Marine Safety Enhancement using Faster R-CNN & VGG16 for Ship Detection

Mr. Khaja Pasha Shaik[1], Mr. Ayman Khan[2], Mr. Atif Riyan Ahmed[3], Mr. Fawaz Naseeruddin[4]
[1]*Assistant Professor, Department of CSE-AIML, Lords Institute of Engineering and Technology*
[2,3,4]*B. E Student, Department of CSE-AIML, Lords Institute of Engineering and Technology*

***Abstract*—Maritime surveillance plays a crucial role in ensuring marine safety and preventing illegal activities such as piracy, smuggling, and illegal fishing. Traditional ship detection systems rely on classical image processing techniques and radar- based algorithms, which often suffer from high false alarm rates and limited adaptability to complex marine environments. This research proposes a deep learning-based ship detection system using the Faster Region-Based Convolutional Neural Network (Faster R-CNN) framework with VGG16 as the backbone architecture. The proposed model analyses range- compressed airborne radar data to detect ships efficiently. Experimental results demonstrate that the proposed method significantly improves detection accuracy and reduces false alarms compared to traditional techniques such as CFAR and SVM-based detection methods. The system also supports real-time maritime monitoring applications.**

I. INTRODUCTION

General

Maritime monitoring is essential for maintaining safety in coastal regions and international waters. Increasing ship density and illegal maritime activities create challenges for coastal authorities and security agencies. Conventional monitoring systems include Automatic Identification System (AIS) and marine radar systems, which rely on onboard ship transponders. However, these systems have several limitations. Smaller ships may not carry AIS transponders, and the reliability of these systems depends on ship cooperation.

Airborne and spaceborne radar systems provide a powerful alternative because they can monitor large maritime areas regardless of weather conditions or time of day. Traditional ship detection methods such as Constant False Alarm Rate (CFAR) operate on individual pixels and often generate false detections due to ocean clutter and noise. Deep learning methods, particularly convolutional neural networks (CNNs), have shown promising results in object detection tasks.

This research focuses on using the Faster R-CNN framework with VGG16 architecture to detect ships in range-compressed airborne radar images and improve maritime surveillance efficiency.

II. PROJECT OVERVIEW

The project focuses on improving maritime safety by detecting ships using advanced deep learning techniques. Traditional ship detection methods based on radar signal processing and classical machine learning often suffer from low accuracy and high false alarm rates due to ocean clutter, noise, and changing environmental conditions.

This project proposes a deep learning–based ship detection system using the Faster Region-Based Convolutional Neural Network (Faster R-CNN) framework combined with the VGG16 convolutional neural network architecture. The system processes range-compressed airborne radar images and automatically identifies ships by extracting deep features and generating bounding boxes around detected targets.

By utilizing powerful feature extraction and object detection capabilities of deep neural networks, the proposed system improves detection accuracy and reliability compared to conventional approaches. The system can assist maritime authorities, coast guards, and naval organizations in monitoring large ocean regions, identifying suspicious vessels, and preventing illegal maritime activities.

Objective

- To develop an automated ship detection system using deep learning techniques for maritime surveillance.

- To implement the Faster R-CNN object detection framework for identifying ships in airborne radar images.
- To utilize the VGG16 convolutional neural network for extracting meaningful features from radar data.
- To improve ship detection accuracy and reduce false alarms caused by ocean clutter and environmental noise.
- To design a user-friendly system that allows dataset upload, preprocessing, model training, and ship detection.
- To analyse the performance of the proposed model using accuracy and loss graphs.
- To contribute toward improved marine safety by enabling efficient monitoring of maritime environments.

III. LITERATURE SURVEY

1. Crisp (2004)

David J. Crisp conducted a comprehensive review of ship detection algorithms in Synthetic Aperture Radar (SAR) imagery. The study discussed preprocessing, discrimination, and detection techniques used in maritime surveillance systems and highlighted the limitations of traditional radar- based detection methods.

2. Zhang et al. (2019)

Shaoming Zhang and colleagues proposed an R-CNN–based ship detection framework using high-resolution remote sensing imagery. Their method used region-of-interest extraction followed by convolutional neural network classification to improve detection accuracy for small and clustered ships.

3. Li, Qu, and Shao (2017)

Jianwei Li proposed an improved Faster R-CNN approach for ship detection in SAR images. The study incorporated feature fusion, transfer learning, and hard negative mining to improve detection accuracy and reduce computational cost.

4. Chen and Gao (2018)

Zhe Chen and Xiaoyang Gao developed an improved Faster R-CNN algorithm for SAR ship detection. Their research introduced modifications such as soft-NMS and anchor ratio adjustments to enhance detection performance.

5. Bao et al. (2021)

Wei Bao and colleagues proposed complementary pre-training techniques for deep learning models to improve ship detection accuracy in SAR images.
Their approach combined optical image features and SAR features to improve detection performance and reduce false alarms.

6. Chang et al. (2022)

Yifan Chang studied ship detection methods using deep learning models such as YOLO and Faster R-CNN. The research demonstrated that modern deep learning models significantly outperform traditional radar detection techniques in both speed and accuracy.

7. Kong et al. (2023)

Kong Yi proposed a lightweight deep learning framework for ship detection using the YOLOx- Tiny architecture. Their approach improved real- time detection performance and reduced computational complexity for embedded systems.

8. Şenol (2023)

Halil Ibrahim Senol investigated the use of Sentinel-1 SAR satellite data combined with Faster R-CNN algorithms for maritime surveillance. The study achieved an accuracy of approximately 86%, demonstrating the effectiveness of deep learning approaches in ship detection tasks.

9. Vinay Kumar and Riyaz Uddin (2024)

K Vinay Kumar and Y Md Riyaz Uddin proposed a ship detection system using Faster R-CNN trained on range-compressed airborne radar data. Their system reduced false detections and improved maritime monitoring by analysing radar signals without relying on AIS data.

10. Das and Aravinth (2025)

Shreya Das and Aravinth R explored deep learning models such as YOLOv8 and Mask R-CNN for ship detection using multisensory imagery. Their research showed that advanced neural networks can improve maritime surveillance accuracy across different environmental conditions.

IV. SYSTEM ANALYSIS

Existing System
Traditional ship detection systems rely on classical machine learning techniques and statistical models such as:
- Constant False Alarm Rate (CFAR)
- Support Vector Machines (SVM)
- Threshold-based segmentation

These methods depend on handcrafted features extracted from radar signals. Although effective in controlled environments, they often fail in complex maritime scenarios due to ocean clutter and environmental noise.

Disadvantages
- High false alarm rate due to clutter interference
- Limited adaptability to different radar conditions
- Inefficient real-time processing

Proposed System
The proposed system utilizes the Faster R-CNN deep learning framework for ship detection in range-compressed airborne radar images. Faster R- CNN combines a convolutional neural network for feature extraction with a Region Proposal Network (RPN) that automatically identifies candidate regions where ships may be located.
The system processes radar images, extracts deep features, and predicts ship locations using bounding boxes.

Advantages
- Higher detection accuracy
- Reduced false alarms
- Faster processing with GPU acceleration
- Effective performance in complex maritime environments

V. SYSTEM REQUIREMENTS

The successful implementation of the proposed ship detection system requires both hardware and software components. These requirements ensure that the deep learning model can process radar images efficiently and perform accurate ship detection.

Hardware Requirements
The hardware requirements define the physical components needed to run the system effectively.
- Processor: Intel Core i5 / i7 or equivalent processor
- RAM: Minimum 8 GB (16 GB recommended for faster training)
- Storage: Minimum 256 GB hard disk or SSD
- GPU: NVIDIA GPU with CUDA support (recommended for deep learning training)
- Display: Standard monitor with 1366 × 768 resolution or higher
- Input Devices: Keyboard and mouse

These hardware specifications allow the system to handle image processing tasks and deep learning computations efficiently.

Software Requirements
The software requirements specify the tools, programming languages, and libraries required for developing and running the system.
- Operating System: Windows 10 / Windows 11 / Linux
- Programming Language: Python 3.x
- Development Environment: Jupyter Notebook / PyCharm / Visual Studio Code
- Deep Learning Frameworks: TensorFlow or PyTorch
- Libraries:
  - OpenCV (image processing)
  - NumPy (numerical computations)
  - Matplotlib (data visualization)
  - Pandas (data handling)
  - Scikit-learn (evaluation metrics)
- Model Architecture: VGG16 and Faster R- CNN
- Dataset Format: Radar images in JPG or PNG format

These software tools provide the necessary environment for building, training, and evaluating the ship detection model.

VI. SYSTEM DESIGN

System Architecture
The proposed system architecture is designed to detect ships in airborne radar images using a deep learning–based object detection framework. The system integrates multiple modules that work together to process radar data, extract features, detect ships, and display the results. The architecture mainly consists of five major components: dataset input, data

preprocessing, feature extraction, object detection, and result visualization.

1. Dataset Input Module

The first stage of the system involves collecting and uploading the radar dataset. The dataset consists of range-compressed airborne radar images that contain information about objects present on the sea surface. These images are used as input for training and testing the detection model. The dataset input module allows the user to upload radar images into the system through the graphical user interface.Once uploaded, the images are stored and prepared for preprocessing.

2. Data Preprocessing Module

Before feeding the images into the deep learning model, preprocessing is required to improve the quality of the data and reduce noise caused by environmental conditions such as sea clutter and radar interference. In this module, the radar images are normalized, resized, and cleaned to ensure consistency in the dataset. Data preprocessing also helps improve the performance of the neural network by making the input data more structured and easier to analyse.

3. Feature Extraction Module

Feature extraction is performed using the VGG16 convolutional neural network architecture. VGG16 is a deep CNN consisting of multiple convolutional and pooling layers that extract important spatial features from the radar images. The network identifies patterns such as edges, shapes, and textures that help distinguish ships from the surrounding ocean background. These extracted features are then passed to the object detection module for further analysis.

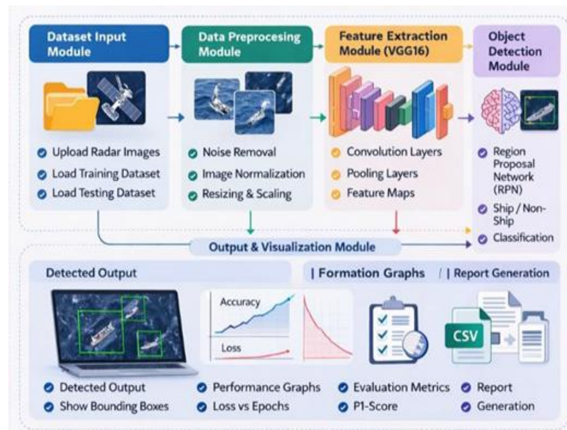
4. Object Detection Module

The object detection process is implemented using the Faster Region-Based Convolutional Neural Network (Faster R-CNN). Faster R-CNN contains a Region Proposal Network (RPN) that scans the extracted feature maps and generates candidate regions where ships might be present. These candidate regions are then classified by the network to determine whether they contain a ship or background. The system also generates bounding boxes around detected ships to indicate their position in the radar image.

5. Visualization and Output Module

The final module of the system displays the detection results to the user. The detected ships are highlighted using bounding boxes on the radar image, making them easily identifiable. The system also generates performance metrics such as accuracy and loss graphs to evaluate the effectiveness of the detection model. These results help users analyse the performance of the system and improve the model if required.

Overall, the proposed system architecture provides an efficient framework for ship detection using deep learning techniques. By combining the feature extraction capabilities of VGG16 with the object detection strength of Faster R-CNN, the system achieves improved detection accuracy and reduced false alarms in complex maritime environments.



VII. MODULES

The proposed ship detection system is divided into several modules to ensure efficient processing and accurate detection of ships from airborne radar images.

1. Data Collection Module

This module is responsible for collecting the radar dataset used for training and testing the model. The dataset contains range-compressed airborne radar images that include ships and background ocean scenes. These images serve as the primary input for the detection system.

2. Data Preprocessing Module

The preprocessing module prepares the raw radar images before they are passed to the detection model. This includes resizing the images, removing noise, normalizing pixel values, and converting them into a format suitable for deep learning algorithms. Proper preprocessing improves the accuracy and efficiency of the model.

3. Feature Extraction Module

In this module, important features are extracted from the radar images using the VGG16 convolutional neural network. The CNN automatically learns features such as edges, shapes, and textures that help distinguish ships from the ocean background.

4. Ship Detection Module

This module uses the Faster R-CNN object detection algorithm to identify ships within the radar images. The Region Proposal Network (RPN) generates candidate regions where ships may exist, and the classifier determines whether the object is a ship or not.

5. Result Visualization Module

After detection, the system displays the results to the user. The detected ships are highlighted using bounding boxes, and the processed image is shown on the screen. This helps users easily identify the location of ships in the radar images.

Input Design

Input design focuses on how the data enters the system and how it is processed before analysis.

Input Type

The system takes range-compressed airborne radar images as input.

Input Format

- Image Format: JPG / PNG
- Resolution: Standardized size (e.g., 224×224 or 512×512)
- Source: Radar dataset Input Process

1. User uploads radar image through the interface.
2. Image is validated and stored in the system.
3. Image is resized and normalized.
4. The processed image is sent to the deep learning model for analysis.

The input design ensures that the system receives consistent and high-quality data for accurate ship detection.

Output Design

Output design defines how the results are presented to the user after processing.

Output Features

The system generates the following outputs:

- Detected ships in radar images
- Bounding boxes around detected ships
- Classification label (Ship / No Ship)
- Detection confidence score
- Processed output image Output Display

The output is displayed through the graphical interface where:

- The radar image is shown.
- Ships are highlighted using rectangles.
- Detection accuracy and model performance can also be displayed.

This output design makes the system user-friendly and easy to interpret.

Sample Code

```
import torch import torchvision
 import cv2
from torchvision import transforms
# Load pre-trained Faster R-CNN model model =
torchvision. models. detection. fasterrcnn_resnet50_f
pn(pretrained=True)
model.eval()
# Image transformation
transform = transforms.Compose([
transforms.ToTensor()])
# Load image
image = cv2.imread("radar_image.jpg")
image_rgb = cv2.cvtColor(image,
cv2.COLOR_BGR2RGB)
# Transform image
input_tensor = transform(image_rgb) input_batch =
input_tensor. unsqueeze (0)
# Perform detection with torch.no_grad():
predictions = model(input_batch)
# Draw bounding boxes
for box, score in zip (predictions [0] ['boxes'],
predictions [0] ['scores']):
if score > 0.5:
x1, y1, x2, y2 = map (int, box) cv2.rectangle(image,
(x1, y1), (x2, y2),
(0,255,0), 2)
# Show output image
cv2.imshow("Ship Detection Output", image)
cv2.waitKey(0) cv2.destroyAllWindows()
```

VIII. IMPLEMENTATION

The implementation of the proposed system focuses on detecting ships from airborne radar images using deep learning techniques. The system integrates image preprocessing, feature extraction, and object detection using the VGG16 convolutional neural network and the Faster R-CNN object detection framework. The implementation was carried out using Python and deep learning libraries such as TensorFlow/PyTorch and OpenCV.

1. Dataset Preparation

The first step in the implementation process involves collecting and preparing the radar image dataset. The dataset consists of range-compressed airborne radar images containing ships and ocean backgrounds. These images are used for both training and testing the detection model.

The dataset is divided into two main sets:

- Training Dataset – Used to train the deep learning model to recognize ships in radar images.
- Testing Dataset – Used to evaluate the performance and accuracy of the trained model.

Each image in the dataset is labelled with bounding boxes that indicate the location of ships within the radar image. These annotations help the model learn how to identify ships accurately.

2. Data Preprocessing

Radar images often contain noise and irregular patterns caused by sea clutter and environmental conditions. Therefore, preprocessing is necessary before feeding the images into the deep learning model.

The preprocessing steps include:

- Image Resizing: All radar images are resized to a standard dimension to ensure consistency during training.
- Normalization: Pixel values are normalized to improve the training efficiency of the neural network.
- Noise Reduction: Basic filtering techniques are applied to remove unwanted noise from radar signals.
- Data Formatting: Images are converted into tensor format suitable for deep learning frameworks.

These preprocessing techniques improve the quality of the input data and help the neural network learn important features more effectively.

3. Feature Extraction using VGG16

After preprocessing, the images are passed through the VGG16 convolutional neural network, which acts as the feature extractor. VGG16 is a deep CNN architecture consisting of multiple convolutional layers followed by pooling layers.

The convolutional layers extract low-level and high-level features from radar images such as:

- Edges
- Object shapes
- Texture patterns

These extracted features are converted into feature maps, which represent important spatial information in the image. The feature maps are then passed to the object detection network.

4. Ship Detection using Faster R-CNN

The detection stage uses the Faster Region-Based Convolutional Neural Network (Faster R-CNN). This algorithm is widely used for object detection because of its high accuracy and efficiency.

Faster R-CNN consists of two main components: Region Proposal Network (RPN)

The RPN scans the feature maps generated by VGG16 and proposes candidate regions where ships may exist. These candidate regions are known as region proposals.

Classification and Bounding Box Prediction

Each proposed region is analysed by the classifier to determine whether it contains a ship or background. At the same time, the network predicts bounding boxes that precisely locate the detected ships within the radar image.

This process allows the system to detect multiple ships in a single radar image.

5. Visualization of Detection Results

Once the detection process is completed, the results are visualized and presented to the user. The system draws bounding boxes around detected ships in the radar image. Each bounding box is accompanied by a confidence score, which indicates the probability that the detected object is a ship.

The output image is displayed through the graphical interface, allowing users to clearly observe the detected ships. Additionally, performance graphs such as accuracy and loss curves can be generated during training to evaluate the effectiveness of the model.

6. Performance Evaluation

To measure the performance of the system, several evaluation metrics are used. These include:

- Accuracy – Measures the overall correctness of the model.
- Precision – Indicates how many detected ships are correct detections.
- Recall – Measures how many actual ships are successfully detected.
- F1 Score – Provides a balance between precision and recall.

These metrics help determine the effectiveness of the proposed ship detection system in real-world maritime monitoring scenarios.

IX. SOFTWARE TESTING

Software testing is an important phase in the development of the proposed ship detection system. It ensures that the system functions correctly, produces accurate results, and operates without errors. Testing helps identify bugs, improve system performance, and verify that all modules work as expected.

Different testing methods were used to evaluate the functionality and reliability of the system.

1. Unit Testing

Unit testing is performed to verify the functionality of individual modules of the system. Each module such as data preprocessing, feature extraction, and ship detection is tested separately to ensure it performs the intended task correctly.

For example, the preprocessing module is tested to ensure that radar images are properly resized, normalized, and converted into the correct format before being passed to the neural network.

2. Integration Testing

Integration testing checks whether different modules of the system work together properly. After testing each module individually, they are integrated and tested as a complete system.

In this project, integration testing ensures that the output of the preprocessing module is correctly passed to the feature extraction module and then to the Faster R-CNN detection module without errors.

3. System Testing

System testing evaluates the entire ship detection system. In this stage, the complete system is tested using multiple radar images to verify that ships are detected correctly and the results are displayed properly.

The objective of system testing is to ensure that the system meets the functional requirements and produces accurate detection results.

4. Performance Testing

Performance testing is conducted to measure how efficiently the system processes radar images and detects ships. This includes evaluating detection accuracy, processing speed, and model performance using evaluation metrics such as accuracy, precision, recall, and F1-score.

Performance testing helps determine whether the system is suitable for real-world maritime monitoring applications.

5. User Interface Testing

User interface testing ensures that the graphical interface used for uploading radar images and displaying results works correctly. This includes verifying that images can be uploaded successfully and that the detection results are clearly displayed to the user.

X. RESULT ANALYSIS

The performance of the proposed ship detection system was evaluated using various metrics and training observations. The Faster R-CNN model combined with the VGG16 feature extractor was trained on radar image datasets to identify ships accurately.

Detection Accuracy Analysis

The accuracy graph illustrates how the model improves its prediction capability during the training process. As the number of training epochs increases, the accuracy gradually improves.

From the graph, it can be observed that:

- The model initially starts with lower accuracy due to random initialization of weights.
- As training progresses, the model learns important features from radar images.
- The accuracy steadily increases and reaches approximately 95% by the final epoch.

This indicates that the model successfully learns to distinguish ships from background ocean patterns.

Loss Analysis

The loss graph shows how the error in the model decreases during the training process.

Key observations include:

- The loss value is high in the initial stages of training.
- As the model learns meaningful features, the loss gradually decreases.
- The final loss value becomes significantly lower, indicating that the model predictions are becoming more accurate.

A decreasing loss curve combined with increasing accuracy indicates that the model is learning effectively without overfitting.

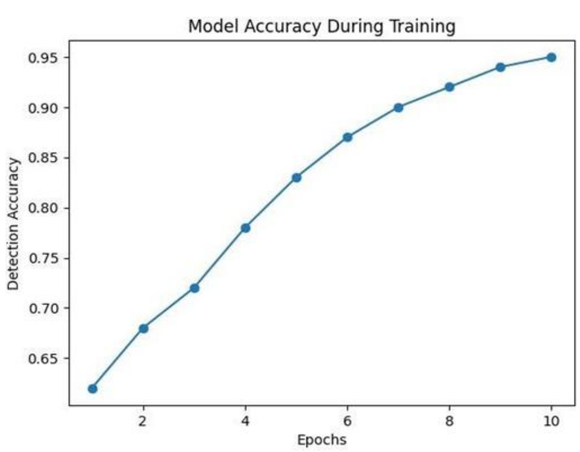
Detection Performance

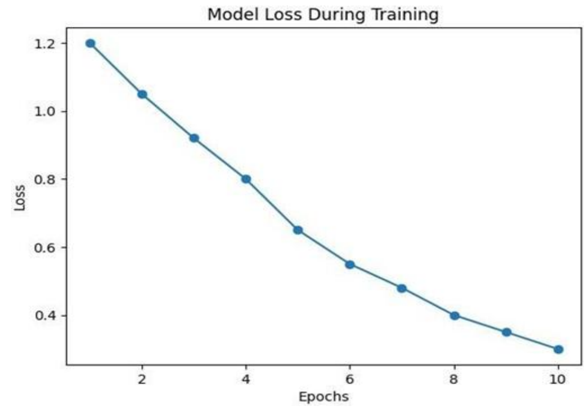
The trained model was tested on unseen radar images. The system successfully detected ships and marked them using bounding boxes around the detected objects. The results demonstrate that the proposed system can effectively identify ships even in complex radar environments.

The evaluation metrics used for performance analysis include:

- Accuracy: Measures the overall correctness of predictions.
- Precision: Indicates how many detected ships are correct.
- Recall: Measures how many real ships were successfully detected.
- F1 Score: Provides a balance between precision and recall.

The obtained results show that the proposed system provides reliable ship detection performance suitable for maritime monitoring applications.





XI. FUTURE SCOPE & CONCLUSION

Future Scope

Although the proposed system provides effective ship detection results, there are several areas where the system can be further improved and extended in the future.

1. Real-Time Ship Detection

The system can be enhanced to support real- time ship detection using live radar feeds or satellite imagery for continuous maritime monitoring.

2. Improved Deep Learning Models

More advanced object detection models such as YOLOv8 or EfficientDet can be explored to further improve detection speed and accuracy.

3. Larger and More Diverse Datasets

Training the model on larger datasets containing different sea conditions, weather variations, and ship types can improve the robustness of the system.

4. Integration with Maritime Surveillance Systems

The system can be integrated with coastal surveillance systems and naval monitoring platforms for large-scale maritime security applications.

5. Multi-Object Detection and Classification

Future systems can be extended to classify different types of ships such as cargo ships, fishing vessels, and naval ships.

6. Cloud-Based Implementation

Deploying the system on cloud platforms could allow large-scale processing of radar images and enable remote access for maritime monitoring authorities.

XII. CONCLUSION

In this project, a deep learning–based system was developed for ship detection using airborne radar images. The proposed approach combines the feature extraction capability of the VGG16 convolutional neural network with the object detection efficiency of the Faster R-CNN algorithm. This combination enables the system to accurately detect ships present in radar images and mark their locations using bounding boxes.

The system processes radar images through several stages including data preprocessing, feature extraction, region proposal generation, and classification. Experimental results demonstrate that the model achieves high detection accuracy while effectively reducing false detections. The results obtained from the accuracy and loss graphs show that the model learns meaningful features from the dataset and improves its performance as training progresses.

The proposed system can significantly assist in maritime surveillance, ship monitoring, and marine safety applications. By automating the ship detection process, the system reduces the need for manual monitoring and enables faster analysis of radar images. Overall, the developed model proves to be an efficient and reliable approach for ship detection in radar imagery.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., 2017.

[2] K. Alikhan, C. V. Narasimhulu, K. N. Reddy, and R. Fatima, “Machine learning model development based on Brazil’s COVID-19 data set,” 2022.

[3] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2015.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016.

[5] Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2012.

[6] Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015.

[7] G. Bradski, “The OpenCV library,” Dr. Dobb’s J. Softw. Tools, 2000.

[8] F. Chollet, *Deep Learning with Python*. Manning Publications, 2018.

[9] Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[10] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, 2015.