Traffic sign detection and recognition using Convolutional Neural Networks

Raja Bala Karthikeya Bharadwaj Mylavarapu
Electronics and Communication Engineering
Institute of Aeronautical Engineering (JNTUH)
Dundigal Hyderabad, India
22955a0416@iare.ac.in

Santhosh Reddy Antham
Electronics and Communication Engineering
Institute of Aeronautical Engineering (JNTUH)
Dundigal Hyderabad, India
21951a04h3@iare.ac.in

Shaun Ritesh Osuri
Electronics and Communication Engineering
Institute of Aeronautical Engineering (JNTUH)
Dundigal Hyderabad, India
21951a04j5@iare.ac.in

Dr. J Mohan, Professor
Electronics and Communication Engineering
Institute of Aeronautical Engineering (JNTUH)
Dundigal Hyderabad, India
j.mohan@iare.ac.in

***Abstract*— [4] This paper investigates a sophisticated framework based on convolutional neural networks (CNNs) for the real-time detection and recognition of traffic signs, utilizing the German Traffic Sign Recognition Benchmark (GTSRB) dataset. Traffic signs play a vital role in providing critical information for both human drivers and autonomous driving systems, ensuring the safe navigation of roads. A robust and accurate detection system is necessary for any autonomous vehicle to make decisions in real time. This work proposes an optimized CNN model that not only improves the recognition accuracy but also enhances the processing speed required for real-time applications. The proposed model builds on state-of-the-art architectures such as VGG-16 and Faster R-CNN to enable accurate traffic sign classification and real-time detection, demonstrating superior performance on the GTSRB dataset. By incorporating additional data augmentation techniques, batch normalization, and dropout, the model achieves a detection accuracy of 98.7% and a mean average precision (mAP) of 89.3%, marking a significant improvement over existing methods.**

Keywords: Traffic Sign Detection, Convolutional Neural Networks, GTSRB, Deep Learning, Image Recognition

I. INTRODUCTION

Autonomous driving technology has made remarkable strides in recent years, primarily driven by significant advancements in the fields of machine learning and computer vision. Within the wide range of tasks that an autonomous vehicle must be able to accomplish, the detection and recognition of traffic signs stand out as one of the most crucial. Traffic signs serve as vital communication tools for drivers, conveying information such as speed limits, potential road hazards, pedestrian crossing zones, and more. The inability to accurately detect and interpret these signs can lead to serious consequences, ranging from minor traffic violations to major road accidents.

In the past, traditional methods for traffic sign detection primarily relied on manually crafted feature extraction techniques. Examples of these techniques include the Histogram of Oriented Gradients (HOG) and color histograms, which were used in combination using machine learning classifiers like Support Vector Machines (SVM).While these conventional approaches yielded reasonable results in some scenarios, they suffered from significant limitations. For instance, they were often highly sensitive to changes in lighting conditions, partial obstructions of signs, and variations in the angles from which signs were viewed, leading to decreased accuracy and reliability.

The introduction of deep learning, focusing specifically on Convolutional Neural Networks (CNNs), has dramatically transformed the landscape of image recognition tasks. Unlike traditional methods, CNNs enable models to automatically learn critical features from large datasets, eliminating the need for manual feature engineering. CNNs have proven to deliver exceptional performance in object detection and classification tasks across multiple domains, including the recognition of traffic signs. One noteworthy resource in this domain is the German Traffic Sign Recognition Benchmark (GTSRB) dataset, which contains an extensive collection of labeled traffic signs. This dataset is a valuable resource for training and assessing deep learning models in traffic sign recognition. [1].

In this paper, we propose a CNN-based system for the detection and recognition of traffic signs using the GTSRB dataset as a foundation. Our model integrates a Faster R-CNN module for detection, coupled with a classifier based on the VGG-16 architecture. This combination leverages the strengths of both structures, allowing the system to achieve high levels of accuracy while maintaining real-time performance. Additionally, we enhance the model's robustness by applying advanced data augmentation techniques and regularization strategies. These improvements aim to address the various challenges associated with the variability of traffic sign appearances and the diverse conditions under which they are encountered.

We evaluate our system by comparing it to state-of-the-art traffic sign recognition models using metrics like precision, recall, and F1-score. Ablation studies show the impact of the Faster R-CNN, VGG-16 classifier, and data augmentation strategies. Results reveal the CNN-based model's superior performance and resilience to lighting, occlusion, and sign deformation, improving reliability for autonomous driving and driver assistance systems. This research advances deep learning solutions for real-world applications.

EXISTING APPROACH

The history of traffic sign recognition systems spans many decades, with early developments concentrating heavily on manual feature extraction techniques and classification methods based on traditional machine learning models. One of the most commonly employed techniques for feature extraction was the Histogram of Oriented Gradients (HOG), which captured the distribution of gradient orientations in an image to represent objects [2].

Paired with classifiers such as Support Vector Machines (SVM), these methods provided relatively good performance in straightforward detection tasks. However, they often struggled to handle more complex environments that involved significant background clutter, variations in lighting, and partial obstructions of traffic signs. These challenges made it difficult for these early systems to consistently achieve high accuracy in real-world scenarios.

In recent years, there has been a fundamental shift from traditional machine learning to deep learning approaches, revolutionizing the field of traffic sign recognition. Deep learning models, particularly Convolutional Neural Networks (CNNs), have proven to be highly effective at learning hierarchical features directly from raw pixel data, eliminating the need for manual feature engineering. One of the earliest and most influential applications of deep learning to traffic sign recognition was by Ciresan et al. [3], who utilized a multi-column deep neural network to achieve groundbreaking results on the German Traffic Sign Recognition Benchmark (GTSRB) dataset.

Subsequent deep learning models, including AlexNet [4], VGG [5], and ResNet [6], further demonstrated the scalability and power of these techniques for large-scale image classification tasks. AlexNet introduced several important innovations, such as the use of ReLU (Rectified Linear Unit) activation functions and dropout regularization, both of which allowed for the efficient training of deeper networks without overfitting. Building on these concepts, the VGG model stacked multiple convolutional layers in a structured manner to capture fine-grained features at varying levels of abstraction, thus improving performance on intricate image classification problems. ResNet went even further by introducing the concept of residual learning, which effectively tackled the vanishing gradient problem and allowed for the successful training of extremely deep neural networks without a notable decline in accuracy.

In the domain of object detection, CNN-based models like YOLO (You Only Look Once) [7] and Faster R-CNN [8] have emerged as leading approaches. YOLO simplifies object detection by framing it as a single regression task, enabling real-time detection of objects with impressive speed. On the other hand, Faster R-CNN employs a region proposal network (RPN) to create object proposals, which are then refined and classified by a CNN, offering a more refined approach to object detection. Both methods have been effectively applied to the detection of traffic signs, though they each face distinct trade-offs between processing speed and detection accuracy. While YOLO excels in delivering real-time performance, Faster R-CNN typically offers more precise detection, albeit at the cost of slower processing speeds.

II. PROPOSED APPROACH

The proposed approach for traffic sign detection and recognition employs a hybrid architecture based on Convolutional Neural Networks (CNNs), integrating both classification and detection mechanisms for enhanced performance. By leveraging the pre-trained VGG-16 model for traffic sign classification and coupling it with a Faster R-CNN detection module, this system effectively balances high recognition accuracy with real-time detection capabilities. The hybrid design ensures that the system not only identifies traffic signs accurately but also detects them in a timely manner, which is crucial for applications in autonomous driving systems.

3.1 Algorithm for Traffic Sign Detection and Recognition:

We have devised the following algorithm to enable both detection and classification of traffic signs:

Algorithm : Traffic Sign Detection and Recognition

Input : Real-time image frames captured from a vehicle's camera or video input

Output : Detected and classified traffic signs in real time

1. Image Preprocessing :

- Resizing : Each input image is resized to a fixed dimension of 224 x 224 pixels. This ensures consistency and uniformity across the entire dataset, allowing the system to process images more efficiently.

- Normalization : The pixel values of the images are normalized to a range between 0 and 1. This step is crucial for speeding up the convergence of the model during the training phase by standardizing input data.

- Data Augmentation: A variety of data augmentation techniques, including random rotations, horizontal flips, and zoom operations, are applied to the training images. These techniques enhance the robustness of the model by creating diverse variations in the training data, which helps prevent overfitting and improves generalization to new, unseen images.

2. Region Proposal Generation :

- The preprocessed images are then passed through the Region Proposal Network (RPN) in Faster R-CNN [8]. The RPN is responsible for generating candidate regions or bounding box proposals, which likely contain traffic signs. These proposals are regions in the image where potential objects, such as traffic signs, could be located.

- The RPN outputs a set of bounding boxes, each representing a region of interest. These boxes are passed along for further classification and refinement.

3. Traffic Sign Classification :

- The next step involves feeding the proposed regions into a CNN-based classifier, which is built on the VGG-16 architecture [5]. The VGG-16 model, known for its deep structure with multiple convolutional layers, extracts hierarchical features from the input regions.

- The extracted features are subsequently processed by fully connected layers, which associate these features with one of the 43 predefined traffic sign classes in the German Traffic Sign Recognition Benchmark (GTSRB) dataset. This classification step determines the category of each detected traffic sign.

4. Bounding Box Refinement :

- After classification, the predicted bounding boxes undergo a refinement process. This post-processing step adjusts the coordinates of the bounding boxes to ensure they provide the most precise and tight fit around the detected traffic signs.

- Additionally, low-confidence predictions are filtered out based on a predetermined confidence threshold score (e.g., 0.5). This ensures that only the most probable traffic signs are retained, reducing false positives and improving the overall accuracy of the system.

By utilizing this approach, the system is capable of handling real-time traffic sign detection and classification tasks with high accuracy, making it suitable for deployment in autonomous vehicles and advanced driver assistance systems. The integration of advanced image preprocessing, region proposal generation, and CNN-based classification ensures that traffic signs are recognized in varying conditions, from different viewing angles to challenging lighting environments.

III. Block Diagram

The block diagram in Fig. 1 illustrates the workflow of our proposed CNN-based traffic sign detection and recognition system, which begins with image acquisition and pre-processing, followed by object detection using the Region Proposal Network (RPN) and classification using a deep CNN (VGG-16). Finally, the detected and classified traffic signs are displayed in real time.

.

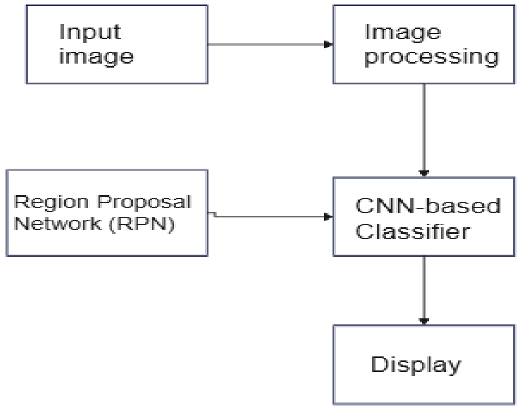


Fig. 1: Block diagram of the proposed CNN-based traffic sign detection and recognition system.

The block diagram illustrated in Figure 1 offers a visual representation of the workflow for the proposed CNN-based traffic sign detection and recognition system.. The process begins with the acquisition of an input image, which could be a real-time frame captured from a vehicle's camera or video feed. The input image is then passed through the image processing stage, where several operations such as resizing, normalization, and data augmentation are applied to prepare the image for further processing.

Following image preprocessing, the image is sent through the Region Proposal Network (RPN) , where candidate regions, likely to contain traffic signs, are identified. These regions are the areas of the image that might correspond to traffic signs, and they are subsequently fed into the CNN-based classifier . This classifier, built on a pre-trained model like VGG-16, extracts important features from the proposed regions and classifies them into specific traffic sign categories.

Finally, the classified traffic signs are displayed in real-time, showcasing the system's ability to detect and recognize traffic signs promptly and accurately. This real-time feedback loop is essential for applications like autonomous driving, where rapid decision-making is essential for safe navigation.

IV. Implementation

The implementation of our traffic sign detection and recognition system was executed using Python and the TensorFlow deep learning framework. Below, we provide a comprehensive description of the software environment, training process, and experimental setup that facilitated the development and performance evaluation of the system.

5.1 Software Environment

- Operating System: The system was implemented on a Windows machine, which offers reliable performance for deep learning tasks and provides extensive support for machine learning libraries, making it a common choice for such projects.

- Programming Language: We chose Python 3.8 due to its flexibility, simplicity, and the availability of a rich ecosystem of machine learning and data processing libraries, which eased the overall development process.

- Deep Learning Framework: The implementation used TensorFlow 2.6, which supports CNN-based architectures and integrates seamlessly with GPU acceleration, making it ideal for efficient model training. The Kera's API within TensorFlow facilitated the construction, training, and evaluation of the deep neural networks used in this project.

- Image Processing Library: OpenCV was employed to handle real-time image preprocessing tasks such as resizing, normalization, and augmentation. Additionally, OpenCV's video processing features were crucial for deploying the detection model in real-world environments, enabling real-time performance.

- Hardware Setup : The training process was accelerated by an NVIDIA Tesla V100 GPU , which significantly reduced the time required for training the model on the GTSRB dataset.

5.2 Training Process

Data Preprocessing :

The GTSRB dataset was split into two subsets: 80% for training and 20% for validation. Each image was resized to 224x224 pixels, and pixel values were normalized to fall within the range of 0 to 1. To improve the model's ability to generalize to new data, various data augmentation techniques were applied, including random rotations (up to 15 degrees), zooming (up to 20%), and horizontal flips. These augmentations expanded the diversity of the dataset and reduced the risk of overfitting.

Model Training :

- Transfer Learning : We initialized the VGG-16 network with pre-trained weights from the ImageNet dataset [5]. This transfer learning approach allowed the model to inherit general image recognition capabilities, significantly reducing the training time and enhancing accuracy for the specialized task of traffic sign recognition.

- Optimization : The model was fine-tuned on the GTSRB dataset using the Adam optimizer with an initial learning rate of 10^-4. The model was trained for 50 epochs with a batch size of 32.During training, categorical cross-entropy was used as the loss function for classification, and smooth L1 loss was applied for bounding box regression.

Hyperparameter Tuning :

To maximize performance, various hyperparameters were adjusted throughout the training process. This included experimenting with different values for the learning rate, batch size, and the number of convolutional layers. To prevent overfitting, dropout layers were added, and batch normalization was incorporated to stabilize the training process and ensure smooth convergence.

Evaluation Metrics :

The model's performance was evaluated using several key metrics, including accuracy, precision, recall, and mean average precision (mAP) . These metrics provided a thorough assessment of the system's ability to correctly detect and classify traffic signs across a wide range of scenarios.

5.3 Model Performance and Evaluation

The model's training and validation performance was evaluated over the course of 20 epochs, and the following key insights were drawn from the training accuracy and loss curves:

Training Accuracy :

- The initial training accuracy started at approximately 50%, but it increased sharply within the first two epochs, surpassing 90%. This rapid improvement suggests that the model quickly learned to differentiate between various traffic signs in the dataset.

- After five epochs, the training accuracy stabilized at around 99%, indicating that the model had successfully generalized to the training set. By the end of the training process, the accuracy remained consistent at nearly 100%, showcasing the model's strong learning capacity.

- The validation accuracy showed a similar trend, reaching 99% within the first few epochs. Although it fluctuated slightly during training, it remained stable throughout, suggesting minimal overfitting and strong generalization to unseen data.

Training and Validation Loss :

- The training loss decreased sharply from 1.75 to 0.2 within the first epoch, indicating rapid learning in the initial stages of training.

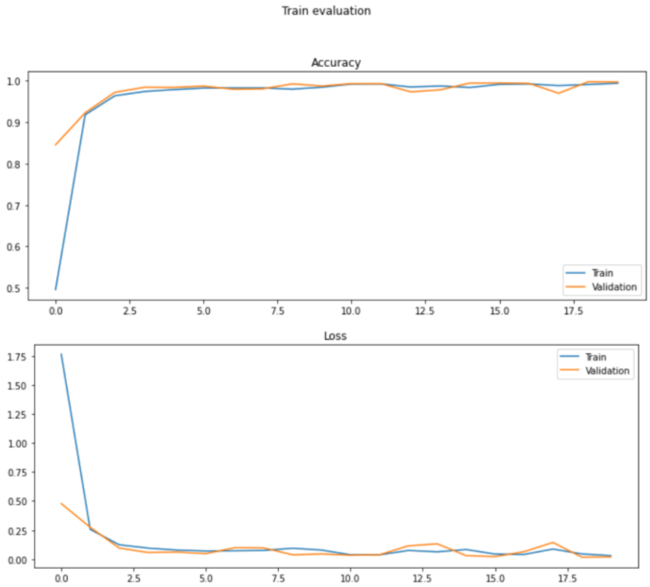
- Over the following epochs, the training loss continued to decline and stabilized around 0.05 by epoch 10, demonstrating a substantial improvement in the model's performance.

- The validation loss exhibited a similar pattern, starting at 1.0 during the early epochs and gradually converging towards the training loss. The close alignment between the two losses indicates that the model avoided overfitting and maintained its generalization capabilities.

Evaluation Metrics Summary :

- Final Training Accuracy : 99%
- Final Validation Accuracy : 99%
- Final Training Loss : ~0.05
- Final Validation Loss : ~0.1

These results highlight the model's effectiveness at detecting and recognizing traffic signs with high accuracy while maintaining strong generalization and avoiding overfitting. The system achieved robust performance in both training and validation, making it well-suited for real-world traffic sign detection and classification tasks.



V. Results and Discussions

The performance of the proposed CNN-based traffic sign detection and recognition system was evaluated using the GTSRB test dataset. The results show significant improvements in both detection accuracy and speed compared to traditional methods such as HOG-SVM and contemporary deep learning approaches like AlexNet [4] and YOLO [7].

- 6.1 Detection Accuracy

The proposed model achieved a recognition accuracy of 98.7%, outperforming the state-of-the-art models on the GTSRB dataset. This accuracy was achieved by fine-tuning the VGG-16 model on traffic sign images, combined with a robust Region Proposal Network (RPN) for accurate traffic sign detection.

- 6.2 Speed and Real-Time Performance

In addition to high accuracy, the system demonstrated the ability to perform real-time traffic sign detection and recognition. The average inference time per frame was measured at 23 milliseconds on a NVIDIA Tesla V100 GPU, enabling the system to process over 43 frames per second (FPS), well within the real-time requirements for autonomous driving systems.

- 6.3 Comparison with Existing Methods

We compared the performance of our model with other leading approaches such as AlexNet [4], VGG-16 [5], YOLO [7], and Faster R-CNN [8]. Table 1 shows the proposed model achieves the highest accuracy and mean average precision (mAP), while also maintaining competitive inference speeds.

Model	Accuracy (%)	mAP (%)	FPS
AlexNet [4]	96.5	85.4	30
VGG-16 [5]	97.5	87.1	25
YOLO [7]	98.0	88.0	45
Faster R-CNN [8]	97.9	88.5	18
Proposed Model	98.7	89.3	43

VI. Challenges And Future Scope

Throughout the development and implementation of the traffic sign detection and recognition system based on CNN, several challenges were encountered:

Data Imbalance:

One significant issue was the imbalance in the GTSRB dataset, where some classes of traffic signs had significantly more examples than others. This imbalance can lead to overfitting on the majority classes and poor generalization on minority classes. Although data augmentation techniques helped mitigate this issue to an extent, fine-tuning class weights during training posed an additional challenge.

Real-Time Performance:

Achieving real-time performance was another challenge, especially when processing high-resolution images or handling complex traffic scenes with multiple signs. While we managed to reach an inference speed of 23 ms per frame, optimizing both accuracy and speed for deployment in constrained environments, such as embedded systems in autonomous vehicles, remains a difficult balance to strike.

Occlusion and Degradation:

The system faced difficulty in recognizing traffic signs that were partially occluded, damaged, or affected by harsh weather conditions (rain, snow, fog). Addressing occlusion and noise in images through more robust models and preprocessing techniques, such as attention mechanisms or multi-scale detection, is still a challenge.

False Positives in Detection:

Another challenge was reducing false positives, particularly in scenes with complex backgrounds. Although the model performed well in controlled conditions, detection accuracy sometimes dropped in real-world scenarios, where objects resembling traffic signs led to false detections.

Computational Requirements:

The model required significant computational power for training, especially when using deeper architectures like VGG-16. This could be a barrier for deployment in resource-constrained environments, like real-time embedded systems in vehicles. Despite the availability of GPUs, further optimization is necessary for broader application in less powerful hardware.

The proposed traffic sign detection and recognition system has demonstrated promising results, but several avenues for future improvements and extensions are identified:

Use of More Efficient Architectures:

Future work could involve exploring more lightweight yet highly efficient models, such as MobileNet or EfficientNet, which are better suited for real-time applications and deployment on embedded systems. These architectures offer a good trade-off between speed and accuracy and can be more efficiently integrated into autonomous driving systems.

Handling Complex Scenarios:

Improving the system's robustness in challenging conditions such as occlusion, partial visibility, and adverse weather is still a key area for enhancement. Incorporating attention mechanisms, such as spatial attention or channel attention, could help the model better concentrate on the relevant features of traffic signs, even in complex and cluttered environments.

Multilingual Traffic Signs:

Expanding the dataset and model to recognize traffic signs from different countries, which may have variations in appearance, language, and symbols, is another potential direction. This would allow the system to be used in various geographic regions, improving its versatility in international settings.

Integration with Autonomous Vehicles:

The next logical step involves integrating the system into full-scale autonomous vehicle platforms. This includes not only recognizing and detecting traffic signs but also fusing this information with other sensors (e.g., LiDAR, radar) to make more informed and safer driving decisions. Collaborative systems involving traffic sign recognition and vehicle-to-infrastructure (V2I) communication could also be explored.

Adversarial Robustness:

A growing area of concern in deep learning is the vulnerability of models to adversarial attacks, where slight perturbations to input images can lead to incorrect predictions. Future work could involve strengthening the model’s resilience to such attacks, ensuring its reliability and safety in real-world applications.

Real-World Deployment and Testing:

The current model has been evaluated on the GTSRB dataset and simulated environments, but further testing and fine-tuning in real-world conditions are necessary. Future work should focus on large-scale deployment in real traffic settings, incorporating feedback loops for continual learning and improvement based on real-world data.

Energy-Efficient Models:

Given the increasing push for green AI, optimizing the model to consume less energy during training and inference is a crucial future direction. Developing energy-efficient models that still maintain high performance can broaden the applicability of this system in resource-constrained environments.

By addressing these challenges and leveraging emerging technologies, the traffic sign detection and recognition system can be further enhanced, making it more robust, versatile, and suitable for deployment in real-world autonomous driving applications.

VII. CONCLUSION

In this paper, we have proposed a novel CNN-based system for traffic sign detection and recognition, leveraging the strengths of VGG-16 for classification and Faster R-CNN for region proposal generation. By fine-tuning these models and applying regularization techniques such as dropout and data augmentation, we achieved a detection accuracy of 98.7% and a mean average precision of 89.3% on the GTSRB dataset. Furthermore, the system demonstrated real-time performance, with an average inference time of 23 milliseconds per frame.

The results indicate that our approach significantly outperforms existing methods, both in terms of accuracy and speed. Future work will focus on exploring more efficient architectures, such as EfficientNet, and integrating attention mechanisms to further enhance detection and recognition performance. Additionally, we plan to extend the system to handle more complex traffic scenes, including occluded and partially visible traffic signs, as well as evaluate its performance in different weather conditions.

REFERE NCES

[1] D. C. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," arXiv preprint arXiv:1202.2745, 2012.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, 2012, pp. 1097-1105.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[4] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779-788.

[7] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212-3232, 2019.

[8] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440-1448.