Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 4, Issue 3, 2026

SYMBOL RECOGNITION AND APPLICATION USING MACHINE LEARNING

Fauzia Talpur[*1], Mir Rahib Hussain Talpur[2], Rana Ehtsham Ul Haq[3], Shakir Hussain Talpur[4], Syed Baig Ali Shah[5], Muhammad Waseem[6], Khan Muhammad Mahar[7]

[*1]*Department of Computer Science, University of Sindh Jamshoro Laar Campus*
[2,3,4,5,7]*Information Technology Centre, Sindh Agriculture University Tandojam*
[6]*Government College University Hyderabad*

DOI: https://doi.org/10.5281/zenodo.19367195

Keywords
Intelligent Detection of Symbols, Traffic Management, Industrial Symbol Detection, Environment

Article History
Received: 31 January 2026
Accepted: 15 March 2026
Published: 31 March 2026

Corresponding Author: *
Fauzia Talpur

Abstract
In the ongoing wave of digitalization, with the advent of the intelligence era, the automation and intelligent detection of symbols have gained significant importance in traffic management and industrial production. This study focuses on symbol detection in two critical domains: traffic sign recognition and industrial symbol detection on iron ladle bodies. Manual detection methods are no longer sufficient to meet the demand for real-time, high-accuracy information processing, necessitating the use of intelligent technologies like machine learning to enable automation. In intelligent transportation systems, rapid and accurate identification of traffic signs improves driver assistance systems, especially under adverse weather or lighting conditions, enhancing road safety. In industrial environments, detecting symbols on iron ladle bodies ensures production safety and efficiency, reducing manual operation costs. This study proposes machine learning-based recognition methods tailored to both scenarios. A high-precision traffic sign detection algorithm based on Faster R-CNN is developed. The gamma transformation is applied to improve feature expression under uneven lighting. The model incorporates a Convolutional Block Attention Module to address network depth issues, enhances shallow feature acquisition, and reduces parameters. A Feature Pyramid Network improves the detection of signs of different sizes. The model achieves 99.79% mAP on GTSDB (↑10.35%) and 87.62% on CCTSDB2021 (31.04%). For industrial symbol detection, a lightweight YOLOv8-based model is proposed for resource-constrained environments. Using PP-HGNetV2 and Ghost-HGNet modules, it reduces parameter count while maintaining performance. BiFPN is introduced for efficient multi-scale feature fusion. On the Iron Ladle dataset, the model achieves 97.57% mAP with 1.29M parameters, 1.72M fewer than the original, with only a 0.27% performance drop. This work demonstrates the effectiveness of machine learning in enabling accurate, efficient symbol detection in complex, real-time traffic and industrial applications.

1. INTRODUCTION
The paper explores the increasing importance of automated symbol recognizers, especially in traffic sign detectors and industrial symbol detectors. Multimedia data and transmission, particularly

images, have become important in the transmission of information, given the growth of the digital information age. The symbols, including the traffic signs and the marks of the industrial products, are essential in the effective and efficient transfer of information (Meifang, 2020). Symbol recognition has become the object of interest in research and the industry, following the growing demand for automated and intelligent systems. The detection of different symbols in images is an important task that requires efficiency and accuracy in a variety of applications, including enhancing road safety and industrial optimization. This paper is devoted to the implementation of machine learning technologies to such challenges, trying to increase the accuracy and effectiveness of the symbol recognition.

The two aspects the research aims at include traffic sign detection and molten iron tank symbol identification in industrial environments. The lighting condition variability, as well as the size of various traffic signs, in the case of traffic sign detection, are critical issues that prevent the correct recognition systems (Meifang, 2020). Likewise, where symbol accuracy is required in industrial symbol detection, e.g., in monitoring molten iron tanks, the efficiency and safety of manufacturing are directly affected by the accuracy of symbols (Xu et al., 2020). The conventional approaches have been unable to deal with these issues because of the weaknesses in dealing with such variables.

Deep learning models, such as Faster R-CNN and YOLO, are among the machine learning methods that are being utilized to address these challenges, as shown to be increasingly effective (Li et al., 2021). The idea is to use these technologies to come up with stronger systems that can respond to environmental changes and adjustments in the size of symbols. The research will enhance the detection power in the real-world situation, which will make a contribution to the automation and intelligence in symbol recognition to meet the increasing demand for an advanced and reliable automated system in society.

It also describes how machine learning is applied to enhance real-time symbol recognition in the industrial setting, especially when restricted computational resources are available, through the introduction of lightweight models that can be used with both mobile devices and at the edge (Singh et al., 2021). The above advances will result in more efficient scales and cost-effective applications of symbol recognition in traffic management as well as in industrial manufacturing processes.

2. Review of Literature

Detecting objects has been a major theme in computer vision, which seeks to determine and locate objects in an image. Conventional approaches normally use manual feature extraction and classification processes, and the processes involved are region proposal generation, feature extraction, and classification. The first models, such as RCNN (Girshick et al., 2014), proposed the selective search procedure in image candidate regions. Convolutional Neural Networks (CNNs) have been used to extract features, and a linear Support Vector Machine (SVM) has been used to classify them. Although RCNN was able to enhance accuracy, it was highly computationally inefficient as it extracted redundant features in each candidate region, and detection processes were done independently, which made optimizing it was not easy.

As a reaction to these shortcomings, Fast RCNN (Girshick et al., 2015) was created. The region-of-interest (ROI) pooling layer of Fast RCNN enabled the extraction of features in the full image instead of individually per region. The pooling technique greatly minimized unnecessary calculations and also made the process faster by combining both the extraction of features and the classification into one and the same network.

The second evolution was Faster RCNN (Ren et al., 2017), which improved on Fast RCNN by adding the Region Proposal Network (RPN). RPN permitted the creation of target candidate boxes in the feature map directly, without any external region proposal techniques, such as selective search. This resulted in Faster RCNN being more efficient, which allowed detecting objects in real-time with greater accuracy, which preconditioned more sophisticated approaches to the matter.

2.1 Development of Approaches: YOLO and Beyond.

Although the accuracy of the RCNN family of algorithms was considered, the multi-stage detection architecture was slower and thus not as useful when real-time detection was required. Redmon et al. (2016) introduced the You Only Look Once (YOLO) algorithm, and this algorithm signified a paradigm shift in real-time object detection. YOLOV1 subdivided the input image into a grid, and it could predict a set of bounding boxes and the probability of a class in every grid cell, making it possible to detect every object in a single pass. This method was very fast in detecting objects, but it was not without its weaknesses in identifying tiny objects and handling the overlapping bounding boxes.

Later versions solved the limitations of YOLOV1. An anchor mechanism was introduced by YOLOV2 (Malaainine et al., 2021), and it improved the recall and location precision of the model, especially for small objects. It also came with a feature-extracting network, Darknet19, which significantly enhanced the processing speed of the model, as compared to earlier models such as VGG16.

Its successor is YOLOV3 (Redmon and Farhadi 2018). This enabled YOLOV3 to deal with different sizes of objects. YOLOV3 also used a deep network, Darknet53, instead of Darknet19, which yielded more features and increased detection accuracy, despite being no slower than in real-time.

It was followed by the evolution of YOLOV4 (2020), which had added the following techniques: cross-scale attention mechanisms, better data augmentation strategies, and more profound feature extraction networks. This also enhanced the accuracy and strength of YOLO, making it one of the top algorithms in real-time object detection. YOLOV5 (Jocher et al, 2022) and YOLOV8 (2023) are more recent advances that follow this pattern and have lighter architectures and loss functions to make them more effective and accurate in an exceptionally wide variety of object detection tasks.

3. Methodology

Intelligent driving has progressively become a reality in recent years as a result of the quick advancement of artificial intelligence technology. The system must be able to correctly detect and comprehend the road environment in order to ensure dependable, intelligent driving. Traffic signs are essential for directing drivers since they are a key component of communicating traffic laws, directing driving directions, and delivering traffic data. Traffic sign detection has become an essential aspect of intelligent driving systems for understanding the road environment in this situation. The automatic recognition of road traffic signals is made possible by artificial intelligence technology, which offers information for smart driving decisions.Two main issues that impair the precision of traffic sign detection are the environmental impact and the inadequate capacity of model feature extraction. Variations in environmental variables like road surface conditions, weather patterns, and lighting changes might alter the color and texture of traffic signs, which would impact the model's capacity to identify and categorize them. These variables contribute to modifications in the background and appearance of traffic signals, which limit the model's capacity to identify and identify them. Simultaneously, the current models are unable to completely capture the wealth of data provided by traffic signals, particularly when dealing with difficult scenarios like occlusion, blurring, or scale change, and the model's feature extraction capability is insufficient to satisfy the needs of precise traffic signal categorization and identification. Consequently, the secret to increasing the accuracy of traffic sign detection is to address these two issues and enhance the model's capacity for feature extraction as well as its capacity for adapting to environmental changes.

A traffic sign detection technique based on Faster RCNN was developed to address the issues of the existing traffic sign detection methods being significantly impacted by light and the model's low accuracy. First, the gamma transform was implemented in the model to improve the capacity for traffic signs to express features in light of the uneven illumination between the sky and non-sky regions in the Image. The gamma transform is able to conduct a nonlinear gray transformation on the image, which improves the model's ability to recognize and locate traffic signals under various lighting circumstances. Second, the issue of deep network degradation is addressed, the feature extraction capabilities of the shallow network are enhanced, and the number of parameters is reduced by employing a high-efficiency convolutional attention module-based network (CBAMEfficientNet). The convolutional attention module enhances the network's capacity to recognize the features of traffic signals and increases the detection accuracy by adaptively weighting the important information in the feature map. The network's capacity to recognize traffic signs of various sizes is further improved by the integration of a feature pyramid network, which allows it to detect traffic signs of varying sizes. The feature pyramid network addresses the challenge of traffic sign size differences by enabling multiscale detection on the feature map at various levels. This improvement enhances the model's ability to detect traffic signs of all sizes, increasing the accuracy and robustness of the detection.

3.1 Faster R-CNN

The evolution of the RCNN series algorithms is first presented in this thesis, and the Faster RCNN algorithms are covered in depth in this section.
The architecture of Faster RCNN, which is an enhancement and extension of the RCNN and Fast RCNN algorithms, is depicted in Figure 3.1. In contrast to RCNN and Fast RCNN, Faster RCNN uses Region Proposal Networks (RPNs) to automatically produce potential areas in images. The fundamental idea behind Faster RCNN is to break down the job of identifying objects into two subtasks: candidate region creation, object categorization, and bounding box regression.

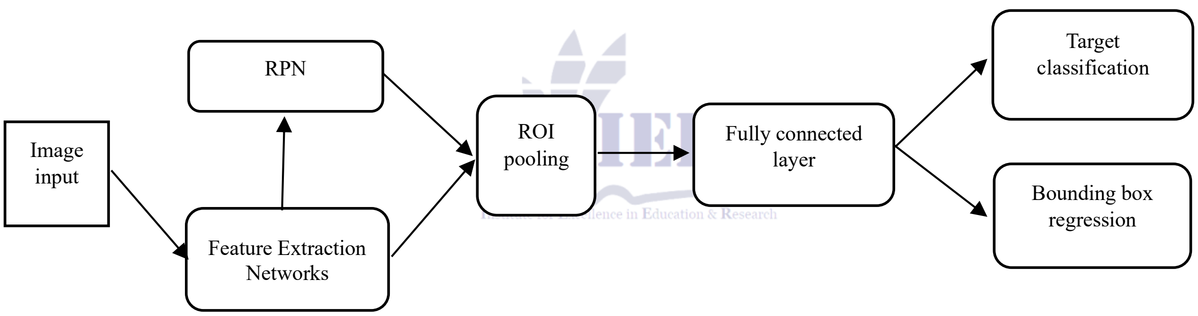

Figure 3.1 Structure of the Faster R-CNN Model

3.2 Feature extraction network

The feature extraction network utilized in the Faster RCNN model is VGG16, which was first put forth by Andrew Zisserman and Karen Simonyan in 2013 and is one of the traditional deep convolutional neural network architectures. It has achieved remarkable results in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and has set a new record for multiple image classification tasks at that time (Simonyan, K., & Zisserman, A., 2014). The Visual Geometry Group at the University of Oxford, to which its founder belongs, gives the VGG network its name. The crucial advancement of the VGG model is the use of the prior AlexNet model.
The convolutional layer, which combines several tiny filters, is shown to perform better than the convolutional layer, which has just one huge filter. The several iterations of the VGG model are named according to their depth and number of parameters, some of which are VGG11, VGG16, and VGG19. The smallest version is VGG11, which has 8 convolutional layers and 3 fully connected layers, whereas the biggest version is VGG19, which has 16 convolutional layers and 3 fully connected layers. With 13 convolutional layers and 3 fully connected layers, VGG16 is in the middle. Additionally, the VGG model includes five pooling layers strategically placed behind various convolutional layers to maintain a

manageable reduction in the feature map's dimensionality, and each convolutional layer is not followed by a pooling layer.

The 224x224 image input that VGGNet receives is in line with the ImageNet competition's criteria. The model's developer maintains a consistent image size by cropping the 224x224 area from the middle of each image as it is input. The VGG model's convolutional layer employs a 3x3 filter for linear transformation and a 1x1 convolutional filter. The ReLU activation function is applied after each convolutional layer, which allows the network to incorporate nonlinearity and thereby increase its expressiveness. The spatial resolution is maintained after convolution by setting the convolution step to 1 pixel at the same time. The VGG model does not utilize Local Response Normalization (LRN), which is mostly used to decrease training time and memory usage, unlike AlexNet. The pooling layer, however, comes after many convolutional layers and helps to lower the feature map's dimensionality and parameters produced at each convolution stage. There are three fully connected layers in the last stage of VGGNet. The first two layers each have 4,096 channels, while the third layer has 1,000 channels, each of which represents a category. With this architecture, VGGNet can do precise categorization. The 3x3 size filter of the VGG model is effective because it captures the spatial characteristics of the image while maintaining the network's manageability and efficiency. In contrast, the use of several smaller convolutional filters helps to minimize the likelihood of the network overfitting during training.

4. Results

4.1 Industrial symbol detection and recognition based on lightweight Yolov8

The automatic identification of molten iron tank characteristics is crucial for the safety and efficiency of industrial manufacturing. In contrast, resources may be scarce in many applications, particularly in intelligent monitoring systems and industrial production facilities, where there may be restrictions on energy usage, storage capacity, and computing power (Gill et al., 2022).

In such a case, adopting a light model is crucial. Lightweight models are designed to minimize the model's computational complexity and number of parameters in order to achieve efficient model inference with limited resources. This lightweight model is advantageous since it not only satisfies the real-time need but also accurately and quickly identifies hot metal cans under resource-constrained circumstances.

Using a lightweight model, you may keep the same level of detection accuracy while minimizing the demand for computer resources and storage space. This not only lowers the cost of the gear but also increases the system's stability and dependability. As an illustration, lightweight models may be installed in edge computing devices or embedded systems at an industrial manufacturing site to provide real-time character recognition of molten metal cans, hence enhancing the production process's safety and efficiency.

This research suggests a character detection model for molten iron tanks based on YOLOV8, which incorporates a large number of C2f modules, leading to a considerable increase in the number of parameters. To meet the demands of lightweight character detection models for molten iron cans, PPHGNetV2 is presented as the architecture of the feature extraction network in order to lower the number of model parameters. As a result, this study further optimizes the model by lowering the number of parameters without sacrificing detection accuracy. The lightweight feature extraction module GhostHGNet is put out, and the PPHGNetV2 architecture is presented, which successfully minimizes the number of parameters in the network model and increases the model's computing efficiency without sacrificing detection accuracy. This research employs the BIFPN network structure in place of the standard PANFPN in order to further decrease the number of parameters in the feature extraction network. The BIFPN combines the concepts of a feature extraction network and bidirectional connection, allowing it to fully utilize the information exchange between feature maps in order to enhance the network's feature expression and perception range. This enhancement not only addresses the problem of the increase in parameter quantity brought about by the feature fusion process, but it also ensures the accuracy of detection, allowing the model to perform exceptionally well in the molten iron character identification job.

4.2 Lightweight YOLOV8 algorithm

The two most crucial elements in YOLOV8 are the feature extraction network and the feature fusion network, which have a direct impact on the model's parameters and performance. The C2f module, which is intended to enhance the model's capacity for feature extraction from the input image, is the core component of the feature extraction network. The C2f module achieves this by adding a lot of bottleneck modules that successfully increase the depth and complexity of the convolutional layer, thereby enhancing the model's capacity to recognize image features. The number of model parameters has, however, increased noticeably as a result of this improvement in depth and complexity.

In contrast, the deep and surface features will be merged via tensor splicing operations after each feature fusion in the feature fusion network. The goal of this experiment is to demonstrate that the model can successfully utilize various levels of feature information to enhance the accuracy of item identification. Nevertheless, the splicing process combines several tensors along a specific dimension, which increases the number of weighted parameters in the model and results in a proliferation of parameters. The number of parameters increases when tensor concatenation is used because each tensor to be spliced has its own parameters, which, after splicing, constitute a new set of parameters.

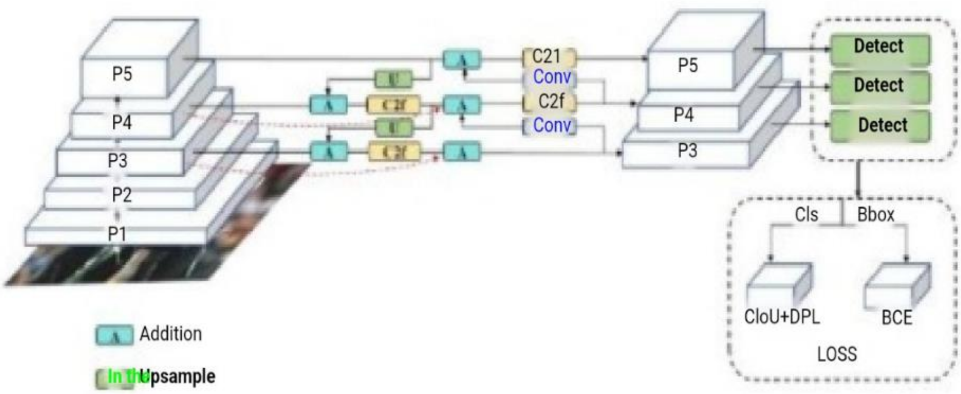

Figure 4.2 Structure of the Lightweight YOLOV8 Model

The model requires more processing power and more training time throughout training and deployment due to the large number of parameters. This study builds upon YOLOV8, as depicted in Figure 4.2, in order to lower the model's parameter count while preserving performance and making it easier to implement and utilize embedded systems or edge computing devices. To further reduce the number of parameters of the feature extraction network while maintaining the feature extraction capability, the lightweight feature extraction module GhostHGNet is proposed, and the PPHGNetV2 architecture is introduced as the Infrastructure of the feature extraction network. This work introduces BIFPN to employ lightweight feature fusion operations and enhance cross-scale connections in terms of feature fusion. This significantly lowers the number of parameters while maintaining the feature fusion capability.

4.3 HGNetV2 network with Ghost convolution

The PaddlePaddle Vision team at Baidu created the high-performance network known as PPHGNet (High-performance GPU Network). The model's accuracy is significantly higher than that of other common models, even when using the same speed. Additionally, the model's inference speed is significantly higher than that of mainstream models, with the same level of accuracy.

The entire architecture of the PPHGNet feature extraction network is seen in Figure 4.3(a), while the entire architecture of the PPHGNetV2 feature extraction network is seen in Figure 4.3(b). Figure 4.3(c) illustrates the structure of HGBlock.

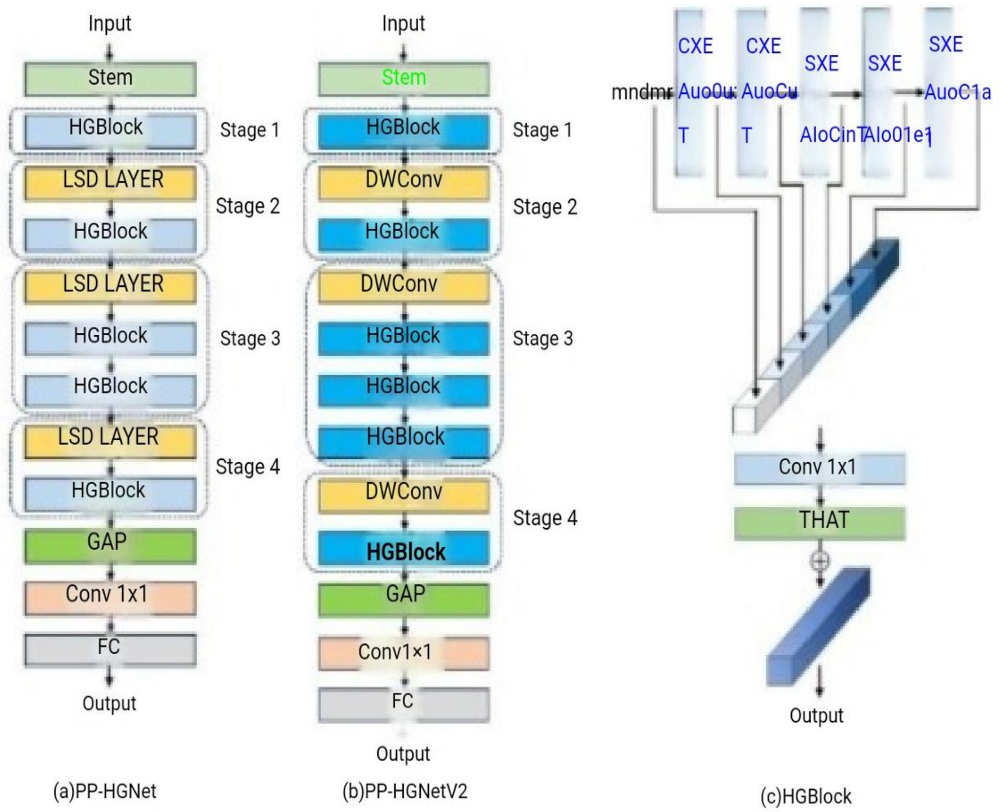

Figure 4.3 PP-HGNet Network Structure

The network's first preprocessing layer, the Stem layer, begins to extract features from the raw input data module for high-grade. It analyzes feature data at various depths for both low-level and high-level features, making it the network's central component. Between the HG modules is the Learnable Down Sampling (LDS) layer, which carries out the down-sampling process, which lowers the spatial dimension of the feature map, decreases the computational burden, and expands the receptive field of the following layers. Before the last classification, Global Average Pooling (GAP) uses the GAP layer to minimize the spatial dimension of the feature map to one vector per feature map, which aids in enhancing the network's resilience to the spatial transformation of the input data. A set of categorization functions is carried out by the last convolutional and Fully Connected (FC) layers.

The high-performance GPU network V2 (PPHGNetV2), which is based on PPHGNet, is further enhanced and optimized. It achieves the ultimate balance between accuracy and latency, and its accuracy is far superior to that of other models with the same inference speed. It excels in tasks such as single-label classification, multilabel classification, object identification, and semantic segmentation. In contrast to PPHGNet, PPHGNetV2 focuses primarily on two changes: increasing the amount of 2x2 convolutional kernel, learning in the Stem portion, and redesigning the Stage distribution of the PPHGNet network.

Huawei has developed GhostNet, a low-cost neural network architecture that prioritizes inference speed and memory usage while taking model performance into consideration, in order to implement neural networks on edge devices like smartphones and wearables (Han K et al. 2020). The Ghost module is the primary component of GhostNet; it can replace the original convolution operation of generating feature maps by generating more feature maps. The typical Ghost module may replace the conventional convolution in two steps, given the input feature X, which has a dimension of 4, where H is the height, W is the width, and C is the number of channels. Its architecture is seen in Figure 4.4.

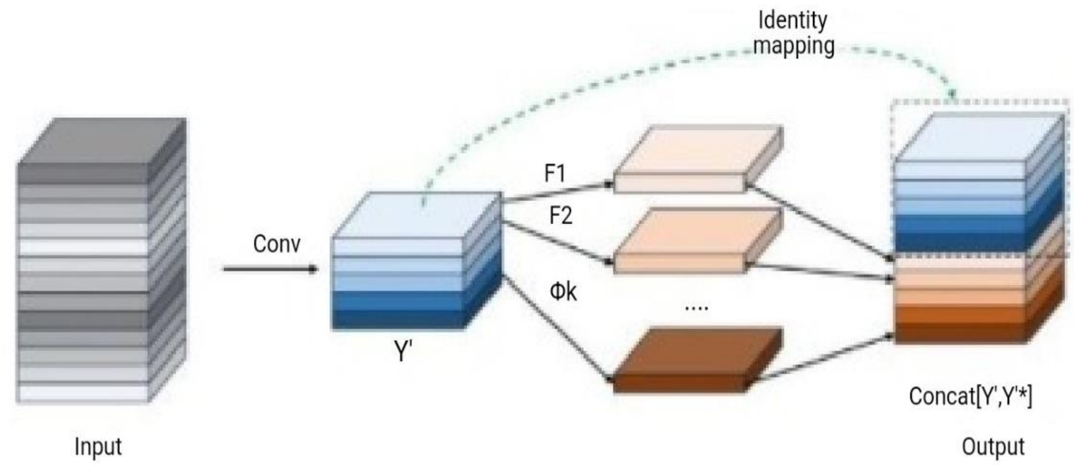

Figure 4.4 Ghost convolutional structures

Y' stands for the intrinsic feature, which typically has a lower dimension than the original output feature, i.e., Cout represents the convolution operation, which is a point-by-point convolution. After that, employ inexpensive methods based on intrinsic, like deep separate convolutions. Features create more features. The two-part feature is connected along the channel dimension, i.e.

$$Y = Concat([Y', Y' * \Phi_k])$$

With the depth separable convolutional filter and Y as the output feature, the representation power of the Ghost module is unavoidably diminished, but it can greatly lower the computational expense. Correct identification depends on the connections between spatial pixels. Spatial information, on the other hand, only captures half of the features in GhostNet through low-cost operations (usually implemented by 3x3 deep separable convolutions). The remaining features are produced exclusively via 1x1 point-by-point convolution, with no interaction with other pixels. This deficiency in spatial information capture can impede additional progress in performance.

To maintain the feature extraction capability while further minimizing the number of parameters in the feature extraction network, a lightweight feature extraction network called GhostHGNet was developed. GhostHGNet combines the Ghost convolution with the HGNetV2 network, and its architecture is depicted in Figure 4.5.

ISSN (e) 3007-3138 **(p)** 3007-312X Volume 4, Issue 3, 2026

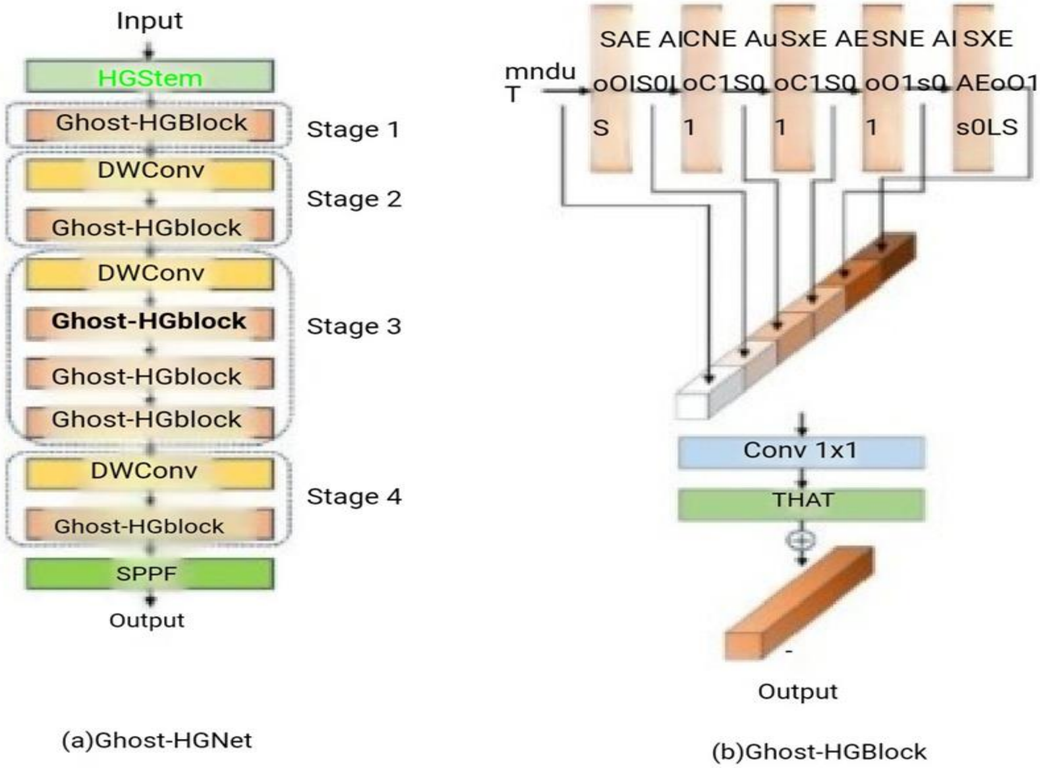

Figure 4.5 Ghost-HGNet Network Structure

4.4 Weighted bidirectional feature pyramid module

The conventional approach to creating feature pyramid networks is top-down, which means beginning with high-resolution feature maps and using upsampling and fusion operations to create a series of feature maps with varying resolutions. This one-way communication, though, may result in knowledge. There are restrictions on the direction of flow, and some important data at the low and high levels may be missed.

BIFPNs, in contrast to conventional FPNS, allow for bidirectional communication between high-resolution and low-resolution characteristics (Tan et al., 2020). This bidirectional connection facilitates a higher degree of integration of features across various scales and enhances the capacity to manage objects of varying sizes and shapes. Consequently, the enhancements suggested in this study will strengthen the basis for target identification in complicated situations and boost the accuracy and robustness of the target detection assignment. Figure 4.6 illustrates the architecture of the BIFPN.

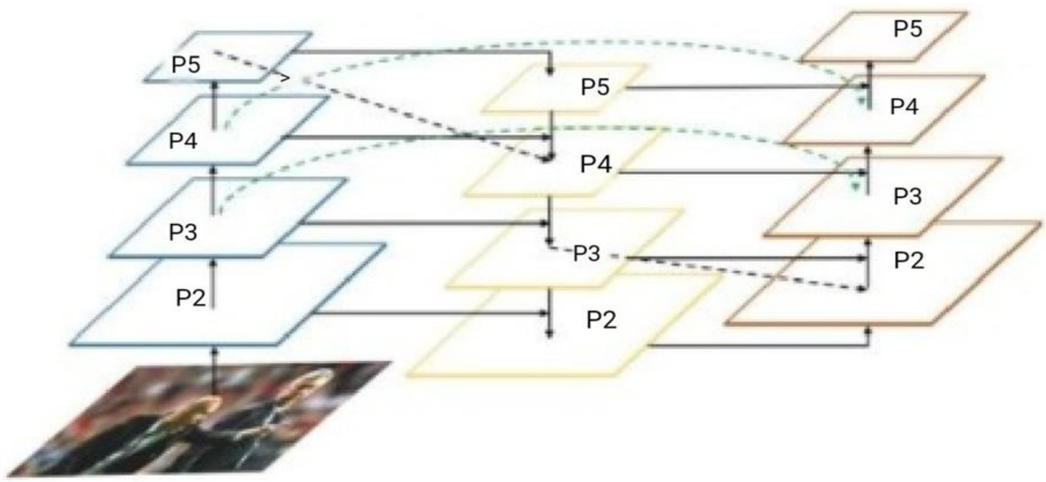

Figure 4.6 BIFPN Network structure

Feature fusion is a critical operation in deep learning network models, and the most popular techniques include: Add tensor splicing and elementwise addition. The tensor splicing technique is used for feature fusion in networks like DenseNet, whereas the element-by-element addition method is used for feature fusion in models like ResNet and FPN. Despite the fact that both are employed to combine feature map data, there are notable distinctions. The element-by-element addition approach increases the amount of information without changing the number of channels, which enhances the precision of image categorization. Although tensor stitching enhances the number of channels in the feature map, it does not enhance the amount of information under each feature. Tensor stitching is implemented on a channel-by-channel basis, whereas element-by-element addition sums up the feature maps of the matching locations.

Since the element-by-element addition process only requires adding the elements at the corresponding position and the computational complexity is linearly related to the size of the feature map, the computational cost of the element-by-element additive operation is typically far lower than that of tensor splicing. The tensor concatenation operation, on the other hand, entails data copying and memory reallocation, which increases the computational complexity. Furthermore, the tensor stitching process might use up more memory space, whereas the element-by-element additive operation only needs to maintain the original feature map and the output feature map. As a result, resource-limited environments are better suited for additive-by-element operations. Tensor splicing operations that fuse features together are used in the PANFPN architecture, which results in a large increase in the number of model parameters.

The structure cannot integrate features across scales. A BIFPN architecture was created in this work to address the aforementioned issues. The addition of features one at a time allows the framework to achieve feature fusion, which mitigates the numerous parameter issues brought on by tensor splicing procedures in the original framework. In addition, the cross-scale connection function is included to enhance the feature integration capability.

5. Discussion

This chapter presents the main findings of this investigation, situating the experimental results in the larger framework of current research on machine learning-based object identification. It analyzes the accomplishments and difficulties of the applied solutions for identifying industrial symbols and traffic signals and assesses their environmental, technological, and social effects.

The objective is to both analyze the performance of the suggested models and critically evaluate how and why they produced particular results. The conversation combines theoretical and practical viewpoints, making it useful for both academics and practitioners. Furthermore, this chapter will provide a thorough comparison of these findings with earlier research and discuss their benefits, drawbacks, and potential future uses.

It is essential to think about ethical and social implications as smart detection systems become more integrated into our daily lives. Traffic detection-based surveillance must strike a balance between safety and the right to privacy. Industrial monitoring should make sure that workers' privacy is protected and that detection technologies are utilized to enhance rather than replace human jobs. Additionally, when training datasets are not representative of all operating conditions, algorithmic bias must be addressed. The development of ethical AI necessitates transparency, fairness, and inclusive design methodologies to guarantee that these technologies benefit all parties involved.

6. Conclusions

The automated and intelligent identification of symbols in two important fields, traffic sign identification and industrial character detection, was the subject of this study. The reliable and effective detection of visual cues has become essential for real-world applications like industrial monitoring and autonomous driving as smart systems continue to expand in scope. The study presented two main machine learning-based solutions: a high-precision traffic sign identification model utilizing an enhanced Faster RCNN architecture, and a lightweight, resource-efficient model for identifying characters on molten iron tanks in industrial settings, based on an optimized YOLOV8 framework.

In summary, this research effectively illustrates how machine learning algorithms may be modified to achieve both high accuracy and efficient use of resources in symbol detection. The models demonstrate strong and adaptable performance in both smart vehicles and industrial monitoring systems. By paving the way for real-world innovation, this research establishes a solid basis for the next generation of smart visual identification systems.

7. REFERENCE

Long Meifang. (2020). Research on Traffic Sign detection and recognition algorithm in complex environment.

Xu, H.P. & Yuan, H Y. (2020). An SVM-Based AdaBoost Cascade Classifier for Sonar Image. *Journal Leee Access*, 8, 115857-115864.

Li Houle, Wang, Fasheng, He Jianjun, (2021). Traffic sign detection based on pseudo-sample regularization Faster R CNN. *Journal of Jilin University (Engineering Science),* 51(04), 1251-1260.

Singh, A., Kumar, D. P. & Shivaprasad K. (2021). Vehicle detection and accident prediction in sand/dust storms [C]// Proceedings of the 2021 International Conference on Computing Sciences (ICCS). *Place of Publication: IEEE*, 107-111.

Girshick R, Donahue J, & Darrell T. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation [C]// Proceedingsof the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. *Place of Publication*, 580-587.

Girshick R. & Fast R-CNN. (2015). [C]// Proceedings of the Proceedings of the IEEE International Conference on Computer Vision. *Place of Publication: IEEE,* 1440-1448.

Ren, SQ, Hek, M, & Girshick R, (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Journal leee Transactions on Pattern Analysis and Machine Intelligence,* 39(6), 1137-1149.

Simonyan, Kzisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *Journal arXiv Preprint arXiv,* 14091556.

Gill, SS., Xu & Mottaviani, C. (2022). Al for next generation computing: Emerging trends and future directions. *Internet of Things,* 19, 100514.

Tan, M., Pang, R. & Le Q V. (2020). Efficientdet: Scalable and efficient object detection [C]// Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. *Place of Publication*, 10781-10790.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Malaainine, M. E. I., Lechgar, H., & Rhinane, H. (2021). Yolov2 deep learning model and gis based algorithms for vehicle tracking. *Journal of Geographic Information System*, *13*(4), 395-409.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Michael, K., ... & Mammana, L. (2022). ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. *Zenodo*.