The Egyptian Journal of Remote Sensing and Space Sciences 26 (2023) 937–953

Contents lists available at ScienceDirect

The Egyptian Journal of Remote Sensing and Space Sciences

journal homepage: www.sciencedirect.com

Research Paper

Classification of buildings from VHR satellite images using ensemble of U-Net and ResNet

S. Vasavi [*], Hema Sri Somagani, Yarlagadda Sai

[a] *Velagapudi Ramakrishna Siddhartha Engineering College, Andhra Pradesh, India*

ARTICLE INFO

Keywords:
Feature Extraction
Convolutional Neural Networks
VHR images
Semantic Segmentation
ResNet- 34
Building Classification
Geo Referencing

ABSTRACT

The urbanization rate of India is 35.9 % by 2022 reports. In this 45.23 % of urbanization is happening in Maharashtra and it is the third most urbanized state of India after Tamil Nadu and Kerala. In metropolitan areas, the classification of land cover from satellite images has been the focus of remote sensing over the years. Due to complex architecture and a lack of labeled data, classifying buildings in metropolitan areas from very high resolution (VHR) satellite imagery is challenging. Traditional approaches for building classification include hand-crafted features and transfer learning methods. These methods often struggle with the variability in building shapes, orientation, and viewpoint, leading to low accuracy in densely populated urban areas and limited performance when dealing with high- resolution satellite images. A deep-learning based approach for semantic segmentation using U-Net with a backbone of ResNet-34 is proposed for building classification. Urban area Dataset with Images of 0.5 m resolution is prepared from SASPlanet. One hot Encoding is applied for classifying buildings. U-Net is trained with encoded data. The proposed model is evaluated on the Indian dataset, specifically, the urban areas of Nashik, Maharashtra state and the accuracy obtained for the classification dataset is 60 % and the accuracy of the building detection is about 85 %. Change detection is calculated from bi-temporal images. The GIS maps are updated to detect changes in buildings, represented by different colors to distinguish newly constructed buildings, existing structures and demolished ones.

1. Introduction

Buildings are more than just physical structures. They are crucial elements that shape urban areas in terms of land use, density and efficiency. They also have significant social, economic and environmental impacts on cities. Therefore, buildings play a vital role in urban planning, as they determine the character, function and sustainability of urban areas (Akçay and Aksoy, 2011). Classification of buildings from satellite images is a valuable tool for urban planners to gain insights into the distribution, types and conditions of buildings in a city. Using image processing techniques (Ansari et al., 2017; Kay et al., 2009; Mohd. Aquib Ansari, 2017) and deep learning algorithms urban planners can accurately identify and classify buildings based on their characteristics, such as roof shape, size and color (Livne et al., 2019). This information can be used to assess building density which can inform urban Planning decisions, including land use planning, and infrastructure development (Katpatal et al., 2008).

Deep learning, which involves training neural.

Networks to recognize patterns and make predictions based on massive amounts of data, is now the most popular technique. Each pixel in an image is categorized by the CNN called U-Net (Ronneberger et al. 2015)), which was created for image segmentation. The encoder and decoder networks in the U-Net architecture have symmetric expanding and contracting paths that go around a bottleneck layer. The encoder network extracts high-level characteristics from the input image similarly to a conventional Convolutional neural network. The decoder network reconstructs the segmentation map from the retrieved characteristics (Zhang and Tang, 2018).

ResNet34 is a 34-layer Convolutional neural network (CNN) architecture built on the idea of residual learning. Using skip connections, enables quick and accurate learning by omitting intermediary layers. The deep representations of ResNet34 enable it to extract intricate visual information from images, making it ideal for applications like image classification, object identification, and image segmentation. It is a potent feature extractor because it has already been pre-trained on huge datasets like ImageNet. ResNet34 is a popular choice for image segmentation due to its balanced performance and simpler architecture compared U-Net++, U-Net + ResNet50, and VGG-19. It is easier to train

* Corresponding author.
E-mail addresses: vasavi_movva@vrsiddhartha.ac.in (S. Vasavi), 208w1a0549@vrsec.ac.in (H. Sri Somagani), 208w1a0565@vrsec.ac.in (Y. Sai).

https://doi.org/10.1016/j.ejrs.2023.11.008
Received 10 July 2023; Received in revised form 16 October 2023; Accepted 5 November 2023

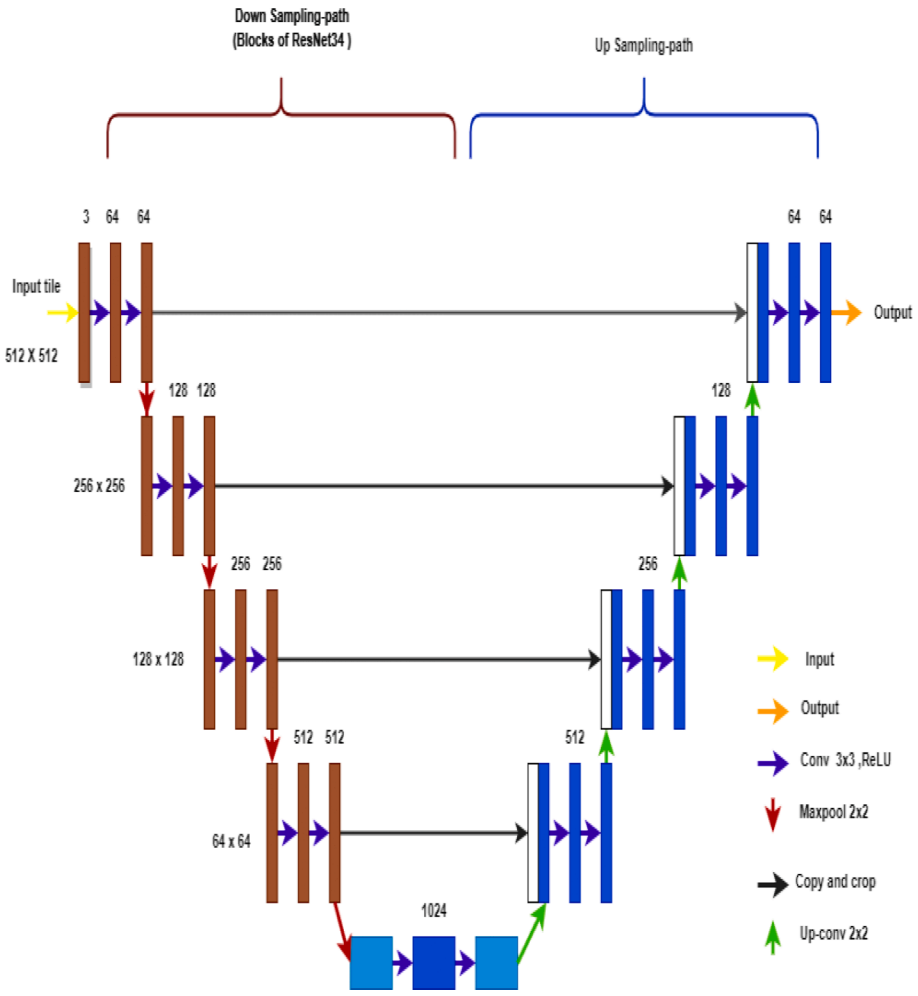


Fig. 1. Proposed U-Net Architecture.

Table 1
Proposed Ensemble Model.

Parameters	**U-Net Model** (Wurm et al., 2015)	**ResNet 34 model** (Alsabhan et al., 2022)	**Ensemble Model**
Channels in input image	1	3	3
Shape of input image	512 x 512x 1	256 x 256 x 3	512 x 512 x3
Strides	1	2	2
Input Kernel size	3x3	3x3	7x7
Initial Number of Filters	64	64	64
Parameters	1.9 M	0.46 M	28 M
Pooling type	Max Pooling	Average Pooling	Max Pooling
Size of Max Pooling at every layer	2 × 2	3 x 3	3 x 3
No.of Layers	27	34	34
Channels in the output image	1	1	1

and offers improved feature learning compared to VGG-19 despite having a smaller kernel channel (He et al., 2016).

1.1. Objectives of the proposed study

- To create a dataset that is specific to the city of Nashik, located in the state of Maharashtra, India. This dataset will be used for further analysis and development of models that can accurately classify various urban building types.
- To develop a deep learning-based model that can accurately classify different types of buildings present in urban areas by analyzing high-resolution satellite images. This model can be used for various applications, such as urban planning, disaster management and environmental monitoring.
- To update the Geographic Information System (GIS) maps with the latest information on urban area changes like new buildings, demolished buildings and unchanged buildings.

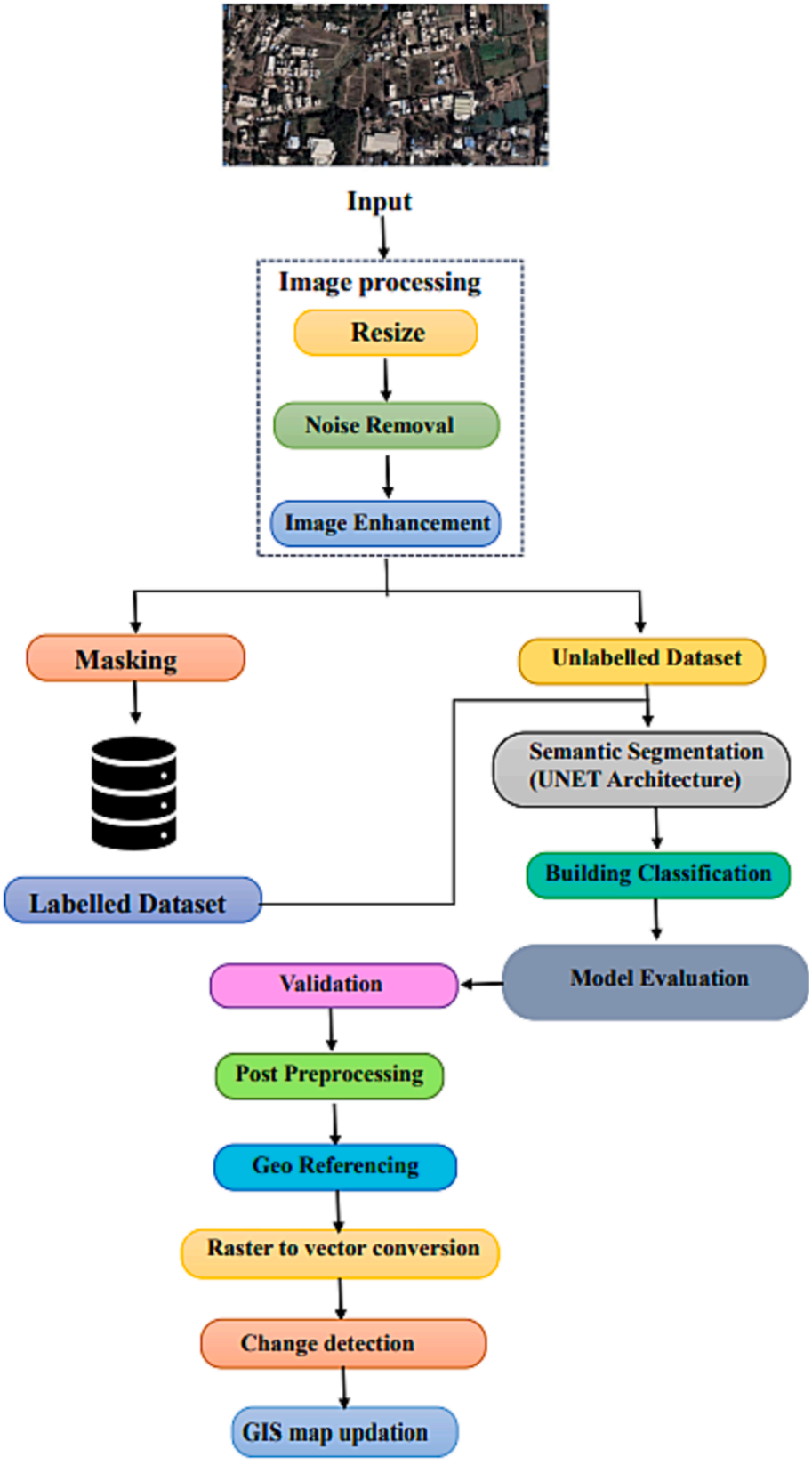


Fig. 2. Proposed System.

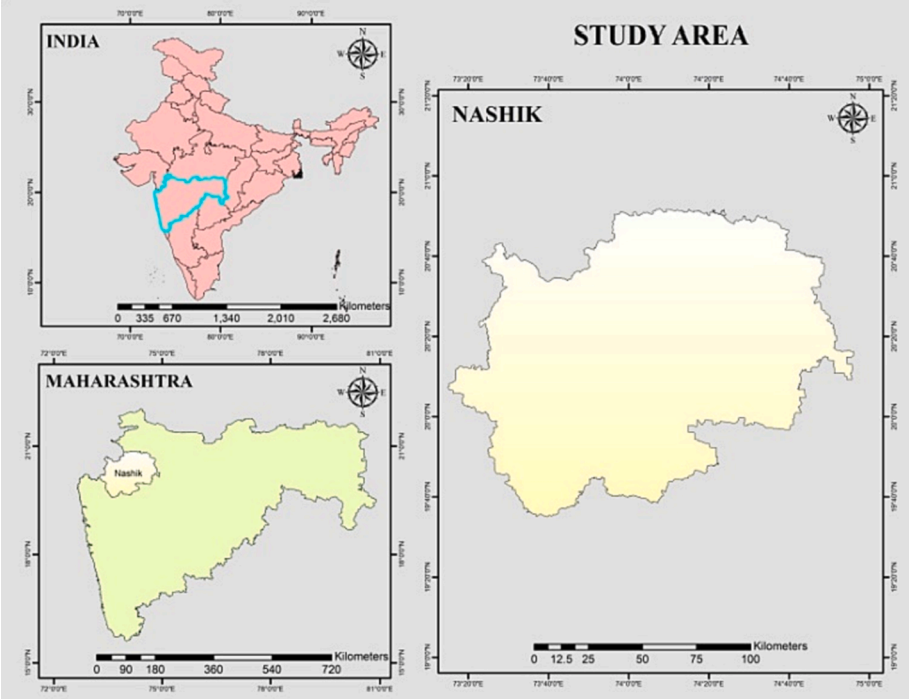


Fig. 3. Nashik Map.

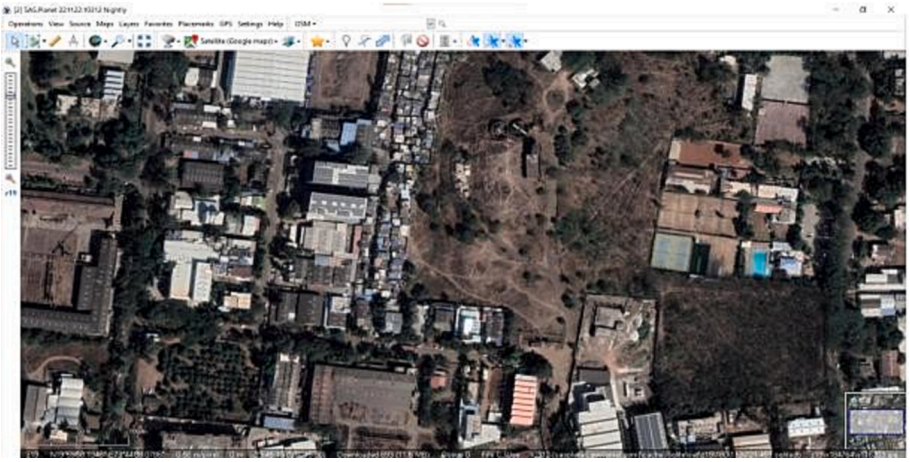


Fig. 4. Satellite image from SASPlanet.

1.2. Organization

This paper is structured as follows: Section II provides a literature review of different building classification and change detection studies. Section III goes into further detail on the recommended approach and design, followed by Section IV, which encompasses the Results and Discussion.

2. Literature review

In this study (Reda and Kedzierski, 2020), Convolutional Neural Networks (CNN) was employed to classify buildings in urban and sub-urban areas. They used a Faster R-CNN model, combining a feature extractor and a Region Proposal Network (RPN) for building structure identification. Additionally, they introduced a novel technique for building border adjustment. To evaluate model accuracy, they created a database of 500 images featuring towns of varying dimensions and architectural styles. Notably, the utilization of Adam and RMSProp

Fig. 5. Image from SASPlanet and corresponding patches.

optimizations delivered superior results in structure identification and classification. However, the model faced challenges in accurately recognizing garages.

The research described in (Lloyd et al., 2020) introduced a Geographic Information System (GIS) method for the semi-automatic preparation of data for a newly created classification model. This model relied on stacked generalizations and harnessed image extraction techniques for building footprint data. Their workflow incorporated the integration of multiple data sources to enhance the model's predictive capabilities. The ensemble model achieved an accuracy range of 85 % to 93 % in classifying structures as residential or non-residential. Nevertheless, the study did not explore the model's transferability at the local/regional level.

In (Zhao et al., 2018), Deep Neural Network (DNN) and Inception-Extended Profile (IEP) models were employed for feature extraction to identify architectural style similarities and differences. Google Net's Inception modules were utilized alongside data augmentation techniques to reduce computational costs and prevent over fitting. Their approach successfully categorized architectural styles into 25 distinct categories, achieving an impressive 98.57 % accuracy in architectural categorization. However, it's important to note that this model's classification capability was limited to Greek and Georgian architectural styles.

The authors of (Kang et al., 2018) proposed a comprehensive framework for classifying land use at the building level. This approach involved the extraction of building footprints from GIS maps, outlier elimination, and the use of a Convolutional Neural Network (CNN) for building categorization, initially employing the publicly available VGG16 model. The framework demonstrated high accuracy in determining the land use of specific buildings. Nonetheless, challenges emerged when dealing with densely clustered buildings, requiring the development of alternative strategies for data retrieval from GIS maps.

A Method in (Huang et al., 2017) introduced a framework for categorizing various building types using data from high-resolution remote sensing images and LiDAR. Building height data from LiDAR was paired with image data, considering spatial relationships and autocorrelation. This method accounted for spatial connections and autocorrelation, resulting in a more dependable classifier performance. However, the study had limitations in terms of in-depth testing and validation, and it did not fully address the landscape's impact on the results.

In this study (Wurm et al., 2015), linear discriminant analysis (LDA) was applied to digital surface models created from aircraft photographs and building footprints from real estate cadastral data to classify various types of buildings. They evaluated 26 shape-based metrics for discriminatory ability, with the 3-D shape index and 2-D assessments of compactness making significant contributions to discrimination. The

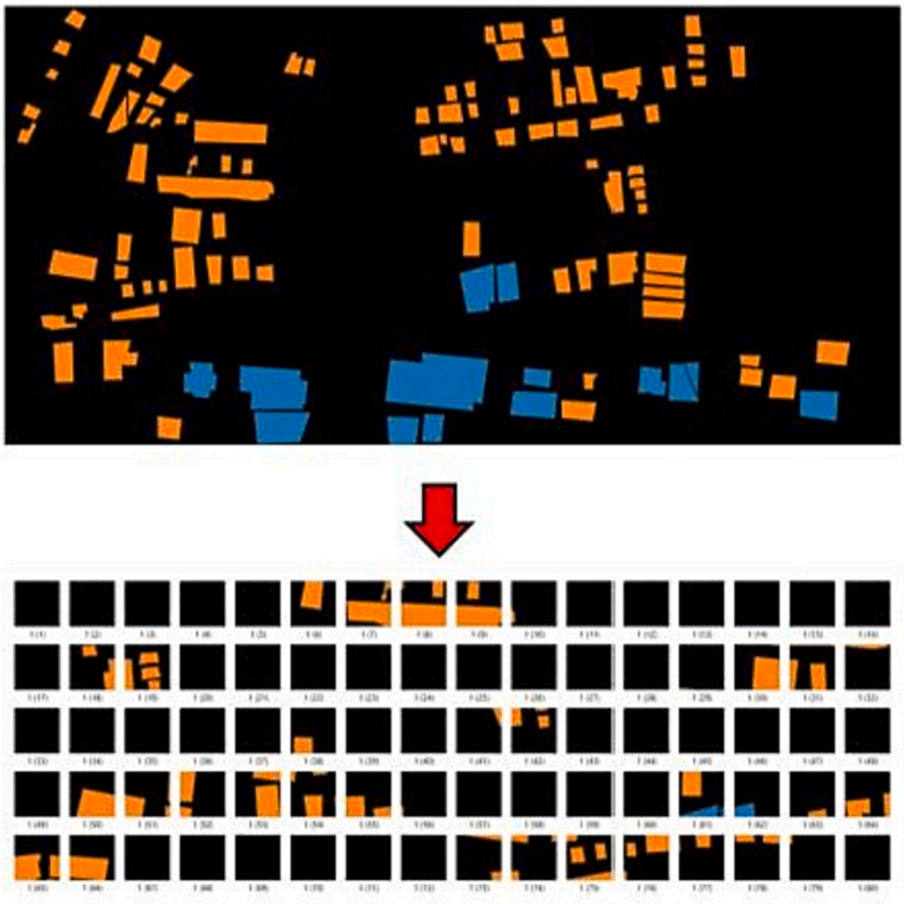


Fig. 6. Masking of SASPlanet image.

Advantages of this approach include robust experimentation (1000 runs) to reduce bias in training data and achieve excellent sensitivity and precision. However, LDA's assumption of a normal distribution for all features can limit its applicability, and linear discrimination occasionally confused similar building types, such as perimeter block development and block development.

According to the study by (Goldblatt et al., 2016), images were classified in three steps: dataset construction, scene selection and pre-processing, and pixel-based classification for built-up area recognition. An advantage noted was that the classifier's performance remained consistent with diverse land cover features in training and test sets, achieving an accuracy of approximately 87 %. However, the study lacked details about socioeconomic variables, physical characteristics, and location data, which could enhance classifier accuracy.

The study in (Kaichang et al., 2000) discussed two strategies for inductive learning from spatial data. One approach proposed two learning granularities for pixels and spatial objects, establishing rules for picture classification based on spectrum, position, and elevation data. An advantage was an overall accuracy increase of over 11 %, particularly in identifying land use, with accuracies of around 94.4 % achieved in some categories like residential areas, paddy fields, irrigated fields, vegetable fields, and water. The use of GIS data for image categorization and data mining-based techniques contributed to this accuracy. However, challenges persisted in the intelligent fusion of remote sensing and GIS data, leading to occasional misidentification of forest shadows as streams.

2.1. *Research gaps*

- Lack of standardized dataset
- Variations in building construction practices
- Limited research on Indian building classification
- Need for improved accuracy
- Limited availability of Indian Building Datasets

2.2. *Requirements*

The proposed model is developed in Google Colab environment with Tensor Flow architecture using libraries that include Operating System, pytorch, json, numpy, pandas, PIL, TensorFlow, gdal, tifffile, patchify, rasterio with python 3.9 as the backend which is powered by a workstation with Intel Core i7-9800X and a single NVIDIA GeForce MX450.

3. Materials and methods

The architecture of the system, the procedure to be utilized to carry it out, and the dataset to be used are all suggested in this section.

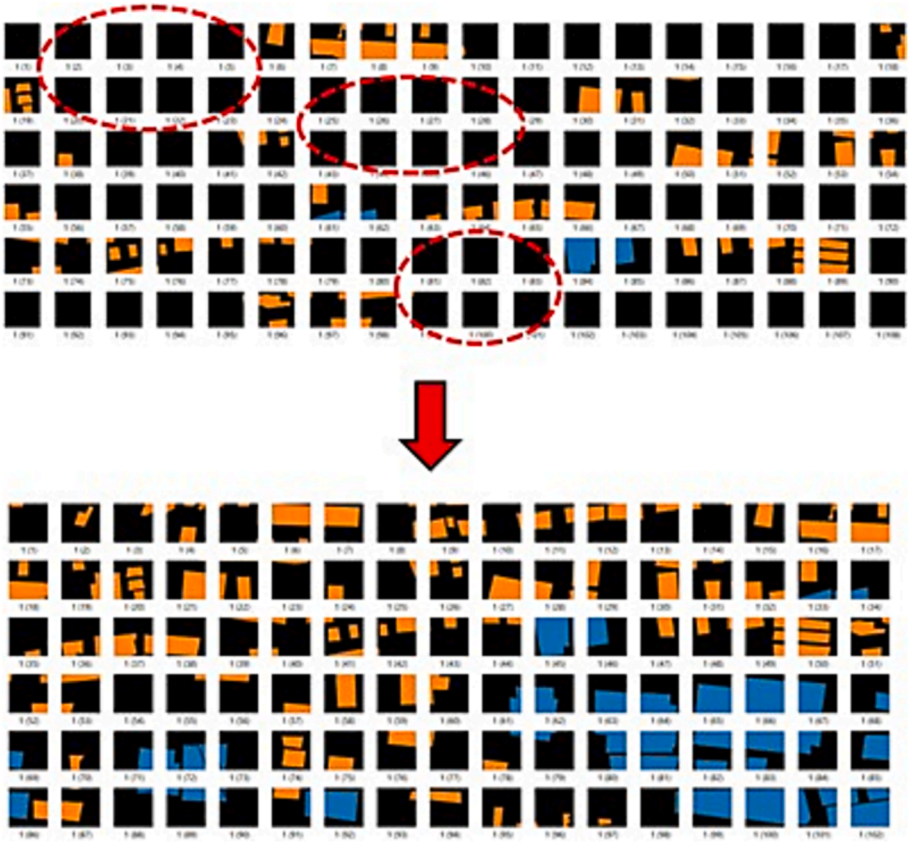


Fig. 7. Removing useless images.

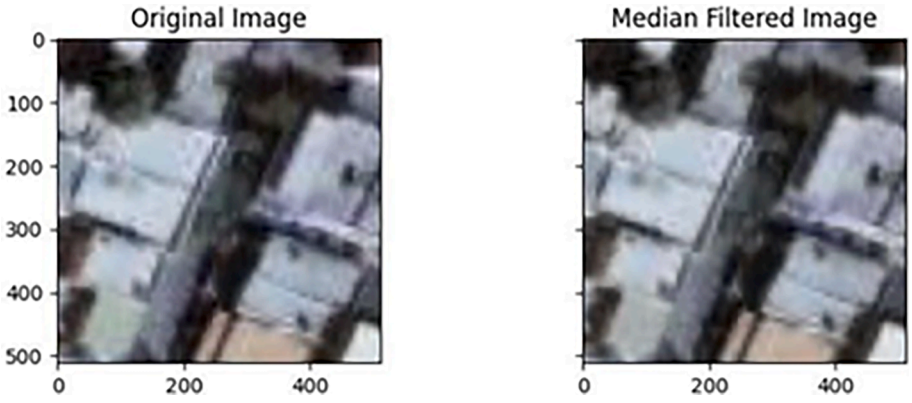


Fig. 8. Removing Noise using Median Filter.

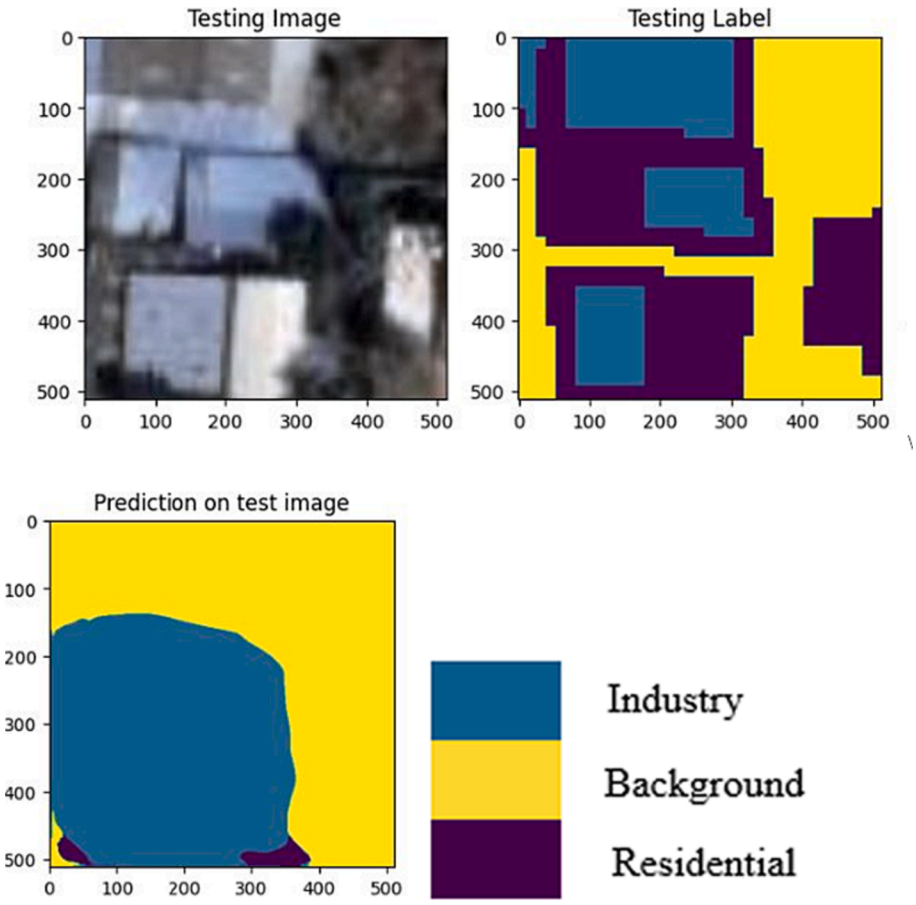


Fig. 9. Model displaying the test image, the test label, and the prediction.

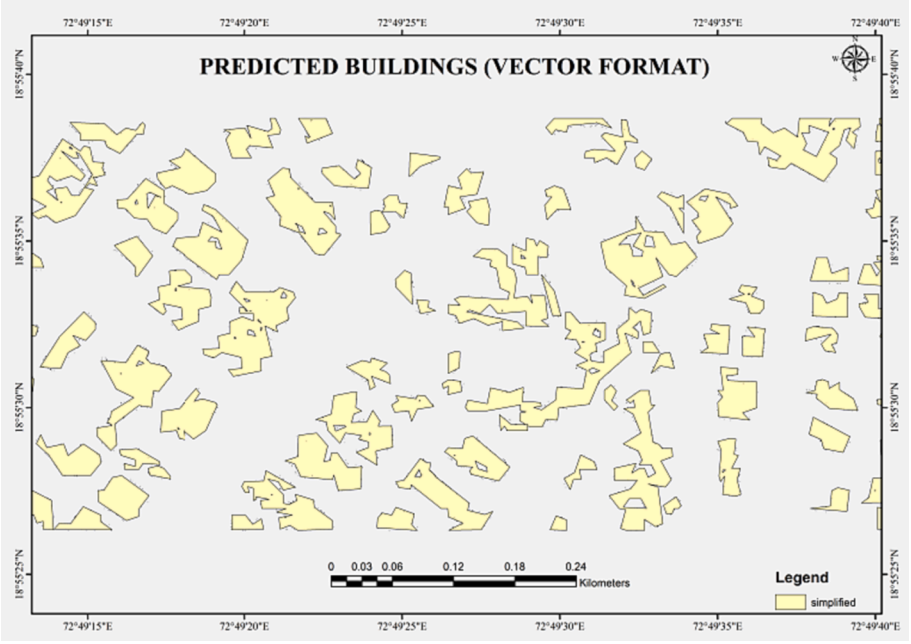


Fig. 10. Raster to vector conversion.

Table 2
Building distribution for Training, Testing and Validation phases.

Type of Buildings	Training	Testing	Validation
Residential	451	110	340
Industries	169	43	121
Holy Places	3	4	3

3.1. Architecture

Fig. 1 defines a U-Net model for image segmentation.

Input Layer: The input layer defines the shape of the input image. The input image for the U-Net model, which is intended for image segmentation tasks, has the following dimensions: 512x512x1.

ResNet-34 Backbone: This structure consists of several convolutional blocks with batch normalization and activation layers. The backbone first extracts the features from input image. The first convolutional layer in the backbone has 64 filters, a 7x7 kernel size, and a stride of 2. The next steps are batch normalization and ReLU activation. Then, using max pooling with a pool size of 3x3 and a stride of 2, the spatial dimensions are down-sampled.

U-Net Encoder Path: The encoder path of the U-Net starts with the input image. The encoder blocks perform a series of operations, including Convolution, batch normalization, activation, and max pooling. Each encoder block has 64, 128, 256, and 512 filters, respectively, and the strides in the Convolutions are set to 1, except for the first one in the path, which has a stride of 2. Max pooling with a pool size of 2x2 and a stride of 2 is also applied after each encoder block, resulting in a down-sampling of the spatial dimensions. The outputs from the encoder blocks are stored in variables s1, s2, s3, and s4, which represent the skip connections.

U-Net Bridge: The bridge links the U-Net's encoder path and decoder path. The output from the final encoder block (s4) is given a Convolutional operation. 1024 filters are in the bridge block.

U-Net Decoder Path: The output of the bridge block initiates the U-

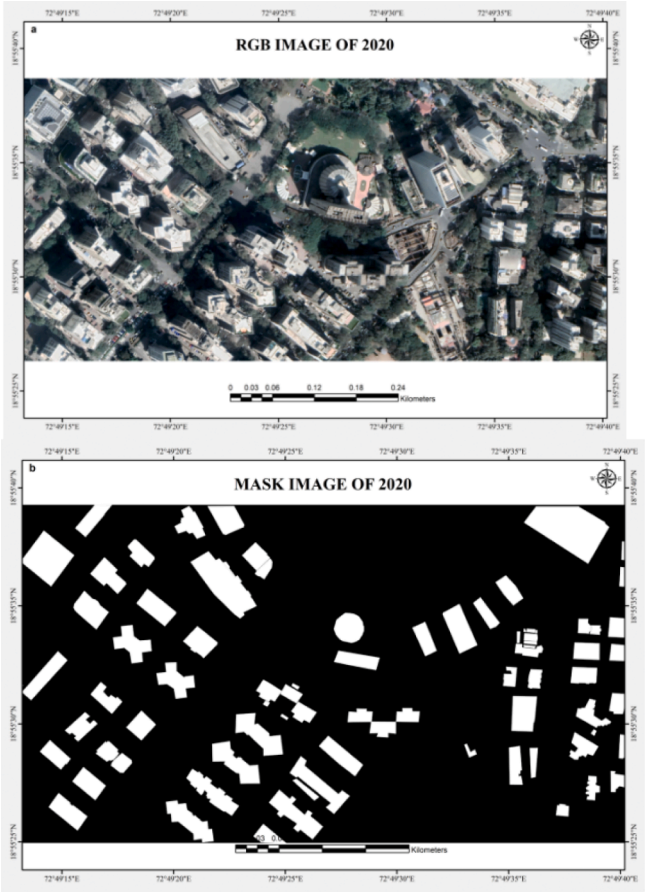


Fig. 12. A) satellite image from2020b) Binary mask of buildings.

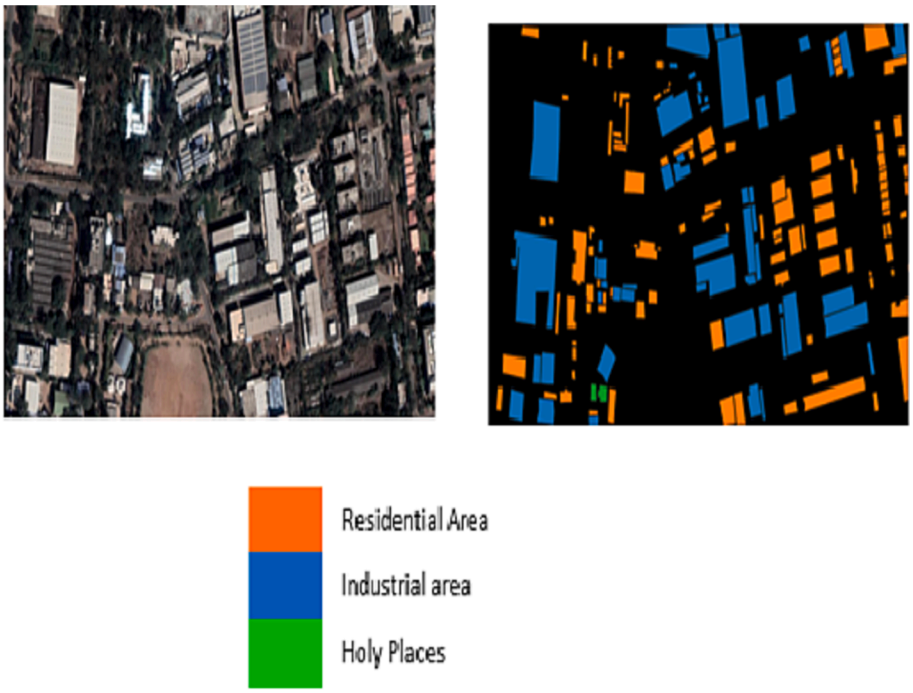
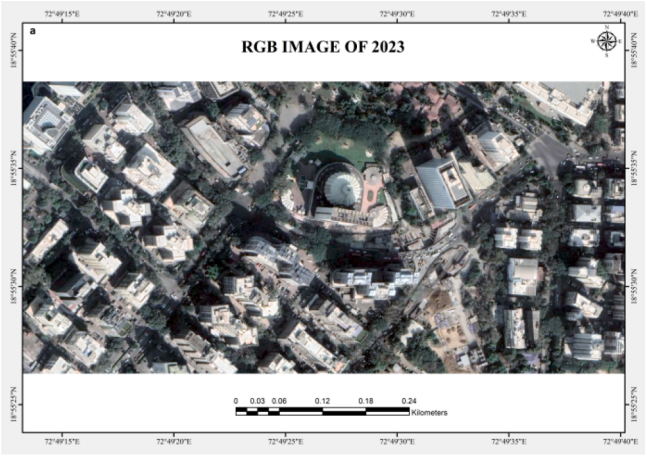


Fig. 11. Dataset distribution.



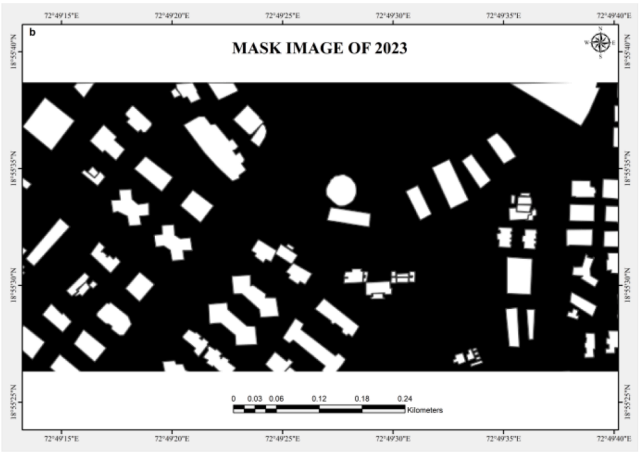


Fig. 13. A) satellite image from2023b) Binary mask of buildings.

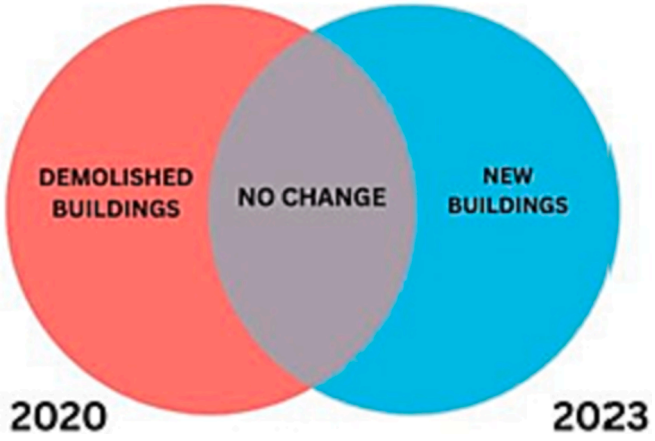


Fig. 16. Set operations.

Net's decoder path. The decoder blocks, represented by the missing decoder block function, perform a series of operations, including up-sampling, concatenation of skip connections, Convolution, batch normalization, and activation. Each decoder block has 512, 256, 128, and 64 filters, respectively. The spatial dimensions are up-sampled in each decoder block using up-sampling operations, and the corresponding skip connections are concatenated with the up-sampled feature maps. As a result, the network may utilize both low-level and high-level characteristics when up-sampling.

Final Output: Applying a Convolutional layer with Softmax activation to the output of the last decoder block yields the U-Net's final output. In this convolutional layer, there are exactly as many filters as classes. The predicted class probabilities are generated by this layer.

$$Softmax(x_i) = e^{x_i} / \sum_{j=1}^{n} e^{x_j}) \quad (1)$$

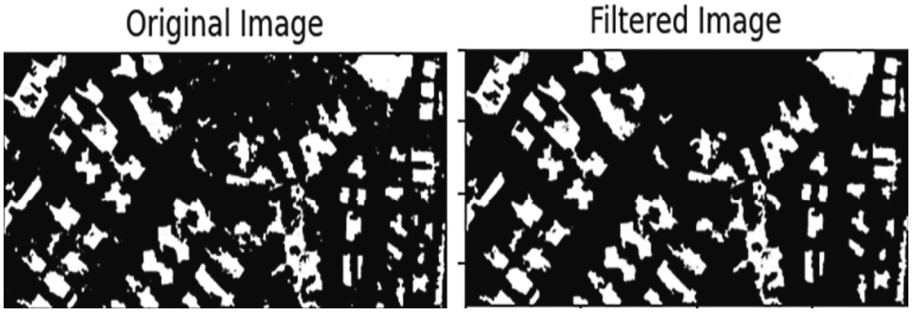


Fig. 14. Input image and area threshold image.

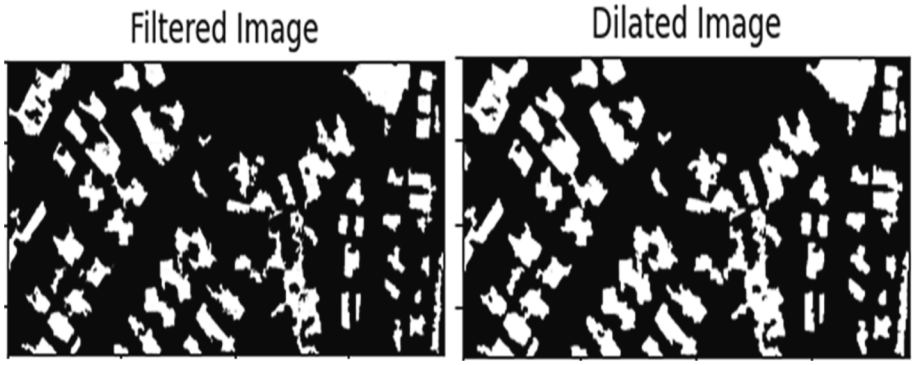


Fig. 15. Dilation.

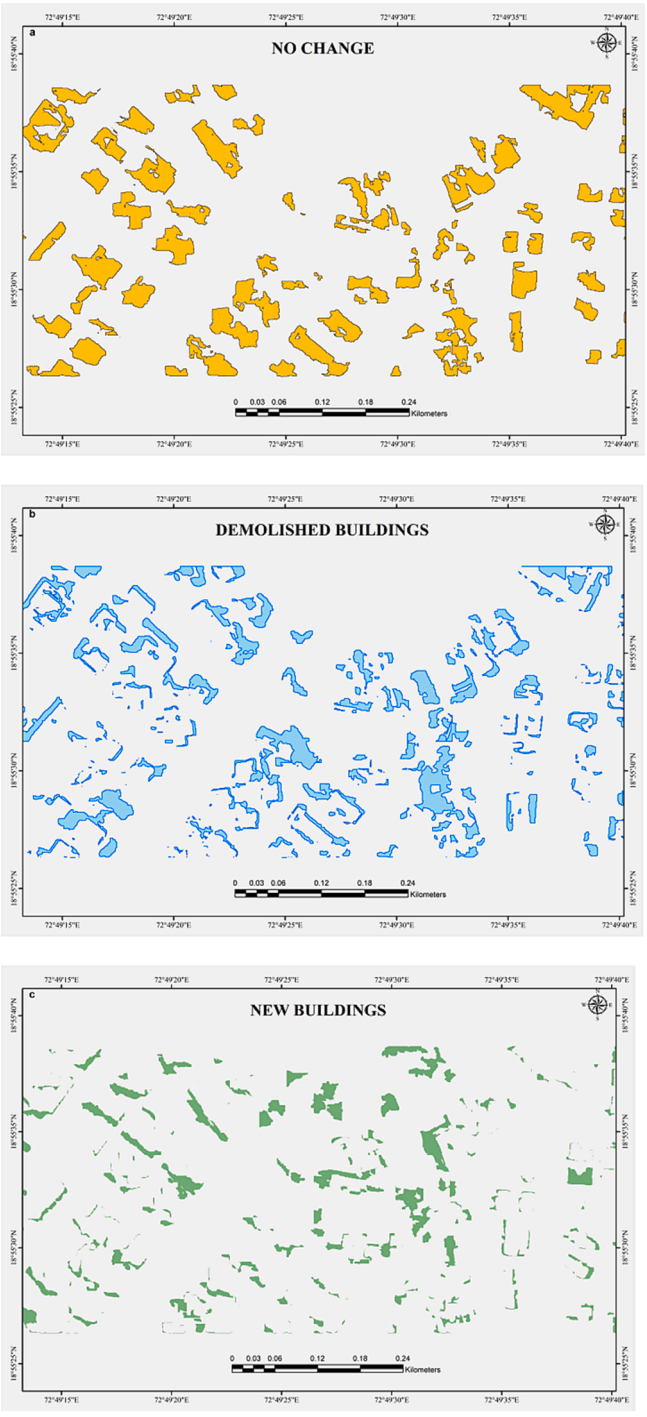


Fig. 17. A) no change b) demolished buildings c) new buildings.

The input vector x's i^{th} element is represented by xi, while the vector's total number of elements is represented by n. With each element of the output vector reflecting the likelihood of a certain class, the SoftMax function generates a probability distribution over the input vector.

Table 1 presents the details of the ensemble model proposed in this study.

3.2. Methodology

In the suggested process for identifying and categorizing structures in metropolitan areas into appropriate classifications is shown in Fig. 2. It comprises a model evaluation process as well as training and testing procedures.

Initially, a dataset consisting of 1700 images was prepared from SASPlanet.

- Pre-processing techniques like resizing, image enhancement and noise removal give pre- processed data
- Masking images from QGIS yield 945 useful images from 1700 images where the buildings are labelled using red, industries are labelled blue and holy places are labelled green.
- A series of steps, including semantic segmentation, feature extraction, building detection, and building classification into residential, industrial, and holy places, were carried out.
- Out of the 945 images, 800 were considered for training, and 145 were used for testing.

The focus of this work was on the training and testing processes required for image analysis. This involved image pre-processing, including resizing the images to 512 X 512 and applying a median filter to remove noise. QGIS software was used to mask the images.

Unlabeled datasets were used to gain insights, and building detection was performed using semantic segmentation, specifically U-NET segmentation. For building classification and model evaluation, a labelled dataset was utilized. The results were then validated. The model was trained for 25 epochs using a high-resolution satellite image dataset. The batch size was set to 16, and a single high-end GPU (NVIDIA GeForce MX450) was used, requiring 12 GB of GPU memory and system RAM. The entire training process took around 4 h. The model had a size of approximately 213 MB and contained 28 million parameters. The efficiency of the model was about 89 %.

We utilized the Adam optimizer, an adaptive learning rate optimization algorithm that amalgamates the advantages of both AdaGrad and RMSprop. For hyper parameter tuning, we opted for a Random search approach. This method, renowned for its computational efficiency, allowed us to uncover optimal hyper parameter configurations with only a limited number of iterations, streamlining the fine-tuning process. In our approach to change detection, we considered the application of pixel-wise binary cross-entropy loss, also referred to as a semantic segmentation loss.

The predicted image output was geo-referenced and converted to a vector format (GeoJSON). Set operations like intersection and set differences were used for change detection of the buildings. This change detection was used to update GIS maps.

3.3. Algorithms

Algorithm 1. Eliminating useless patches.

Step 1: Read the image and mask.

Step 2: Calculate the unique values and their counts in the mask by dividing the count of non-zero labels by the total count of labels. The expression

$$\frac{counts[0]}{counts.sum()} (2)$$

Calculates the percentage of the first unique value (which represents the background or label 0) in the mask. It divides the count of the first unique value by the sum of all counts, giving the proportion of the background label in the mask.

By subtracting this value from 1, you can obtain the percentage of the useful area in the mask as given in equation

$$1 - \frac{counts[0]}{counts.sum()} (3)$$

Step 3: Calculate the percentage of useful area in the Calculates the percentage of useful area by subtracting the background label proportion from 1.

Step 4: If the percentage of useful area is greater than 5 percentage, Save the image and mask.

Step 5: If area percentage is less than 5 percentage, then the image is considered as useless image and not considered for training the model.

Algorithm 2. Median Filtering.

Step 1: Define the size of the neighbourhood as (2 k + 1) x (2 k + 1) pixels.

Step 2: For each pixel in the image, extract the k x k neighbourhood centered on the pixel.

Step 3: In increasing order, sort the neighborhood's pixel values.

Step 4: Change the pixel value with the median value of the sorted neighbourhood.

If the neighbourhood size (2 k + 1) is odd:

$$\text{Median} = sortedValues[(2k+1)/2] \ (4)$$

If the neighbourhood size (2 k + 1) is even:

$$\text{Median} = \frac{1}{2}((sortedValues[k]) + (sortedValues[k+1])) (5)$$

Step 5: Repeat steps 2–4 for every pixel in the image.

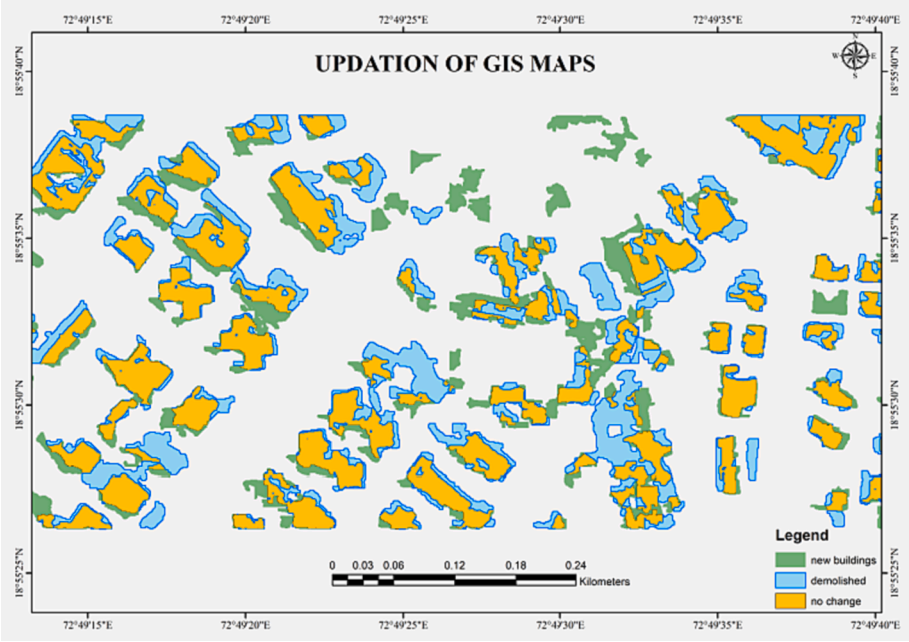


Fig. 18. Updation of GIS map in QGIS Software.

Algorithm 3. Geo-referencing Algorithm.

Step 1: Obtain a reference image that contains the desired geographic information, such as a satellite image.

Step 2: Determine the spatial reference system used by the reference image.

Step 3: Georeferencing Transformation

3.1 Identify ground control points (GCPs) in both the reference image and the predicted building segmentation images. GCPs are points with known geographic coordinates that can be used to align the two images.

3.2 Choose an Affine transformation:

$$x\_out = a\_0*x\_in + a\_1*y\_in + a\_2 \quad (6)$$

$$y\_out = a\_3*x\_in + a\_4*y\_in + a\_5 \quad (7)$$

where:

(x_in, y_in) are the pixel coordinates in the input image.

(x_out, y_out) are the pixel coordinates in the output image.

a_0, a_1, a_2, a_3, a_4, and a_5 are the parameters of the affine transformation model.

3.3. Fit the georeferencing transformation model to the GCPs. This involves solving a system of equations to determine the parameters of the transformation model.

Step 4: Coordinate Assignment.

For each pixel in the predicted building segmentation images:

4.1 Apply the georeferencing transformation model to the pixel's coordinates.

4.2 Convert the transformed coordinates to latitude and longitude.

$$latitude = \arctan\left(\frac{y_{out} - reference\_projection\_origin\_y}{x\_out - reference\_projection\_origin\_x}\right) \quad (8)$$

$$longitude = \frac{latitude*(reference\_projection\_origin\_x)}{reference\_projection\_origin\_y - y_{out}} \quad (9)$$

where: x_out and y_out are the transformed pixel coordinates.

reference_projection_origin_x and reference_projection_origin_y are the coordinates of the projection origin in the reference image are the latitude and longitude of the transformed pixel.

Step 5: Saving the Geo-referenced Image.

Algorithm 4. Raster to Vector format.

Step 1: Read the image in 'TIF' format.

Step 2: Get the coordinate reference system information

Step 3: Polygonize the raster image from CRS information

Step 4: Transform all the geometries of a polygonized raster in an active geometry column to a different CRS.

Step 5: Save the vector image of the raster image and display.

3.4. Study area and data preparation

Nashik, which is illustrated in Fig. 3 and is situated at 19.9975° N latitude and 73.7898° E longitude in the Indian state of Maharashtra, is taken into consideration for the building classification.

The population of Nashik is estimated to be 2,047,000 in 2023, with an urban area of 259.10 square kilometers. The reasons for urbanization are due to industrialization, economic opportunities, and better infrastructure (https://www.census, 2023).

Dataset is prepared using SASPlanet with 0.5 m resolution. This dataset has been collected and compiled, with the goal of being able to classify buildings into three main categories: residential, industrial, and holy places. The collected dataset consists of images of size 12025 x 5878 with a 0.5 m/pixel spatial resolution in TIF format.

Classes: 4.

Number of Images: 945.

Train set size: 750 *(80* %).

Test set size: 95 *(20* %).

Image resolution: 512 x 512 pixels.

Validation data: For validation Mumbai, India dataset is taken which is of size: 6097 x 9058 pixels. This image is patched into size of 512 x 512. After patching we will get 285 images of size 512 x 512.

Evaluation data: Standardized public datasets such as LEVIR-CD (Mohammad et al., 2022), SpaceNet (Van Etten et al., 2018), and WHU Buildings datasets (Ji et al., 2018) are used for evaluation purpose.

Fig. 19. Loss vs. Epochs.

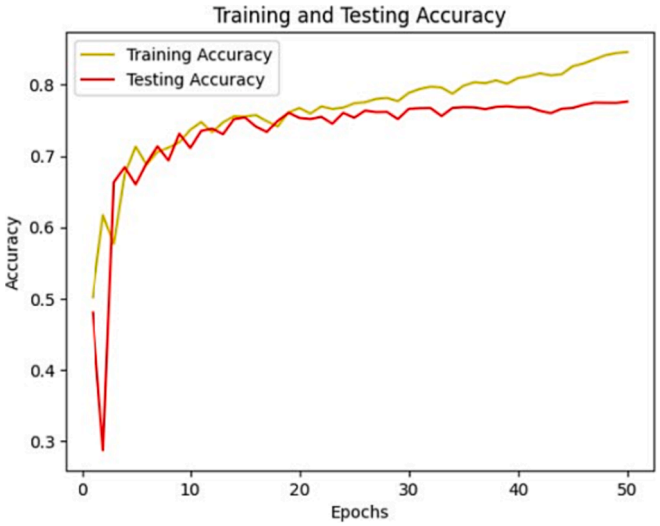


Fig. 20. Accuracy Vs. Epochs.

Fig. 21. IoU for Training and Testing.

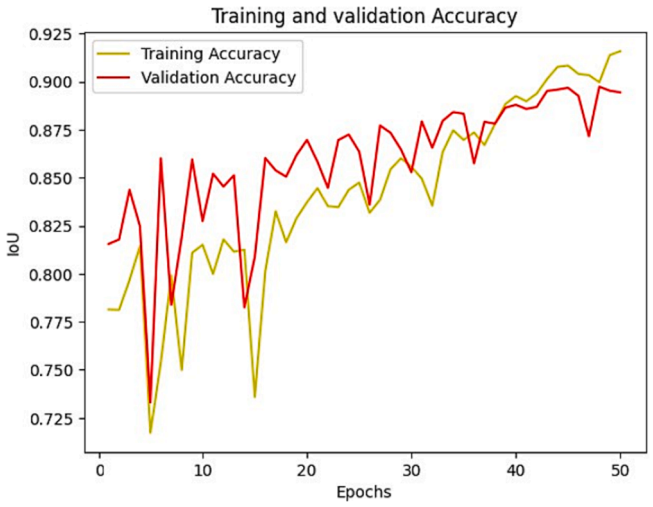


Fig. 22. Graph Showing Training and Validation Accuracy Vs. no. of Epochs.

3.5. Evaluation metrics

Several metrics, including as precision (P), recall (R), IoU and F1 score are used to assess the effectiveness of the proposed technique as given in Equations 10 to 15.The proportion of pixels that were accurately classified as buildings is known as True Positive (TP). The amount of pixels that were incorrectly classified as backgrounds is known as false positives (FP). The proportion of pixels that were correctly classified as backgrounds is known as True Negative (TN). The pixels known as False Negative (FN) pixels were mistakenly classified as structures.

$$IOU = \frac{TP}{(TP + FP + FN)} \tag{10}$$

The IoU runs from 0 to 1, and a greater value denotes stronger ground truth and prediction mask overlap. An IoU of 0 indicates no overlap at all, whereas an IoU of 1 indicates a perfect match.

$$MeanIOU = \sum_{i=1}^{N} IOU_i / N \tag{11}$$

Where N is the total number of classes, and IOU1, IOU2, …, IOUn are the IOU values for each class.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{12}$$

Precision is the fraction of correctly predicted positive samples among all expected positive samples.

$$Precision = \frac{TP}{(TP + FP)} \tag{13}$$

Recall is the proportion of all real positive samples that were correctly predicted among all positive samples.

$$Recall = \frac{TP}{(TP + FN)} \tag{14}$$

The harmonic mean of recall and precision, which is the F1 score, offers a fair comparison of the two.

Table 3
Comparison of Performance in Different Phases.

Phase	Mean IoU	Precision	Recall	F1-Score	Accuracy
Training	0.765	0.84	0.84	0.83	0.84
Testing	0.69	0.776	0.77	0.77	0.78
Validation	0.78	0.64	0.80	0.71	0.89

Table 4
Comparison with Other Models.

	Dataset	Methodo logy	Accuracy Test	Accura cy
(Aghayari, 2023)	Inria Dataset, America	U-Net, ResNet	Dice metric Test	97.95 %
(Alsabhan et al., 2022)	Xinxing County,, China	UNet,Re sNet, .VGGNet	Accuracy score	84.9
(Alsabhan and TurkyAlotaiby., 2022)	Boston	U-Net,ResNet50	IoU accuracy	82.2 %
(Erdem and UğurAvdan., 2020)	Chicago dataset	U-Net ,ResNet V2	F1 score	0.86
(Mohammad et al., 2022)	Chandiga rh	U-Net ,ResNet5 0	IoU	88 %
Propos ed Model	Nashik, Maharastr a	U-Net ,ResNet3 4	Accuracy	89 %
Proposed model	LEVIR-CD dataset (Mohammad et al., 2022)	U-Net ,ResNet34	Accuracy	88.5 %
Proposed model	SpaceNet (Van Etten et al., 2018)	U-Net ,ResNet34	Accuracy	84.7 %
Proposed model	WHU Buildings datasets (Ji et al., 2018)	U-Net ,ResNet34	Accuracy	89.6 %

$$F1-score = \frac{2*precision*recall}{(precision + recall)} \tag{15}$$

4. Results and discussion

This section has a discussion of the outcomes produced by the suggested system.

4.1. Results

Initially, the images are downloaded from SASPlanet and masking is done. The Sample image is shown in Fig. 4.

Patchify and the masking process are shown in Fig. 5 and in Fig. 6, where each satellite image of size 12025 x 5878 is divided into patches of size 512 X 512. The images are divided into patches of size 512 x 512 pixels.

Fig. 7 presents the process of eliminating useless patches. Images with an area less than of 5 % are removed. Only the images that have an area of more than 5 % are considered for training the model.

Pre-processing techniques such as splitting, noise removal and image enhancement are applied to these images. Fig. 8 presents the noise removal process.

After applying the median filter, PSNR values are calculated. If the PSNR value is less than 30 then there is no use in applying the median filter. For our own dataset, the PSNR values range from 50 to 60. Building classification is performed using the U-Net model trained with the ResNet34backbone.Fig. 9 presents the sample output of training and testing images. The model is trained with 280 images of size 512 x 512 along with their corresponding masks. The formula for calculating the Peak Signal-to-Noise Ratio (PSNR) between two images is given in Equation (16).

$$\text{PSNR} = 20*\log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right) \tag{16}$$

MSE stands for the Mean Squared Error between the two pictures, where MAX is the highest feasible pixel value for the image.

Buildings, the white area, and other objects can be predicted by the model after training. As a result of the building segmentation, the geographic information, or coordinate information, is lost. Geo-referencing is carried out to recover the geographic data. This is done to enable the overlay of the image on a map or the presentation of the image alongside other geographical data in a geographic information system (GIS).

GIS organizes raster datasets of satellite images in the GeoTIFF format and vector datasets in the GeoJSON format since the result of this is to update the GIS maps with change detection of buildings. A GeoTIFF file, or raster picture, is the result of the georeferencing process. The vector-based GIS maps use this format. Therefore, to update the GIS maps, the GeoTIFF file (i.e., raster image) is transformed to the GeoJSON as shown in Fig. 10.

Table 2 shows the distribution of buildings across different types for the training, testing, and validation datasets. The dataset is categorized into three types: Residential, Industries, and Holy Places.

Table 2 shows the count of buildings used for training, testing, and validation.

This distribution of buildings as shown in Fig. 11, across different datasets provides insights into the composition of the data used for training, testing, and validation.

In India, it can be difficult to classify mixed-use buildings as they may appear residential from a satellite view. Therefore, we have limited our classification to residential buildings, industrial buildings and holy places.

Fig. 12 represents the Satellite image of Mumbai City in the year 2020 and the corresponding Binary mask. Masking is done with the help of QGIS software. Fig. 13 represents the Satellite image of Mumbai City in the year 2020 and the corresponding Binary mask. After training the model, the user will give the input image.

After applying area thresholding to the predicted output in Fig. 14, unwanted white pixels were eliminated. In post-processing, objects or regions are filtered on the area using a fixed criterion of 10,000 pixels. Objects with an area greater than 10,000 pixels are retained, while those

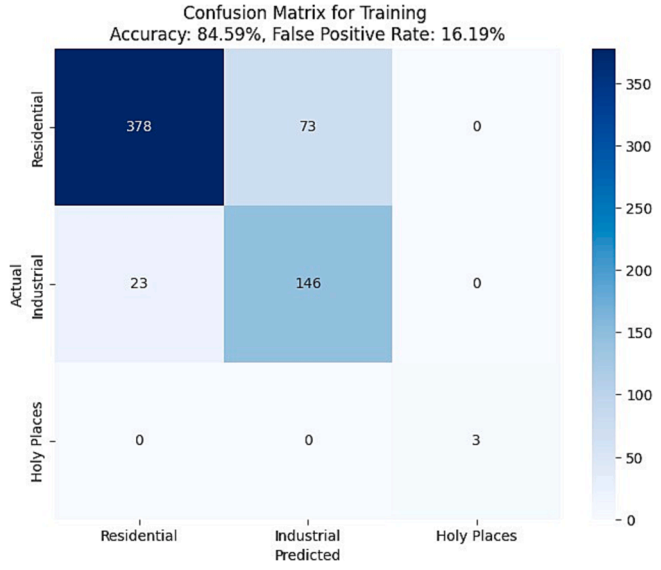


Fig. 23. Confusion Matrix for Training.

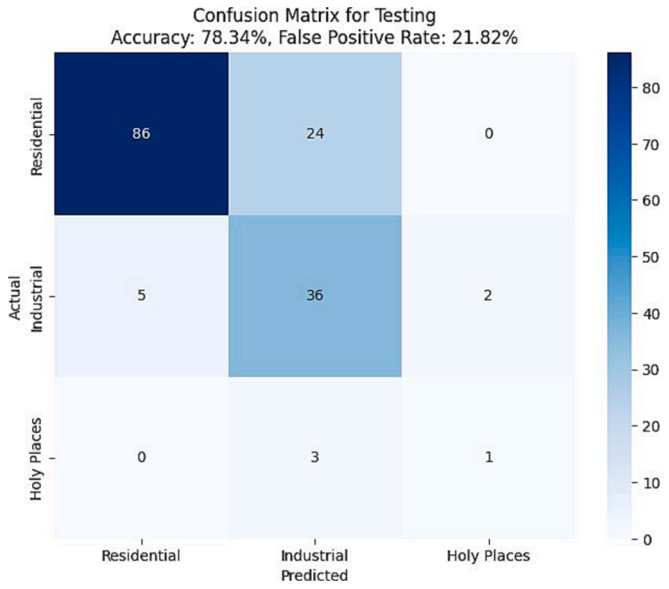


Fig. 24. Confusion Matrix for Testing.

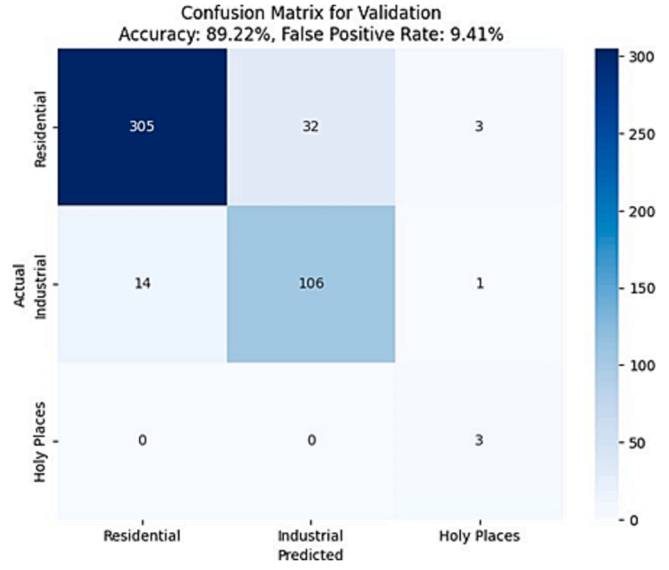


Fig. 25. Confusion Matrix for Validation.

below this threshold are typically ignored.

Rectangular white shapes of some white trucks and cars are detected as buildings from satellite view, so area thresholding is done to address this issue.

Fig. 15 shows that after applying dilation to the filtered image, some pixels are added to the edges of the buildings. The resulting image is saved in.jpg format and needs to be georeferenced according to Algorithm 3, using the input image, which is a GeoTIFF file. The output will be a GeoTIFF file with a.tif extension. This GeoTIFF file, in raster format, was converted to a GeoJSON file, i.e., vector data, as mentioned in Algorithm 4. Two layers of GeoJSON data, the 2020 and 2023 layers, are added to the QGIS environment. Set operations are then performed using Python and QGIS integration for change detection.

After performing the set operations as in Fig. 16. The intersection of 2020 and 2023 vector data gives the No change in the buildings. The difference between 2020 and the intersection gives the demolished buildings. Similarly, the difference between the 2023 vector data and the intersection gives new buildings.

Fig. 17 (a), (b) and (c) represents the 3 images as the layers of GeoJSON.

The updating of GIS maps in QGIS is shown in Fig. 18. To acquire the change detection of buildings, several set operations are used for the vector building segmentation layers Mumbai 2020 GeoJSON and Mumbai 2023 GeoJSON. Green denotes newly constructed buildings, Orange denotes existing structures, and red denotes structures that have been demolished.

4.2. Performance analysis

The Training and Testing loss throughout the Training Epochs is shown in Fig. 19. While the Testing loss line is depicted in red, the Training loss line is colored yellow.

The Training and Testing Accuracy during the training epochs is shown in Fig. 20. For 50 epochs, the Testing accuracy line is plotted in red, while the Training accuracy line is plotted in yellow. The accuracy of the model improves with the number of epochs.

The testing accuracy is about 77.6 % The IoU metric as given in Equation (17), which measures how well the model predicts the segmentation masks for the training and testing sets during the training epochs, is displayed in Fig. 21.

$$IOU = \frac{TP}{(TP + FP + FN)} \tag{17}$$

The Mumbai dataset is used to validate the model, and the validation accuracy is roughly 89 %. According to Fig. 22, accuracy grows as the number of epochs does.

When an image is given as test input it will detect the type of building and each is color labeled as follows, the Background of the image is yellow colored, the industry is Grey colored and Residential buildings are purple colored with Mean IOU as given in Equation (14) is 0.42

$$MeanIOU = \sum_{i=1}^{N} IOU_i / N \tag{18}$$

The performance of the suggested system during the phases of training, testing, and validation is shown in Table 3.

The system's performance is contrasted with that of the current methods in Table 4.

In Table 4, a comparison with other models is presented. It is important to note that some of the other models used on different datasets may have higher accuracy compared to our model. However, it should be noted that in India, buildings are typically tightly packed and both residential and industrial buildings look similar when viewed from a satellite, which can lead to a decrease in the accuracy of our model.

With our dataset, U-Net++ achieved an accuracy of 67 %, while VGG-19 achieved 73 %, and U-Net with ResNet50 achieved 79 % accuracy.

Our model achieved 88.5 % accuracy on the LEVIR-CD (Mohammad et al., 2022) dataset, 84.7 % accuracy on the Space Net (Van Etten et al., 2018) dataset, and 89.6 % accuracy on the WHU Buildings datasets (Ji et al., 2018), with 89 % accuracy on the Nashik dataset.

Fig. 23 represents the performance evaluation of a model trained on a dataset consisting of three types of buildings: Residential, Industrial, and Holy Places. The matrix provides valuable insights into the model's accuracy and false positive rate.

The accuracy of the model on the training dataset is calculated to be 84.59 %. This metric indicates the proportion of correctly classified instances out of the total predictions made by the model.

Furthermore, the false positive rate as given in Equation (19) is determined to be 16.19 %.

$$FPR = \frac{FalsePostive}{FalsePositive + TrueNegative} \tag{19}$$

The false positive rate of 16.19 % indicates the percentage of times the model wrongly classified residential buildings as industrial. Fig. 24 shows that the model achieved an accuracy of 78.34 % on the testing

dataset. This means that the model correctly classified 78.34 % of the instances. However, the false positive rate for the same dataset was determined to be 21.82 %.

Fig. 25 represents the confusion matrix for the validation dataset. The validation accuracy is about 89.22 %. The False Positive Rate is 9.41 %.

5. Conclusion and future work

Life on Earth and building structures are constantly changing. Rapid changes, especially in areas like urban planning, disaster response, environmental monitoring, security, and real estate management, need to be identified and stored in records. Using pre-processing techniques and QGIS, we divided SAS Planet imagery into patches to train a model with an 85 % accuracy rate, which predicted output accuracy between 80 % and 90 %. However, in Maharashtra, where residential buildings and industries look alike, the model could not classify them accurately.

This analysis has practical applications that empower urban planners with detailed insights for informed decisions on infrastructure, zoning, and resource allocation. Local communities can benefit from enhanced civic participation and community-led development through access to environment information, while knowledge-sharing among government agencies, research institutions, and the public promotes collaboration for improved urban development.

Our research has encountered challenges in classifying buildings with complex shapes, as the model's performance tends to decrease when presented with non-rectangular or atypical geometries. The model's sensitivity to building orientation is another limitation that may affect its performance when presented with satellite images of buildings with varying orientations. In densely populated urban areas, the high density of buildings, complex spatial arrangements, and overlapping structures can pose challenges for accurate classification, thereby limiting the model's ability to generalize. We will overcome these challenges in our future work. It also involves expanding the categories to include mosques, churches, and other places in urban areas, such as mixed-use buildings or informal settlements. A similar analysis will be done on the remaining categories. Future work also concentrates on making the model robust with additional shape features.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The Authors would like to thank Dr K.VenugopalaRao, Scientist SG (Retd), Group Director Urban studies NRSC, ISRO, Dr SC Jayanthi, Scientist SG, Group Head, Urban Studies and Applications (USAG), National Remote Sensing Centre (NRSC), Ms J.KAMINI, Sci/Eng. SF HEAD, Urban Studies DIV. NRSC, Ms. ReedhiShukla, Scientist/Engineer 'SE', Urban Studies Division, NRSC, ISRO, for providing the required data during evaluating the model.

References

Aghayari, S., et al., 2023. Building detection from aerial imagery using inception resnet unet and unet architectures. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 10, 9–17.

Akçay, H. G., Aksoy, S. Detection of compound structures using hierarchical clustering of statistical and structural features. In 2011 IEEE International Geoscience and Remote Sensing Symposium (pp. 2385-2388).

Alsabhan, W., Alotaiby, T., Dudin, B., 2022. Detecting buildings and nonbuildings from satellite images using U-Net. Comput. Intell. Neurosci. https://doi.org/10.1155/2022/4831223.

Alsabhan, W., TurkyAlotaiby., 2022. Automatic building extraction on satellite images using unet and resnet50. Comput. Intell. Neurosci.

Ansari, M.A., Kurchaniya, D., Dixit, M., 2017. A comprehensive analysis of image edge detection techniques. Int. J. Multimedia Ubiquitous Eng. 12 (11), 1–12.

Erdem, F., UğurAvdan., 2020. Comparison of different U-net models for building extraction from high-resolution aerial imagery. Int. J. Environ. Geoinformatics 7 (3), 221–227.

Goldblatt, R., You, W., Hanson, G., Khandelwal, A., 2016. Detecting the boundaries of urban areas in india: a dataset for pixel- based image classification in google earth engine. Remote Sens. (Basel) 8 (8), 634.

He, K., Zhang, X., Ren, S., Sun, J., Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 10.1109/CVPR.2016.90.

https://www.census2011.co.in/census/city/361-nashik.html last accessed on 29 September 2023.

Huang, Y., Zhuo, L., Tao, H., Shi, Q., Liu, K., 2017. A novel building type classification scheme based on integrated LiDAR and high- resolution images. Remote Sens. (Basel) 9 (7), 679.

Ji, S., Wei, S., Meng, L.u., 2018. Fully convolutional networks for multi-source building extraction from an open aerial and satellite imagery dataset [J]. IEEE Trans. Geosci. Remote Sens. https://doi.org/10.1109/TGRS.2018.2858817.

Kaichang, D., Deren, L., Deyi, L., 2000. Remote sensing image classification with GIS data based on spatial data mining techniques. Geo- Spatial Information Sci. 3 (4), 30–35.

Kang, J., Körner, M., Wang, Y., Taubenböck, H., Zhu, X.X., 2018. Building instance classification using street view images. ISPRS J. Photogramm. Remote Sens. 145, 44–59.

Katpatal, Y.B., Kute, A., Satapathy, D.R., 2008. Surface-and air-temperature studies in relation to land use/land cover of Nagpur urban area usingLandsat 5 TM data. J. Urban Plann. Dev. 134 (3), 110–118.

Kay, S., Hedley, J.D., Lavender, S., 2009. Sun glint correction of high and low spatial resolution images of aquatic scenes: a review of methods for visible and near-infrared wavelengths. Remote Sens. (Basel) 1 (4), 697–730.

Livne, M., Rieger, J., Aydin, O.U., Taha, A.A., Akay, E.M., Kossen, T., Madai, V.I., 2019. A U-Net deep learning framework for high performance vessel segmentation in patients with cerebrovascular disease. Front. Neurosci. 13, 97.

Lloyd, C.T., Sturrock, H.J., Leasure, D.R., Jochem, W.C., Lázár, A.N., Tatem, A.J., 2020. Using GIS and machine learning to classify residential status of urban buildings in low- and middle-income settings. Remote Sens. (Basel) 12 (23), 3847.

Mohammad, A., Gullapalli, O. S., Vasavi, S., Jayanthi, S., Updating of GIS maps with Change Detection of Buildings using Deep Learning techniques, 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India,2022, pp. 1-6, 10.1109/INCOFT55651.2022.10094545.

Aquib Ansari, M., Kurchaniya, D., Manish D., A comprehensive analysis of image edge detection techniques, published in 2017 by SERSC; ISSN: 1975-0080 IJMUE.

Reda, K., Kedzierski, M., 2020. Detection, classification, and boundary regularization of buildings in satellite imagery using faster edge region Convolutional neural networks. Remote Sens. (Basel) 12 (14), 2240.

Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv.org. Available at: https://arxiv.org/abs/1505.04597.

Van Etten, A., Lindenbaum, D., & Bacastow, T.M. (2018). SpaceNet: A Remote Sensing Dataset and Challenge Series. ArXiv, abs/1807.01232.

Wurm, M., Schmitt, A., Taubenböck, H., 2015. Building types' classification using shape-based features and linear discriminant functions. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 9 (5), 1901–1912.

Zhang, T., Tang, H. Built-up area extraction from Landsat 8 images using Convolutional neural networks with massive automatically selected samples. In Pattern Recognition and Computer Vision: First Chinese Conference, PRCV 2018, pp. 492-504, Springer International Publishing.

Zhao, P., Miao, Q., Song, J., Qi, Y., Liu, R., Ge, D., 2018. Architectural style classification based on feature extraction module. IEEE Access 6, 52598–52606.