Hindawi
Mobile Information Systems
Volume 2023, Article ID 1814906, 11 pages
https://doi.org/10.1155/2023/1814906

*Research Article*

# Satellite Imageries for Detection of Bangladesh's Rural and Urban Areas Using YOLOv5 and CNN

**Mirajul Islam**,[1,2] **Nushrat Jahan Ria**,[1] **and Jannatul Ferdous Ani**[1]

[1]*Department of Computer Science and Engineering, Daffodil International University, Dhaka 1341, Bangladesh*
[2]*Faculty of Graduate Studies, Daffodil International University, Dhaka 1341, Bangladesh*

Correspondence should be addressed to Mirajul Islam; merajul15-9627@diu.edu.bd

Received 7 December 2022; Revised 19 May 2023; Accepted 7 July 2023; Published 17 July 2023

Academic Editor: A. H. Alamoodi



In recent years, there have been significant advancements in object identification in natural photos. However, when applying natural image object recognition techniques directly to satellite images, the results are often unsatisfactory. This is primarily due to inherent disparities in the object scale and orientation caused by the omniscient viewpoint of satellite imagery. The distinguishing factors between rural and urban areas lie in the objects that cover them. Furthermore, the complex backdrop of satellite photos poses challenges in accurately extracting features, leading to the omission of small objects in many regions. The performance of object detection, which is crucial for area identification, is also affected by dense object overlap and occlusion. To address these aforementioned issues, we made modifications to the generalized one-stage detector YOLOv5, specifically tailored for satellite photos. For this research, we manually collected data from Google Earth, meticulously labeling them and subsequently verifying them with human annotators. We then preprocessed the data using computer vision techniques, such as resizing and normalization. Next, we employed YOLOv5 and transfer learning-based CNN architectures of InceptionV3, DenseNet201, and Xception to compare their performances. The goal was to accurately identify rural and urban areas from remote sensing images.

## 1. Introduction

High-resolution satellite imagery is obtained through the utilization of advanced earth satellite technology to observe the surface of our planet. However, the processing of a large volume of satellite photos poses significant challenges for current interpretation algorithms. One of the fundamental tasks in computer vision is object detection, which involves accurately and efficiently identifying predefined objects within images. This capability finds extensive application in areas such as precision farming, urban traffic control, and various other domains [1–3]. The Earth's orbiting fleet of commercial satellites produces an ever-increasing amount of imagery, growing at an exponential rate. Satellite imagery serves a multitude of purposes, including agricultural crop classification [4, 5], scene classification [6, 7], wildlife monitoring [8, 9], forest characterization [10, 11], meteorological analysis [12, 13], infrastructure assessment, building localization [14, 15], and soil moisture estimation [16, 17].

Recent advancements in segmentation and object detection tasks have been significantly facilitated by data-driven deep learning techniques. The size and quality of the training dataset have an impact on detection precision. The development of object detection has been fueled by a number of extensive and difficult natural picture datasets, including PASCAL VOC and MS COCO. Nevertheless, recognizing objects in optical satellite photos remains challenging [18]. The causes are listed as follows. First, satellite photographs taken from a bird's eye view provide a broad imaging range with full information, in contrast to the natural images captured by ground-based cameras with horizontal views. There is an uneven distribution of foreground items and intricate background information in complex landscapes and urban settings [19]. In addition, objects in satellite pictures often exhibit varying visual appearances and optical properties due to a variety of imaging circumstances, such as perspectives, illumination, and occlusion. Finally, smaller objects frequently have less

information about their appearance than larger ones, making it harder to distinguish them from the background or other nearby objects.

To address the aforementioned issues, this research focuses on improving area identification performance in satellite pictures. The detection speed also presents a substantial challenge for the detection algorithm as region detection in satellite images often needs to occur in real time. You only look once (YOLO) neural networks can significantly enhance detection speed by combining object categorization and localization (two-stage) into a one-stage regression problem. To the best of our knowledge, YOLOv5 is the most recent version of YOLO, which demonstrates the best object detection performance on natural photos. This is because YOLOv5 utilizes the path aggregation network (PANet) and the enhanced CSPDarknet53 as the network's neck and backbone, respectively.

It is challenging to directly apply YOLOv5 to satellite photos for area recognition. In this study, we utilized transfer learning-based CNN architectures and made updates to YOLOv5 from three perspectives listed as follows. First, due to excessive downsampling, the deep feature maps fused in the neck of YOLOv5 would lose information about tiny details. To overcome this issue, we implemented a new branch in the shallower network layer to perform the initial detection of each area. This allows us to preserve the feature information to the greatest extent possible. Second, while YOLO net is typically built on a convolutional neural network (CNN), the CNN is primarily effective at capturing local information. However, when processing high-resolution satellite photos, the traditional transformer would incur a square computational cost, despite its ability to compensate for global modeling capability.

The main contribution of this study can be summarized as follows:

(i) We have proposed a deep learning-based method for identifying rural and urban areas using satellite images.

(ii) We generated a dataset that included two classes, namely, rural and urban areas in Bangladesh.

(iii) We conducted a comparative analysis of the same dataset using two techniques: a YOLOv5-based detection technique and a CNN-based classification technique.

The structure of the paper is as follows. Section 2 clarifies the relevant work of several disease classification methods. The method and materials that were used are illustrated in Section 3. The experimental analysis, including performance and results, is depicted in Section 4. Section 5 discusses the article's conclusion.

## 2. Related Work

Significant progress has been made in the field of Satellite Imagery, with several notable research studies that have been explored. Some of these studies are listed as follows.

The deep learning approach by Kadhim and Abed [20] presented practical deep learning-based approaches for satellite image classification, which involved extracting features using four pretrained CNNs. The paper [21] focused on object and facility classification in high-resolution multispectral satellite imagery, utilizing a deep learning system. The system combined CNN predictions with satellite metadata through postprocessing neural networks. In another study [22], the speed and performance of modern object detection algorithms were compared in commercial EO satellite imagery datasets, specifically for oil and gas fracking wells and small cars. Article [23] examined the effective classification of aerial images using their emergency net model while onboarding a UAV for monitoring and responding to emergencies. Pan et al. [24] introduced a paradigm for mapping a Chinese urban village in Guangzhou City using the U-net deep learning architecture. Their findings suggested that combining U-net-based deep learning with high spatial resolution satellite photos can provide valuable building information in complex urban settlements, crucial for urban revitalization. Yoo et al. [25] compared CNN to an RF classifier in order to map the local climate zone, using bitemporal Landsat images.

Other approaches: Yang et al. [26] utilized ensemble projection (EP) to learn semi-supervised features for satellite image classification, especially in scenarios with limited labeled data and a large amount of unlabeled data. Paper [27] focused on classifying specific land cover in satellite images using the biogeography-based optimization approach. Dai and Yang [28] introduced a technique that incorporated visual attention in satellite image classification and addressed the classification task without a learning phase. Li et al. [29] investigated image cropping strategies for object detection, involving the cropping of large aerial images into uniformly sized smaller images. Their density-map guided object recognition network (DMNet) was inspired by the understanding that an image's object density map reveals the distribution of objects in terms of pixel intensity. Rahman et al. [30] employed a hierarchical clustering approach based on five specified spatial criteria to divide the 331 cities of Bangladesh into six classes using remote sensing data. Research [31] demonstrated the usefulness of satellite images in detecting land use and land cover (LULC) analysis, as well as analyzing the coastal dynamics of agriculture in the Bhola region (characterized by dense forests) and the Dhaka region (characterized by dense cities). Mathieu et al. [32] explored the effectiveness of object-based classifications that extract relevant ground features from images using automated image segmentation techniques.

## 3. Materials and Methods

In this section of the article, we will provide a concise summary of the stages involved in data collection, preprocessing, and preparation. The next step is algorithm selection, where we study each model employed in detail. Then, we will discuss the platforms and the key parameters for training and evaluation metrics. Figure 1 provides a visual overview of the steps involved in the classification

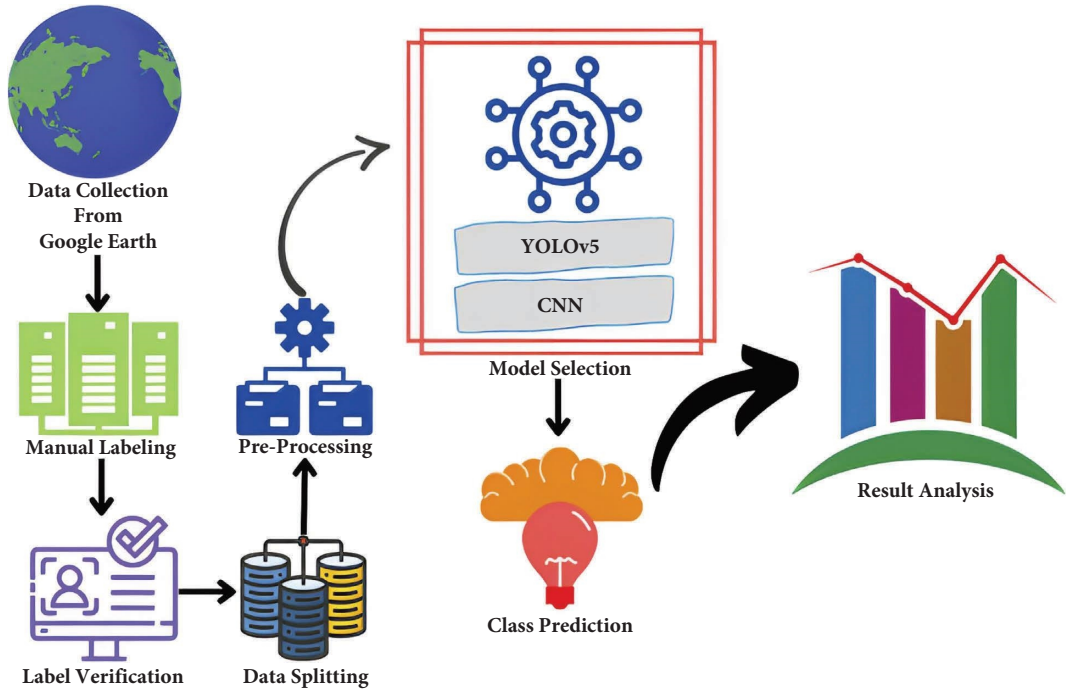


Figure 1: Working flow of the entire classification detection process.

detection process, highlighting the flow of information and the key stages.

*3.1. Dataset Description.* The data collection process involved meticulous manual gathering of satellite images using Google Earth. A total of 3267 satellite images were collected from diverse regions in Bangladesh, with 1631 images representing urban areas and 1636 images representing rural areas. Separate datasets were prepared for CNNs and YOLOv5, as shown in Table 1. For the experiment involving YOLOv5, a subset of 200 images was selected. The data collection process aimed at ensuring a comprehensive representation of the target regions and facilitating accurate analysis and evaluation.

*3.2. Preprocessing.* To enhance the predictive performance of the CNN architecture, the recommended approach used in this research minimizes the number of preprocessing steps. We optimized the training process for CNN models using three standard preprocessing steps.

*3.2.1. Resizing.* Usually, raw collections of images are in different formats, which can lead to imbalanced image features. Technically, the total dataset should be unified into one structure by resizing the image shape. Different sizes of images can be resolved using increasing or decreasing resizing matrix operations. There are two specific solutions for effective performance and reduced complexity metrics. This dataset includes images of various resolutions and sizes. To ensure that all input images have the same dimension, we resized all images to $224 \times 224$ pixels from their original size.

*3.2.2. Normalization.* As a preprocessing step of image normalization, utilizing ImageNet's mean subtraction process, we rescaled the pixel intensity values. We normalized the intensity values of all the images within the range [0, 255] to the standard normal distribution by applying min-max normalization [33] to the intensity range [0, 1], where

$$X\text{norm} = \frac{(X - X\min)}{(X\max - X\min)}, \tag{1}$$

where $x$ denotes pixel intensity. In equation (1), the input image's minimum and maximum intensity values are Xmin and Xmax, respectively.

*3.2.3. Augmentation.* Image augmentation is a technique utilized to expand the available resources within an image by generating nonduplicate regions. It involves applying various transformations to the original image, such as texture reflections, grayscale variations, adjustments in brightness levels, color contrasts, and other relevant image modifications. By introducing bounding boxes during augmentation, the accuracy of object detection can be improved, leading to the creation of synthetic data. Through operations such as image flipping and rotation, the dataset size can be significantly increased, resulting in a larger and more diverse collection of images. This augmentation process contributes to the augmentation of image quantity while preserving the integrity of important regions. In the case of 2D images, factors such as resolution and image quality hold significant importance, particularly when dealing with images that exhibit substantial disparities in size, shape, and color. Synthetic data offer immense potential to exponentially enhance accuracy by generating images that belong to the same category.

Table 1: Dataset description for CNN and YOLOv5.

| Methods | Files | Files | No. of samples |
|---|---|---|---|
| CNN | Train | Urban area | 1305 images |
| | | Rural area | 1309 images |
| | Test | Urban area | 326 images |
| | | Rural area | 327 images |
| YOLOv5 | Images | Train | 160 images (both urban and rural area) |
| | | Val | 40 images (both urban and rural area) |
| | Labels | Train | 160 txt files (both urban and rural area) |
| | | Val | 40 txt files (both urban and rural area) |

For YOLOv5, the dataset contains two types of data files. (1) Raw digital photos, consisting of 200 JPG images in total. (2) Image annotation, consisting of 200 .txt files. These files provide information that specifies the exact locations of the items in the corresponding images that have labels attached to them. Manual annotation was used, and the annotated data was saved in .txt in YOLOv5 format. The images were cautiously labeled using the popular annotation application LabelImg.

*3.3. Selection of Algorithm.* We employed the object detection architecture YOLOv5 and two pretrained CNN models, such as MobileNetV2 and NASNetMobile, for classification and compared their results. In deep learning, large amounts of data are often used to improve the network's ability to predict. Due to the lack of data, we employ the transfer learning [34] approach and pretrain weights from the used models to make the model better at making predictions.

*3.4. YOLOv5.* The network structure diagram of YOLOv5 consists of two main sections. The first section is the main architecture, which includes the input side and the backbone portion. The second section is the detection architecture, comprising the neck and the prediction part [35]. YOLOv5 is trained on the COCO dataset, an object detection model, which contains 80 different classes and a total of 200,000 annotated images. The YOLO family of models, including YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6, and the recent YOLOv7, are widely employed for recognition tasks. The variations in size among the different models of the YOLOv5 family, such as YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, are determined by the width and depth of the BottleneckCSP module [36]. The primary function of the BottleneckCSP module is to extract features from the feature map, enabling the extraction of valuable information from the input image. In this study, the YOLOv5 model summary consisted of 270 layers, 7025023 parameters, 7025023 gradients, and a computational complexity of 16.0 GFLOPs. Figure 2 showcases the architecture of YOLOv5, highlighting its components.

*3.5. Transfer Learning-Based Convolutional Neural Networks (CNNs).* The final step of our work involves classification using transfer learning. Deep convolutional neural networks (DCNNs) have recently attained a state-of-the-art performance in a variety of high-level computer vision tasks. Convolution neural networks (CNNs), more commonly referred to as ConvNets, are a type of feed-forward neural networks that employ a series of convolutional layers, each of which is followed by a pooling layer, to learn to extract features from input data and build a series of high-level feature maps. The proposed CNN-based categorization approach has been evaluated on InceptionV3, DenseNet201, and Xception. The selected architectures' network structures are as follows.

*3.5.1. InceptionV3.* InceptionV3 is a deep convolutional neural network architecture that was introduced by Google. It employs the concept of "inception modules" which consist of parallel convolutional layers with different filter sizes. This allows the network to capture features at multiple scales and resolutions. InceptionV3 is often used for transfer learning due to its strong performance on image classification tasks, as shown in Figure 3. In transfer learning, the pretrained InceptionV3 model is used as a feature extractor, where the initial layers are frozen, and only the final layers are fine-tuned on the target dataset. This enables the model to leverage the learned representations from a large-scale dataset, such as ImageNet, and adapt them to the specific task at hand.

*3.5.2. DenseNet201.* DenseNet201 is a deep convolutional neural network architecture that emphasizes feature reuse and alleviates the vanishing gradient problem. It introduces dense connections between layers, where each layer receives input from all preceding layers. This facilitates the flow of gradients and encourages feature reuse, leading to a better gradient flow and improved information propagation throughout the network. DenseNet201 is commonly used in transfer learning scenarios, where the pretrained model is employed as a feature extractor, as illustrated in Figure 4. By freezing the initial layers and fine-tuning the later layers, DenseNet201 can effectively transfer knowledge from the source dataset to the target task, improving both training efficiency and generalization performance.

*3.5.3. Xception.* Xception, derived from "Extreme Inception," is an architecture that extends the Inception concept further by replacing the standard convolutional layers with depthwise separable convolutions. This factorizes the convolution operation into a depthwise convolution and

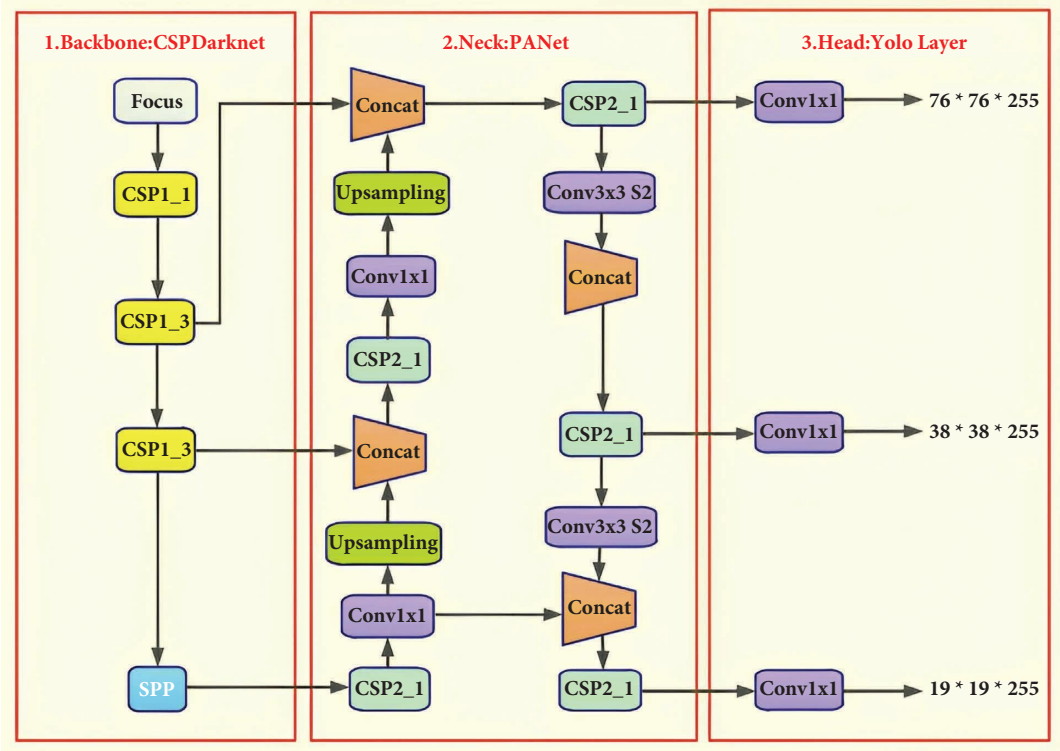


Figure 2: Architecture of YOLOv5 [37].

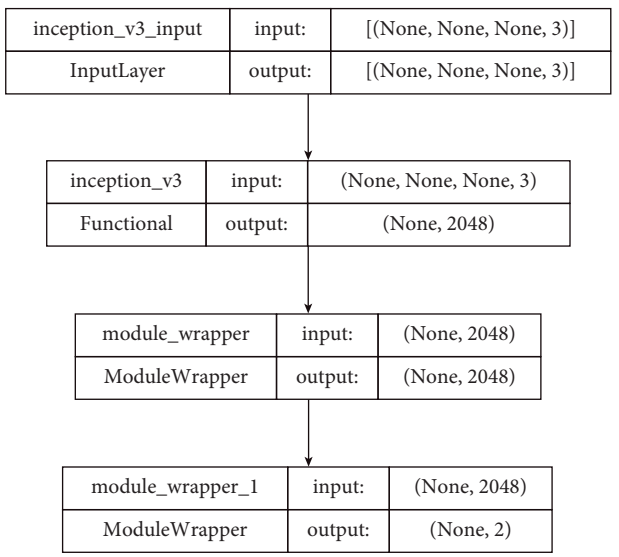


Figure 3: Architecture of InceptionV3.

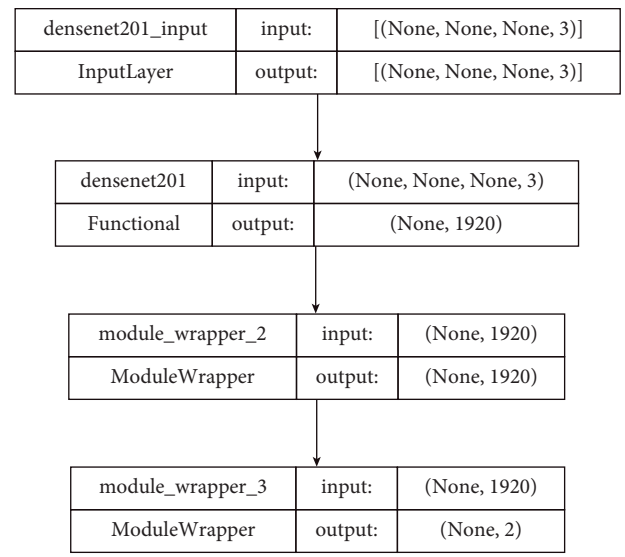


Figure 4: Architecture of DenseNet201.

a pointwise convolution, reducing the computational cost while maintaining expressive power. Xception has shown excellent performance on various image classification benchmarks, as depicted in Figure 5. In transfer learning, Xception is commonly utilized by leveraging its pretrained weights as a feature extractor. The initial layers are frozen, and only the final layers are fine-tuned on the target dataset. This approach allows Xception to transfer high-level features learned from large-scale datasets, enabling effective generalization to new tasks with limited training data.

*3.6. Training Experiment Setup.* This experiment was carried out, and Google Colab was used to train both the YOLOv5 and CNN models, which provides free access to powerful GPUs with no configuration required. For our research, 80% of the images belonging to each class were placed in the training set, while the remaining 20% were placed in the test set.

The size of the image was set at 640 × 640 pixels as part of the YOLOv5 training parameter setting. Throughout the duration of the training procedure, we experimented with

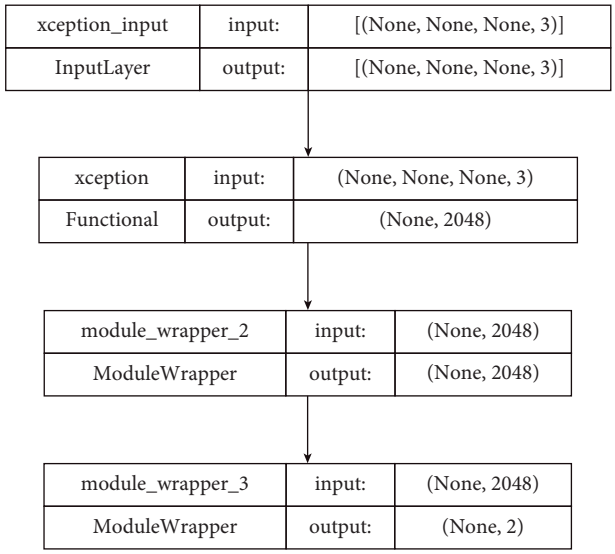


Figure 5: Architecture of Xception.

a large variety of batch sizes and numbers of epochs, all of which featured early stopping conditions. In our trial-and-error experiments, the best results for prediction were obtained with a batch size of 1, a total of 100 epochs, and a learning rate of 0.01. We utilized a notebook invented by Roboflow [38] based on YOLOv5 [39] and employed pretrained COCO weights. The three different types of losses are shown in Figure 6, which are box loss, objectness loss, and classification loss. To determine an algorithm's performance, researchers have used a metric called "box loss," which evaluates how well it can locate an object's center and how completely it predicts a box around that object. Objectness measures the probability that an object exists in the proposed region of interest. Finally, the algorithm's ability to correctly predict an object's class is reflected in its classification loss.

The training parameters for all convolutional neural networks are learning rate $\eta = e - 5$, $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\varepsilon = e - 8$, and decay rate is set to $1e - 5$ for adaptive moment estimation (Adam) optimizer. Activation function Softmax is used which sets a dropout rate of 0.5 to prevent the model from becoming overfit. All models are trained over the duration of 15 epochs, with a batch size of 16.

*3.7. Evaluation Metrics.* To assess the prediction performance of the algorithms in this study, we used highly regarded evaluation metrics such as recall, precision, accuracy, F1-score, and mAP (mean average precision).

The ratio of the number of cases that were correctly classified to the total number of test images is the commonly used measure of accuracy. This can be shown by

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} * 100\%. \quad (2)$$

Precision, often known as a positive predictive value, is defined as the percentage of labels accurately identified in patients who are actually positive and is stated as

$$\text{Precisions} = \frac{\text{TP}}{\text{TP} + \text{FP}} * 100\%. \quad (3)$$

The weighted average of precision and recall, known as the F1-score or F-measure, combines precision and recall. The F-measure is written as

$$F1 - \text{score} = 2 * \frac{\text{Precisions } x \text{ Recall}}{\text{Precisions} + \text{Recall}} * 100\%. \quad (4)$$

The percentage of correctly classified objects is measured by recall or sensitivity. And it is presented as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} * 100\%. \quad (5)$$

The overall intersection over union (IoU) thresholds or the mean average precision across all classes are utilized to determine the mAP value. It is expressed as [40]

$$\text{AP} = \frac{1}{11} \sum \text{recall} \epsilon [0, 0.1, \ldots, 1] * \text{Precision}(r). \quad (6)$$

According to the abovementioned section, the number of correctly predicted cases is referred to as true positives (TPs), while the number of incorrectly predicted cases is referred to as false negatives (FNs), and true negatives (TNs) are the number of negative instances that were correctly predicted. In comparison, the number of mistakenly predicted negative events is known as false positives (FPs).

## 4. Result Analysis and Discussion

After training the YOLOv5 model with our data, we used it to make predictions for images in our test set that had not been seen before. Figure 7 demonstrates how the algorithm can more accurately identify both urban and rural areas.

Table 2 displays the performance of YOLOv5 after training using different measures such as precision, recall, and mAP (mean average precision) when IOU is set to 0.5 (50%) and 0.95 (95%). A validation precision score of 0.995, a recall score of 0.999, and mAP scores of 0.995 and 0.978 for @0.5IOU and @0.95IOU, respectively, were obtained for the YOLO v5 model after evaluation.

Figure 8 presents a collection of images extracted from the test set, illustrating the performance of the Xception model in accurately detecting urban and rural areas. Each image is accompanied by its corresponding actual label (urban or rural) and the target label, along with the associated confidence level. The depicted results highlight the model's ability to classify the regions correctly, as indicated by the alignment between the actual and target labels and the confidence level assigned to each prediction. This visual representation provides valuable insights into the effectiveness of the Xception model in discerning urban and rural areas based on the provided dataset.

The performance of three deep learning models, namely, InceptionV3, DenseNet201, and Xception, was evaluated for classifying cases into the urban and rural classes, as depicted in Figure 9 in the form of a confusion matrix. InceptionV3 exhibited 44 misclassified cases, DenseNet201 had 22 errors, and Xception demonstrated the lowest number of errors

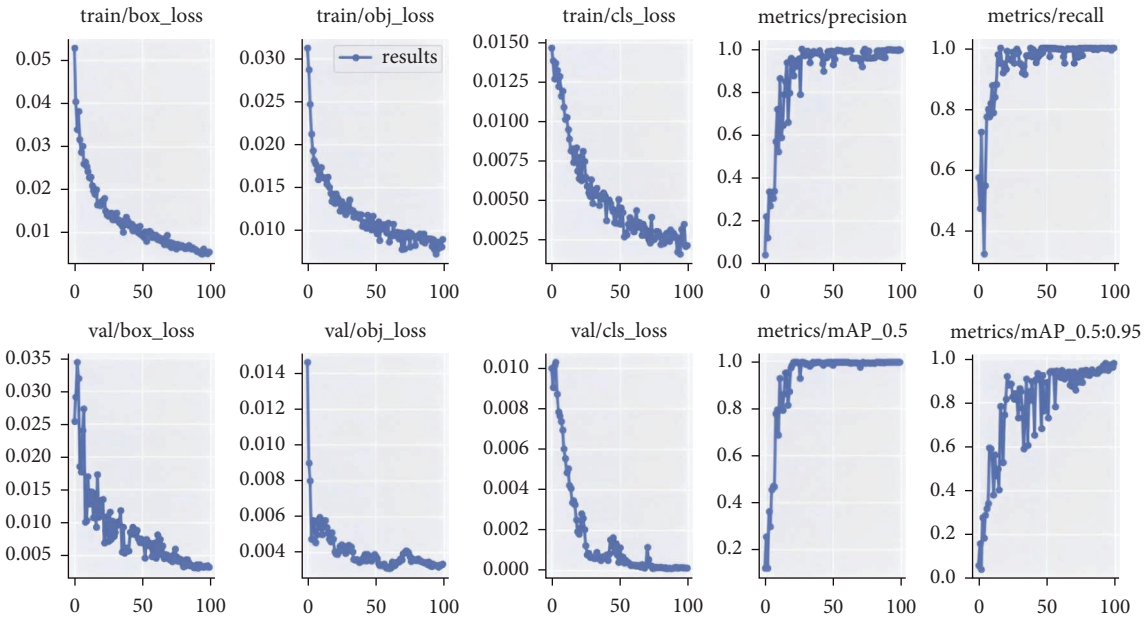


Figure 6: Graph of precision, recall, and mAP for YOLOv5 over the training epochs.


Figure 7: Images from the test set showing the performance for detecting urban areas and rural areas using YOLOv5.

Table 2: YOLOv5 performance across precision, recall, and mAP.

| Model | Precision | Recall | mAP @ 0.5 | mAP @ 0.5: 0.95 |
|---|---|---|---|---|
| YOLOv5 | 0.995 | 0.999 | 0.995 | 0.978 |

with 15 instances. These findings provide valuable insights into the accuracy and effectiveness of these models in accurately classifying cases into the urban and rural categories. Such information is crucial for researchers and practitioners in the field of deep learning when selecting appropriate models for similar classification tasks.

The performance of each architecture is individually examined to justify the performance of the proposed classification approach based on pretrained networks. Table 3 displays the accuracy of three deep learning models, namely, InceptionV3, DenseNet201, and Xception, in classifying cases into the urban and rural classes. InceptionV3 achieved an accuracy of 93.26%, DenseNet201 demonstrated a higher accuracy of 96.63%, and Xception showcased the highest accuracy of 97.70%.

Receiver operating characteristics (ROC) curves are a way to show how the true positive rate (TPR) compares to the false positive rate (FPR) based on the values of the classification thresholds. The receiver operating characteristics (ROC) curves for the two pretrained architectures


Figure 8: Images from the test set showing the performance for detecting urban areas and rural areas using Xception.

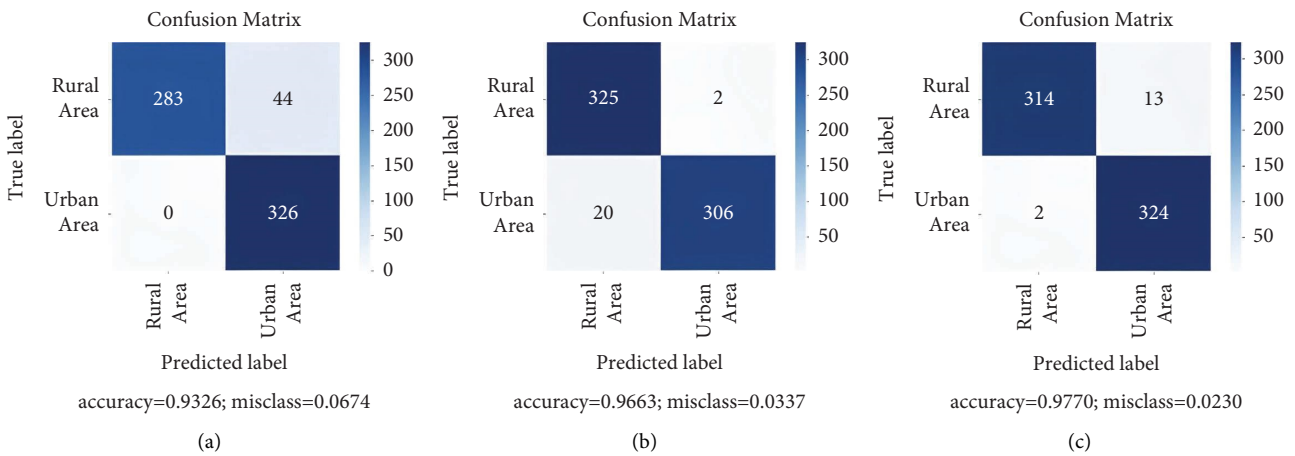


Figure 9: The confusion matrix for (a) InceptionV3, (b) DenseNet201, and (c) Xception.

Table 3: Classification results from pretrained networks.

| Models | Precision (%) | Recall (%) | F1-score (%) | Accuracy (%) |
|---|---|---|---|---|
| InceptionV3 | 88.11 | 100 | 93.68 | 93.26 |
| DenseNet201 | 99.35 | 93.87 | 96.53 | 96.63 |
| Xception | 96.14 | 99.39 | 97.74 | 97.70 |

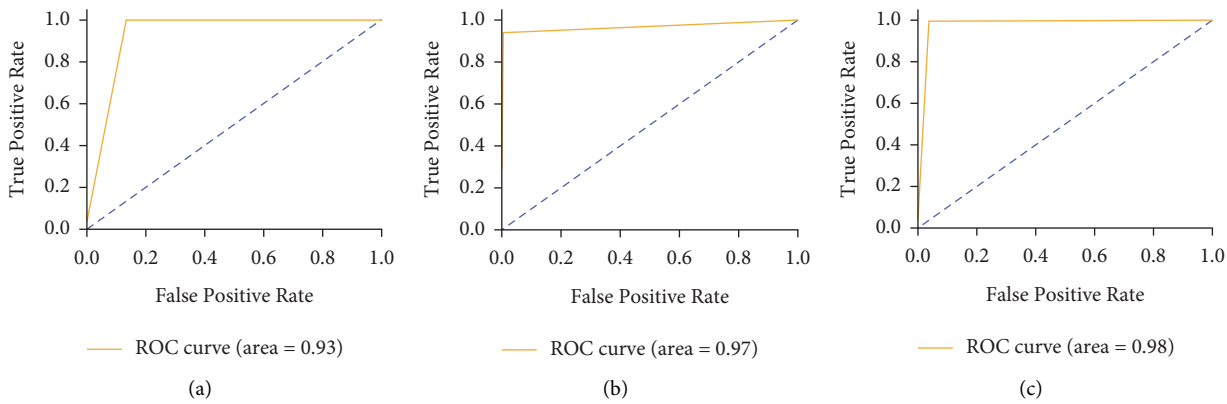


Figure 10: ROC curves of (a) InceptionV3, (b) DenseNet201, and (c) Xception.

on our area dataset are visible in Figure 10. In terms of the ROC curve performance, it can be seen clearly that Xception performs better than InceptionV3 and DenseNet201.

## 5. Conclusions

This article presents the development of a dataset for the identification of rural and urban areas in Bangladesh, along with an investigation of two distinct approaches: a detection approach utilizing YOLOv5 and a classification approach employing CNN. The principal limitation encountered in this study pertains to the restricted quantity of available images. In order to address this constraint, transfer learning techniques were applied, leveraging pretrained YOLOv5 and three DCNN architectures, namely, InceptionV3, DenseNet201, and Xception. The detection approach based on YOLOv5 exhibited favorable outcomes, achieving mean average precision (mAP) scores of 0.995 and 0.978 at intersection-over-union (IOU) thresholds of 0.5 and 0.95, respectively, when evaluated against the test datasets. In the classification approach, Xception emerged as the most proficient model, attaining an accuracy of 97.70%. To augment the comprehensiveness and reliability of the study, future efforts will entail an expansion of the image dataset, incorporating an increased number of images and classes. This expansion aims to facilitate more robust and precise conclusions. In addition, the exploration of ensemble methods integrating alternative architectural models will be pursued, with the objective of gauging their impact on overall performance. The findings presented in this research contribute to the ongoing advancement of rural and urban area identification in the context of Bangladesh, leveraging computer vision methodologies. The identified limitations and proposed avenues for further investigation establish a foundation for future research endeavors in this domain.

## Data Availability

The data used in this study are available upon request from the corresponding author.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

[1] E. J. Sadgrove, G. Falzon, D. Miron, and D. W. Lamb, "Real-time object detection in agricultural/remote environments using the multiple-expert colour feature extreme learning machine (MEC-ELM)," *Computers in Industry*, vol. 98, pp. 183–191, 2018.

[2] V. Reilly, H. Idrees, and M. Shah, "Detection and tracking of large number of targets in wide area surveillance," in *Computer Vision – ECCV 2010. ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., Vol. 6313, Springer, Berlin, Germany, 2010.

[3] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.

[4] H. Gao, C. Wang, G. Wang, Q. Li, and J. Zhu, "A new crop classification method based on the time-varying feature curves of time series dual-polarization sentinel-1 data sets," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 7, pp. 1183–1187, 2020.

[5] M. Rußwurm, C. Pelletier, M. Zollner, S. Lefèvre, and M. Körner, "BREIZHCROPS: a time series dataset for crop type mapping," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B2-2020*, pp. 1545–1551, 2020.

[6] A. Davari, V. Christlein, S. Vesal, A. Maier, and C. Riess, "GMM supervectors for limited training data in hyperspectral remote sensing image classification," in *Computer Analysis of Images and Patterns. CAIP 2017*, M. Felsberg, A. Heyden, and N. Krüger, Eds., Vol. 10425, Springer, Berlin, Germany, 2017.

[7] K. Raiyani, T. Gonçalves, L. Rato, P. Salgueiro, and J. R. Marques da Silva, "Sentinel-2 image scene classification: a comparison between Sen2Cor and a machine learning approach," *Remote Sensing*, vol. 13, no. 2, p. 300, 2021.

[8] I. Duporge, O. Isupova, S. Reece, D. W. Macdonald, and T. Wang, "Using very-high-resolution satellite imagery and deep learning to detect and count African elephants in heterogeneous landscapes," *Remote Sensing in Ecology and Conservation*, vol. 7, pp. 369–381, 2021.

[9] E. Guirado, S. Tabik, M. L. Rivas, D. Alcaraz-Segura, and F. Herrera, "Whale counting in satellite and aerial images with

deep learning," *Scientific Reports*, vol. 9, no. 1, Article ID 14259, 2019.

[10] E. H. Helmer, N. R. Goodwin, V. Gond, C. M. Souza, and G. P. Asner, "Characterizing tropical forests with multispectral imagery," in *Land Resources Monitoring, Modeling, and Mapping with Remote Sensing*, pp. 367–396, CRC Press, Boca Raton, FL, USA, 2015.

[11] J. H. Lee, J. T. S. Sumantyo, M. M. Waqar, and J. H. Kim, "Analysis of forest loss by Sentinel-1 SAR time series," in *Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 182–184, IEEE, Jeju Islan, Korea, October, 2020.

[12] V. Lebedev, V. Ivashkin, I. Rudenko et al., "Precipitation nowcasting with satellite imagery," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2680–2688, Anchorage, AK, USA, July 2019.

[13] B. L. Pavuluri, R. S. Vejendla, P. Jithendra, T. Deepika, and S. Bano, "Forecasting meteorological analysis using machine learning algorithms," in *Proceedings of the 2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 456–461, Trichy, India, September 2020.

[14] A. Femin and K. S. Biju, "Accurate detection of buildings from satellite images using CNN," in *Proceedings of the 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1–5, IEEE, Istanbul, Turkey, June 2020.

[15] B. Oshri, A. Hu, P. Adelson et al., "Infrastructure quality assessment in africa using satellite imagery and deep learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 616–625, London, UK, July 2018.

[16] N. Efremova, D. Zausaev, and G. Antipov, "Prediction of soil moisture content based on satellite data and sequence-to-sequence networks," 2019, https://arxiv.org/abs/1907.03697.

[17] M. Foucras, M. Zribi, and A. Kallel, "Soil moisture estimation at 500m using sentinel-1: application to african sites," in *Proceedings of the 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 1–5, IEEE, Sfax, Tunisia, September 2020.

[18] H. Gong, T. Mu, Q. Li et al., "Swin-transformer-enabled YOLOv5 with attention mechanism for small object detection on satellite images," *Remote Sensing*, vol. 14, no. 12, p. 2861, 2022.

[19] G. Cheng, C. Lang, M. Wu, X. Xie, X. Yao, and J. Han, "Feature enhancement network for object detection in optical remote sensing images," *Journal of Remote Sensing*, vol. 2021, Article ID 9805389, 2021.

[20] M. A. Kadhim and M. H. Abed, "Convolutional neural network for satellite image classification," *Intelligent Information and Database Systems: Recent Developments*, vol. 11, pp. 165–178, 2020.

[21] M. Pritt and G. Chern, "Satellite image classification with deep learning," in *Proceedings of the 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–7, IEEE, Washington, DC, USA, October 2017.

[22] A. Groener, G. Chern, and M. Pritt, "A comparison of deep learning object detection models for satellite imagery," in *Proceedings of the 2019 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–10, IEEE, Washington, DC, USA, October 2019.

[23] C. Kyrkou and T. Theocharides, "EmergencyNet: efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1687–1699, 2020.

[24] Z. Pan, J. Xu, Y. Guo, Y. Hu, and G. Wang, "Deep learning segmentation and classification for urban village using a worldview satellite image based on U-net," *Remote Sensing*, vol. 12, no. 10, p. 1574, 2020.

[25] C. Yoo, D. Han, J. Im, and B. Bechtel, "Comparison between convolutional neural networks and random forest for local climate zone classification in mega urban areas using Landsat images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 157, pp. 155–170, 2019.

[26] W. Yang, X. Yin, and G. Xia, "Learning high-level features for satellite image classification with limited labeled samples," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 8, pp. 4472–4482, 2015.

[27] V. K. Panchal, P. Singh, N. Kaur, and H. Kundra, "Biogeography based satellite image classification," 2009, https://arxiv.org/abs/0912.1009.

[28] D. Dai and W. Yang, "Satellite image classification via two-layer sparse coding with biased image representation," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 1, pp. 173–176, 2011.

[29] C. Li, T. Yang, S. Zhu, C. Chen, and S. Guan, "Density map guided object detection in aerial images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 190-191, Seattle, WA, USA, June, 2020.

[30] M. S. Rahman, H. Mohiuddin, A. A. Kafy, P. K. Sheel, and L. Di, "Classification of cities in Bangladesh based on remote sensing derived spatial characteristics," *Journal of Urban Management*, vol. 8, no. 2, pp. 206–224, 2019.

[31] A. Rahman, H. M. Abdullah, M. T. Tanzir et al., "Performance of different machine learning algorithms on satellite image classification in rural and urban setup," *Remote Sensing Applications: Society and Environment*, vol. 20, Article ID 100410, 2020.

[32] R. Mathieu, J. Aryal, and A. Chong, "Object-based classification of ikonos imagery for mapping large-scale vegetation communities in urban areas," *Sensors*, vol. 7, no. 11, pp. 2860–2880, 2007.

[33] H. Henderi, "Comparison of min-max normalization and Z-score normalization in the K-Nearest Neighbor (knn) algorithm to test the accuracy of types of breast cancer," *IJIIS: International Journal of Intelligent Information Systems*, vol. 4, no. 1, pp. 13–20, 2021.

[34] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan, "Analysis of transfer learning for deep neural network based plant classification models," *Computers and Electronics in Agriculture*, vol. 158, pp. 20–29, 2019.

[35] K. Zhang, C. Wang, X. Yu et al., "Research on mine vehicle tracking and detection technology based on YOLOv5," *Systems Science and Control Engineering*, vol. 10, no. 1, pp. 347–366, 2022.

[36] Y. Y. Liau and K. Ryu, "Status recognition using pre-trained YOLOv5 for sustainable human-robot collaboration (HRC)

system in mold assembly," *Sustainability*, vol. 13, no. 21, Article ID 12044, 2021.

[37] S. Han, X. Dong, X. Hao, and S. Miao, "Extracting objects' spatial–temporal information based on surveillance videos and the digital surface model," *ISPRS International Journal of Geo-Information*, vol. 11, no. 2, p. 103, 2022.

[38] Roboflow, "How to train YOLOv5 on custom objects," 2016, https://blog.roboflow.com/train-yolov5-classification-custom-data.

[39] G. Jocher, A. Stoken, J. Borovec et al., "Ultralytics/yolov5: v4.0—nn.SiLU() activations, weights & biases logging, PyTorch hub integration," 2021, https://zenodo.org/record/4418161.

[40] F. Zhou, H. Zhao, and Z. Nie, "Safety helmet detection based on YOLOv5," in *Proceedings of the 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, pp. 6–11, IEEE, Shenyang, China, January 2021.