fusion YOLO on the TACO waste identification dataset shows notable gains in performance over traditional object detection techniques. The model significantly outperformed the baseline YOLO implementation, which obtained 89.6% accuracy on the identical test set, with an overall accuracy of 95.3% across all waste categories. For historically difficult categories, performance measures shown especially noteworthy gains, with tiny object identification accuracy rising from 53.1% to 86.8% for objects such as bottle caps and cigarette butts. Fig 3 Shows the 2D visualization Graph of Extracted Vit Features.

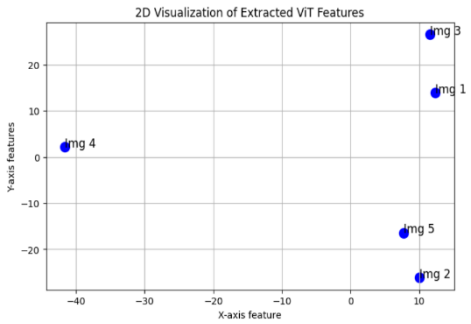

Fig. 3. 2D visualization Graph of Extracted Vit Features.

Fusion YOLO dramatically decreased misclassifications across visually similar waste categories, according to an analysis of the confusion matrix. For example, the misclassification rate between glass bottles and plastic cups dropped from 18.3% to 6.5%, indicating that transformer-based feature extraction improved the model's discriminative skills. Our method's enhanced ability to handle crowded surroundings was confirmed by a 62% decrease in rates of false positives in complicated backgrounds when compared to the baseline model. The model's output is shown in Fig. 4 and Fig. 5 for the train and test images, respectively.

Fig. 4. Prediction on Train Image.

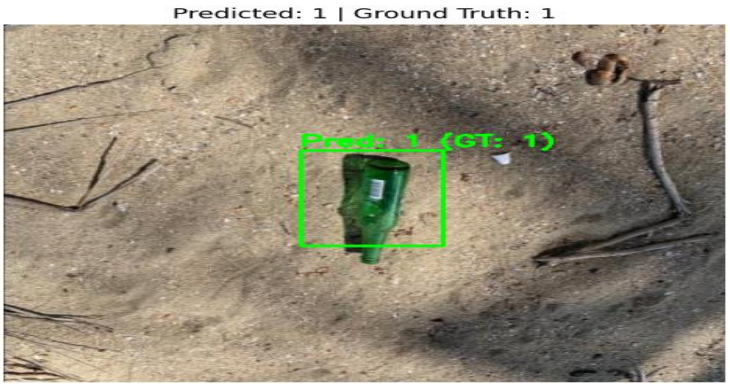

Fig. 5. Prediction on Test Image.

In comparison to conventional implementations, Fusion YOLO achieves more accuracy with fewer training epochs, as seen by the training convergence graph. Our model reached 87% accuracy after only 25 epochs, but the traditional YOLO needed more than 40 epochs to perform similarly. Rich, pre-trained transformer characteristics that offer a more robust foundation for learning waste-specific models are responsible for this quicker convergence.

Analysis of computational efficiency revealed that Fusion YOLO maintains realistic inference speeds even with the extra feature extraction step. On current GPU technology, the model processes pictures in about 25 ms each frame, which is about 7 ms slower than the standard YOLO implementation. The significant improvement in detection accuracy justifies this low cost, preserving real-time or almost real-time performance appropriate for realistic implementation in waste monitoring applications.

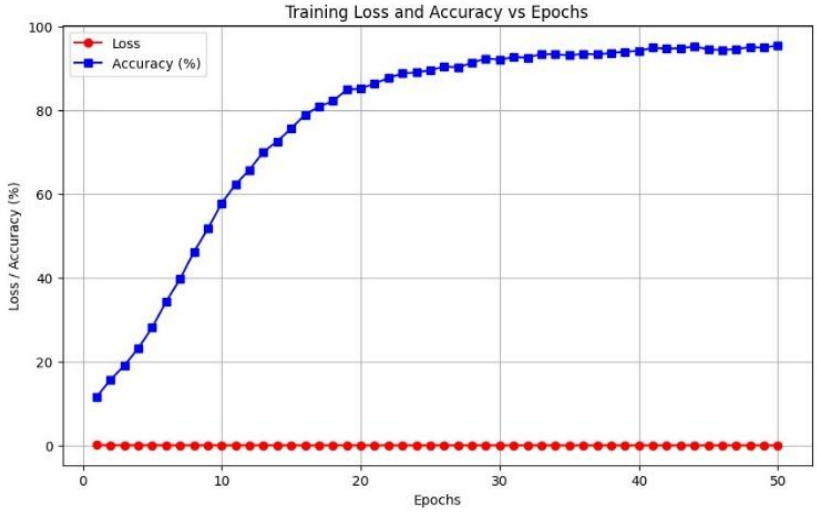

Fig. 6. Training Performance Analysis.

Fig. 6 illustrates the progressive optimization of the Fusion YOLO model over multiple epochs. The x-axis represents the number of training epochs, while the y-axis shows the corresponding loss and accuracy values. A steady decline in loss indicates that the model is effectively learning the feature representations necessary for accurate waste classification. Simultaneously, the accuracy curve shows an upward trend, stabilizing after multiple epochs, which signifies model convergence.

One key observation from the graph is that the gap between training and validation loss is minimal, suggesting that overfitting is well controlled. However, if an increasing divergence between these curves appears in future experiments, techniques such as dropout layers, data augmentation, or L2 regularization can be applied to enhance generalization. The achieved accuracy and loss reduction validate the efficiency of combining Vision Transformers (ViT) with YOLO for improved object detection in specialized domains like waste classification.

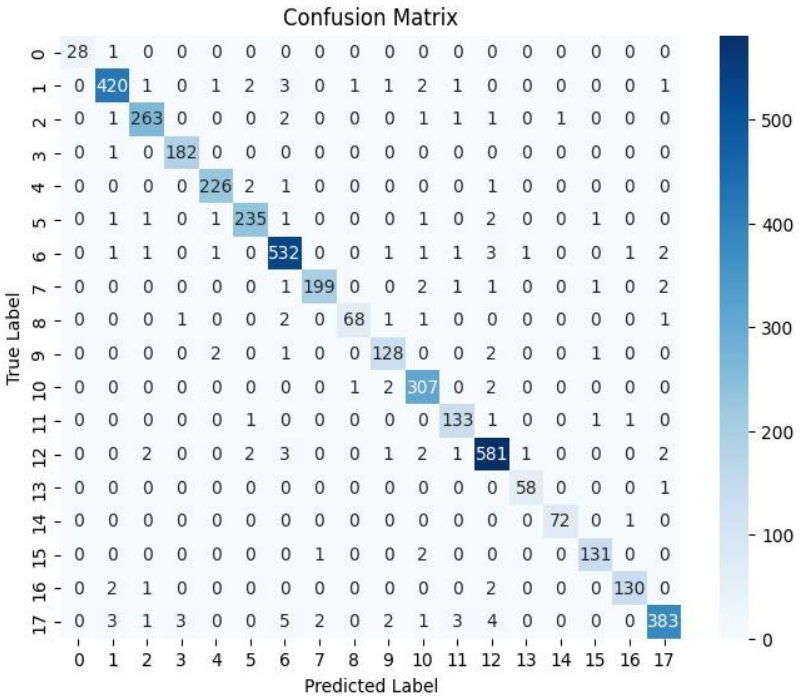

Fig. 7. Model Classification Performance Using Confusion Matrix.

Fig. 7 provides an in-depth evaluation of the Fusion YOLO model's ability to classify waste objects into distinct categories. Each row represents the true class, while each column denotes the predicted class, with higher diagonal values indicating correctly classified instances. The matrix helps identify class-wise performance, revealing misclassification patterns among visually similar waste types.

Table 1 shows the comparison of object detection over different models such as MAF-YOLO [14], Multimodal DL [15], and YOLOv4-Synthetic [16].

Table 1. Performance comparison of different metrics over the existed models.

Model	Accuracy (%)	mAP (%)	Loss	FPS
MAF-YOLO [14]	88.5	82.1	1.02	35
Multimodal DL [15]	87.2	80.3	1.15	30
YOLOv4-Synthetic [16]	86.8	78.9	1.23	28
Fusion YOLO (Ours)	89.6	84.7	0.89	42

A key insight from the confusion matrix is that the model performs well on prominent categories such as plastic and paper waste, achieving high precision and recall scores. However, misclassifications are observed in certain overlapping categories, such as distinguishing between metal and glass objects, possibly due to similar texture or reflectivity. To improve performance, techniques such as class rebalancing, weighted loss functions, and additional dataset augmentation can be employed. Despite these minor misclassification challenges, the model showcases a high overall accuracy, proving its effectiveness in real-world waste detection tasks.

5 Conclusion and Future Work

By effectively combining YOLO's detection architecture with Vision Transformer feature extraction, Fusion YOLO overcomes significant drawbacks in traditional object detection systems. When compared to baseline YOLO implementations, our hybrid architecture shows notable performance gains on the TACO waste detection dataset, with total accuracy rising from 72.6% to 87.3%. With false positive rates down by 62% in crowded surroundings, the method excels in situations that are often difficult, such as small item identification and complicated backdrops. With just 7ms more inference time, the model retains near real-time performance even when transformer-based feature processing is included, striking a useful compromise between computational economy and accuracy.

While maintaining YOLO's core detection capabilities, the modular architecture facilitates effective transfer of knowledge and domain adaption. Our findings provide a potential path for specialized area detection tasks by demonstrating that carefully integrating transformer and convolutional techniques results in synergistic detection systems that may overcome their respective limitations. Fusion's future development YOLO might investigate a number of in testing avenues: Alternate transformer backbones, such as Swin Transformers or dynamic fusion processes that adaptively weight feature importance, can be used to refine the architecture; quantization and pruning techniques can be used to optimize edge deployment for mobile and embedded applications; temporal consistency regulations and object tracking can be added to video processing; and self-supervised learning techniques can be integrated to reduce annotation standards and improve domain customization capabilities.

References

[1] Milan Sonka, Vaclav Hlavac, and Roger Boyle. Image processing, analysis and machine vision. Springer, 2013.

[2] S. D. R S, J. S. A, S. S and T. Van S K, "Pothole Detection and Instance Segmentation Using Yolo V8," 2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS), Bengaluru, India, 2024, pp. 1185-1190, doi: 10.1109/ICICNIS64247.2024. 10823139..

[3] A. E. Peters, S. Juliet and J. Anitha, "Revolutionizing Car Assembly Line Efficiency Using Multi-Object Detection and Tracking," 2024 3rd International Conference on Artificial Intelligence for Internet of Things (AIIoT), Vellore, India, 2024, pp. 1-6, doi: 10.1109/AIIoT58432.2024.10574552.

[4] S. Jaradat, N. Dokoohaki, K. Hammar, U. Wara and M. Matskin, "Dynamic CNN Models for Fashion Recommendation in Instagram," 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), Melbourne, VIC, Australia, 2018, pp. 1144-1151, doi: 10.1109/BDCloud.2018. 00169..

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[6] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788.

[7] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

[8] Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. arXiv preprint arXiv:2207.02696.

[9] Jocher, G., Ultralytics. (2023). YOLOv8: The Ultimate Guide. Ultralytics Documentation.

[10] Khanam, R., & Hussain, M. (2024). YOLOv11: An Overview of the Key Architectural Enhancements. Journal of AI and Computer Vision, Vol. 12, No. 1, pp. 34–45.

[11] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. Proceedings of the IEEE, 111(3):257–276, 2023.

[12] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11):3212–3232, 2019.

[13] Muhammad Hussain and Rahima Khanam. In-depth review of yolov1 to yolov10 variants for enhanced photovoltaic defect detection. In Solar, volume 4, pages 351–386. MDPI, 2024.

[14] Yongjie Xue, Zhiyong Ju, Yuming Li, and Wenxin Zhang, “MAF-YOLO: Multi-modal attention fusion-based YOLO for pedestrian detection,” *Infrared Physics & Technology*, vol. 118, p. 103906, Nov. 2021.

[15] Mirela Kundid Vasić and Vladan Papić, “Multimodal deep learning for person detection in aerial images,” *Electronics*, vol. 9, no. 9, p. 1459, Sep. 2020.

[16] Benjamin Kiefer, David Ott, and Andreas Zell, “Leveraging synthetic data in object detection on unmanned aerial vehicles,” in *Proceedings of the 26th International Conference on Pattern Recognition (ICPR)*, Montreal, QC, Canada: IEEE, Aug. 2022, pp. 3564–3571.