Advancing GIS map maintenance: change detection and update using ResU-Net: a case study on Chandigarh and Hyderabad cities, India

S. Vasavi[1,]*, Dibyaranjan Parida[1], N. Malathi[1], M. Sobhana[1] and Reedhi Shukla[2]

[1]Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada 520 007, India
[2]National Remote Sensing Centre, ISRO, Hyderabad 500 037, India

India's metropolitan cities have been growing rapidly for many years. To keep geographical information accurate and current, it is essential to update GIS maps. Traditionally, experts have analysed new data sources and made necessary adjustments to the maps manually. Such manual monitoring is a laborious test both economically and in terms of workforce. Geographical data are transformed into digital maps by GIS mapping, making it simple to spot patterns, trends and linkages. Extraction of humanmade objects, such as roads, water bodies and buildings, from remotely sensed imageries holds significance in various urban applications, including urban land-use and land-cover assessment, geographical database updates and change detection. Cartosat-3 data can provide detailed information about buildings and their changes over time. Additionally, GIS maps are manually updated by rasterizing vector data. The suggested system consists of ResNet and U-Net architecture as its core. The bi-temporal images are initially co-registered to completely align 2020 and 2022 satellite images with respect to the coordinates. Buildings are then segmented using U-Net with ResNet as the backbone, and the resultant segments are converted from raster to vector format. The suggested model has been tested and trained using the Chandigarh dataset, which resulted in an accuracy of 95%.

Keywords: Change detection, digital maps, geographical data, remote sensing, urban planning.

URBAN planning relies heavily on buildings, water bodies and roads. They significantly impact various aspects of urban development, including land use, transportation, infrastructure and the overall liability of urban areas. Information on these objects is crucial for transportation, urban land registry and disaster/hazard risk assessment, to name just a few applications[1,2]. For urban planners, object segmentation from satellite images is a useful tool.

'Satellite imagery' is a term used to describe images of the Earth or other planets captured by humanmade satellites. Higher resolution for remotely sensed images, particularly satellite images (e.g. Cartosat-3 images have a resolution of 0.28 m), is now possible due to the rapid advancement of sensor technology. In the previous 20 years, numerous studies have been conducted to accurately extract aspects from these incredibly exact images, such as buildings, roads and water bodies. If the satellite imagery has been spatially corrected to ensure good alignment with other datasets, it can be merged with vector or raster data in a geographic information system (GIS)[3,4].

The very high resolution satellite (VHRS) images are raster images in the PNG, JPG and GeoTIFF formats[5]. The file format GeoTIFF has a three-letter extension (.tif). It is a specific kind of TIFF file that includes GeoTIFF tags as the spatial referencing data for geo-referenced raster imagery. Vector data are used to generate the GIS maps. Since vector data offer a more precise and accurate representation of the real-world characteristics and objects the maps intend to represent, they are crucial for updating GIS maps[6].

To enable accurate and insightful analysis when working with multiple satellite images obtained at various periods or from various sensors, it is crucial to co-register or align them spatially. The geometrical disparities between images are addressed via co-registration, allowing for efficient comparison, integration and analysis. Geometric distortions are corrected, spatial consistency is maintained, and satellite images are aligned to a common reference system through co-registration. It also enables accurate comparison, change detection, fusion and integration of multiple datasets by bringing images into a common reference frame[7,8].

Deep learning methods for updating GIS maps can offer insightful information and improve the precision and effectiveness of the mapping process. Convolutional neural networks (CNNs)[6,7] and recurrent neural networks (RNNs) are examples of deep learning algorithms that can be applied to a variety of tasks in GIS map updating, such as semantic segmentation, object detection and image classification. This automatic analysis accelerates the map-updating process and requires less human labour, enabling more frequent updates and real-time data[8].

An encoder and a decoder network make up the common picture segmentation architecture known as U-Net. A deep learning architecture called ResU-Net (Residual

*For correspondence. (e-mail: vasavi_movva@vrsiddhartha.ac.in)

U-Net) can be used to update GIS maps[9]. U-Net (a convolutional neural network architecture) and residual networks (ResNet) are combined in the ResU-Net design. An encoder and a decoder network make up the common image segmentation architecture known as U-Net. The encoder extracts features from the input image at various abstraction levels, and the decoder reconstructs the segmented image by applying these features. The introduction of residual connections by ResNet, on the other hand, helps solve the vanishing gradient issue during training and enables the efficient learning of deep architectures.

The objectives of the present study are as follows:

- To develop an automated and efficient approach to update GIS maps with the change detection of objects.
- Assess the performance of the enhanced ResU-Net model that performs change detection of building objects.
- Analyse the impact of different input data sources.
- Assess the generalization ability of the model.
- Provide recommendations and guidelines.

Literature survey

Change detection in GIS maps is done using deep learning approaches, such as ResNet and U-Net models. The authors demonstrated the effectiveness of utilising an ensemble of ResNet and U-Net models for precise change detection and compared the performance of several deep-learning model[10]. Researchers have suggested a method for detecting building changes that combines GIS data with deep learning models like ResNet and U-Net. They have emphasized the advantages of utilizing ResNet for feature extraction and U-Net for precise segmentation when identifying changes to buildings[11]. Studies have compared various deep learning models, including ResNet and U-Net, for automatic change detection in GIS maps[12]. The authors evaluate the performance of different models and discuss the advantages of using an ensemble of ResNet and U-Net models for accurate and robust change detection. An ensemble learning method that combines various deep learning models for change detection in GIS data has also been suggested[13]. The study explored how ResNet and U-Net models can be used in an ensemble framework and showed superior building change detection. U-Net and ResNet models are used to semantically segment buildings in high-resolution satellite data. The significance of precise segmentation in the update process has been underlined, and the potential use of these models for developing change detection in GIS maps is addressed[14]. For water body extraction, a multi-task fully convolutional network (FCN) has been proposed. The method included border delineation as well as binary categorization of water and non-water areas, resulting in the precise mapping of water bodies. A convolutional neural network (CNN) architecture was suggested in this study for the semantic segmentation of water bodies. The model distinguished water bodies from satellite images with good accuracy[15]. For water body extraction, this work presented a deep residual U-Net architecture. To increase segmentation accuracy and manage complicated water body forms, the model used residual connections and U-Net structure[16]. For segmenting water bodies, a deep active learning system has been proposed. The method successfully reduced annotation efforts while retaining good segmentation accuracy by combining deep learning with active learning techniques[17,18].

Study area for training dataset generation

Chandigarh is known for its well-planned infrastructure, modern architecture and beautiful urban design. Its overall area is about 44 square miles, making this capital city one of the smallest Indian states. Chandigarh is situated 100 km east of Ludhiana and 260 km north of New Delhi, the national capital (Figure 1 *a*). Geographically, it is situated at 76.7794° E long. and 30.7333°N lat.

Study area for validation dataset generation

The proposed model was validated using Hyderabad. As a fast-growing city, Hyderabad has undergone substantial urban development and expansion throughout the years. The footprint of buildings in Hyderabad varies by area and neighbourhood. The city is situated at 17.366°N lat. and 78.476°E long. (Figure 1 *b*).

Data preparation

Data preparation is necessary regardless of the expected model performance. To improve model performance, datasets must be carefully prepared before use. The dataset was ordered from the Indian Space Research Organizations (ISROs) Bhoonidhi Portal, and QGIS was used to label the satellite image. ISRO has built the high-resolution Cartosat-3 Earth monitoring satellite. By combining high-resolution panchromatic images, the pan sharpening approach, as shown in Figure 2, improves the spatial resolution of multispectral imaging. Table 1 presents list of data and date of acquisition.

Data-splitting is the process of breaking up the annotated dataset into subsets for training, validation and testing. The validation set is used for model selection and hyperparameter tweaking, while the training set is used to train the deep learning model[19,20]. The performance of the model on untested data is assessed using the testing set. The distribution of building instances throughout the subgroups should be carefully considered.

Materials and methods

This section gives a summary of the approach, methods, dataset and suggested architecture for the planned system.

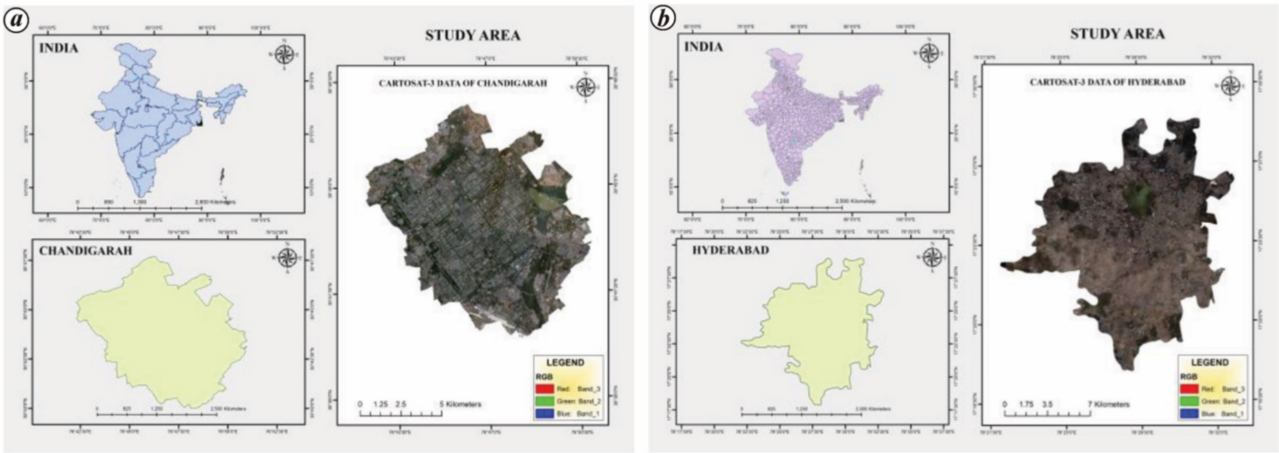

Figure 1. Study area for (***a***) training and testing and (***b***) validation.

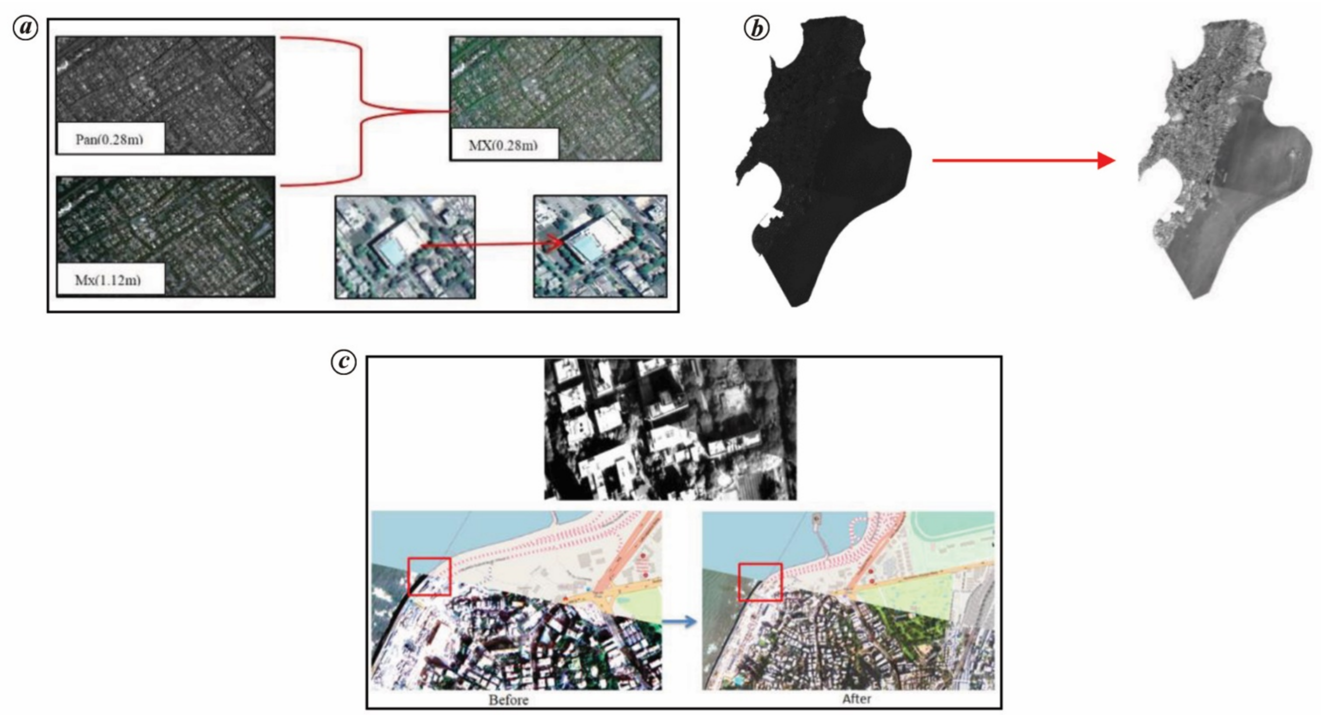

Figure 2. ***a***, Image enhancement (pan-sharpening). ***b***, Radiometric correction. ***c***, Geometric correction and coregistration.

Architecture

Figure 3 presents the enhanced ResU-Net model proposed in this study for performing change detection of buildings. The U-NET architecture and deep residual learning are both utilized by ResU-Net. For building extraction tasks, the ResU-Net design combines the strength of residual networks (ResNet) with the well-liked U-Net architecture. The U-Net architecture with a ResNet encoder employs a 7 × 7 input kernel size, three initial filters, and a max pooling size of 2 × 2 for down sampling. The right identity block characterized by its ability to maintain the dimensions of the input. By utilizing residual blocks, a deeper network can be constructed without dealing with vanishing or expanding gradient issues.

It also makes network training simple. Rich skip connections in ResU-Net facilitate greater information flow between layers, facilitating better gradient flow during training. Similar to a U-Net, ResU-Net is made up of a bridge linking the encoding and decoding networks. Two 3 × 3 convolutions are used by U-Net, and each is followed by a ReLU activation function. In ResU-Net, a pre-activated residual block fills the place of these layers. The network is assisted in learning an abstract representation by the encoder, which takes the input image and runs it through several encoder blocks. The pre-activated residual block is used to assemble the three encoder blocks. The output from each encoder block serves as a skip connection for the associated decoder block. The first 33 convolution layers employ a stride of 2 in the second and third encoder blocks to minimize the spatial dimensions (height and width) of the feature maps. When the stride value is 2, the spatial dimensions are cut in half, from 512 to 256.

ResNet50, the suggested approach, accepts input of 512 × 512 with three filters. Convolutional layers are used here for down sampling with a stride of 2, which results in halving the image size and doubling the filters. The convolutional layer conv_1, which is 3 × 3 and has a stride of 2, receives the input first. The max pooling layer, which is 2 × 2 and has a stride of 2, is then fed to the input. The conv_1 layer sends the output to the conv2_x layer. Each residual block that makes up the layer is constructed from two 3 × 3 convolutional layers employing batch normalization and ReLu activation functions; they are all connected by a shortcut connection. By acquiring more intricate information from the input image, networks can increase accuracy. This is done by repeatedly stacking residual blocks.

The energy function is derived from the pixel-by-pixel ReLu activation function and the Tversky loss function of the final feature map.

The segmentation performance of the network improves with decreasing values of the loss function, as the predicted mask matches the ground truth mask more closely. The weight map enables the network to concentrate on separating boundary-adjacent pixels, which is more crucial for precise segmentation. Table 2 provides the details of the modified ResU-Net model proposed in this study.

Table 1. List of data and date of acquisition (DOA)

Sat Id.	Place	Sensor	DOA
Cartosat-3	Chandigarh	PAN	5/12/2020
		MX	5/12/2020
		PAN	25/05/2022
		MX	25/05/2022
	Hyderabad	PAN	14/11/2020
		MX	14/11/2020
		PAN	20/12/2022
		MX	20/12/2022

Table 2. Comparison of the existing models with the proposed ResU-Net model

Parameters	Existing model	Modified model
Channels in input image	1	3
Shape of input image	256 × 256	512 × 512
Strides	1	2
Input kernel size	3 × 3	7 × 7
Initial no. of filters	64	3
Number of trainable parameters in a neural network model	52 million	73 million
Pooling type	PSP	Max pooling
Size of maximum pooling at every layer	2 × 2	2 × 2
No. of layers	27 U-Net	50 U-Net
Channels in output image	1	1
Activation function	Sigmoid	ReLu
Loss function	Dice	Binary cross Entropy loss
Encoder and decoder blocks	3	5
Dilation rate	1	1
Optimizer	ADAM	ADAM
Channels in output image	1	1

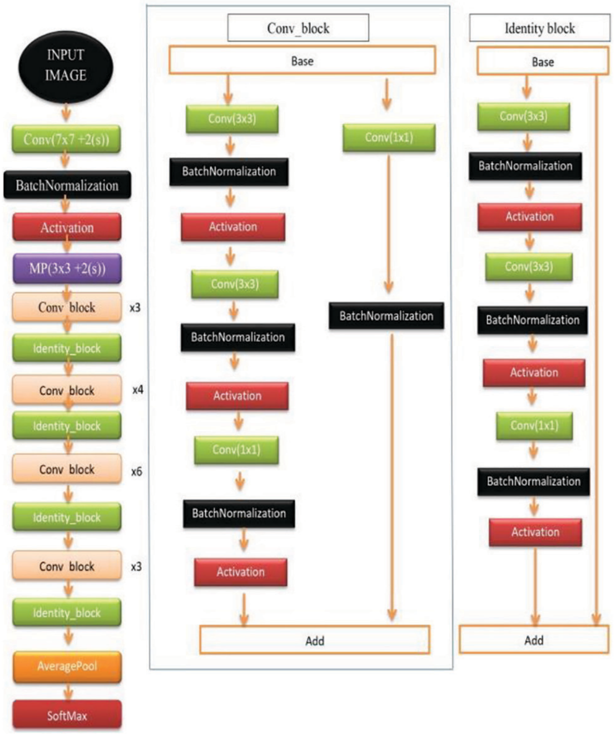

Figure 3. ResU-Net architecture.

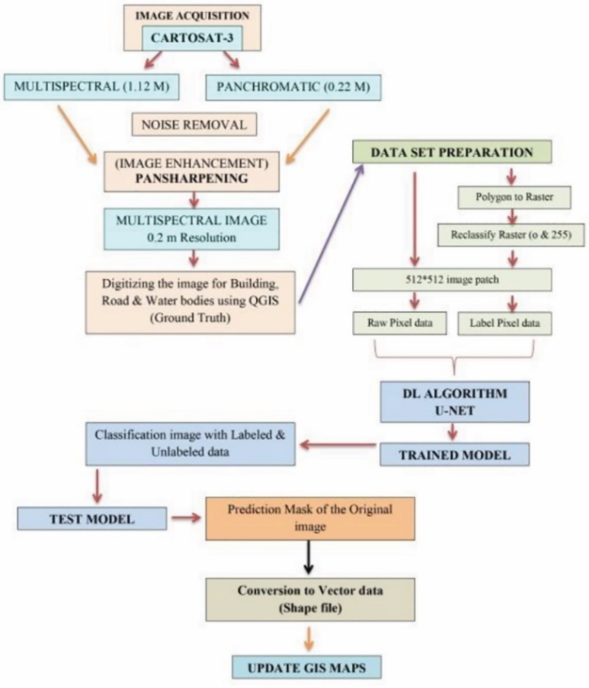
IMAGE ACQUISITION
CARTOSAT-3
MULTISPECTRAL (1.12 M)
PANCHROMATIC (0.22 M)
NOISE REMOVAL
(IMAGE ENHANCEMENT) PANSHARPENING
MULTISPECTRAL IMAGE 0.2 m Resolution
Digitizing the image for Building, Road & Water bodies using QGIS (Ground Truth)
DATA SET PREPARATION
Polygon to Raster
Reclassify Raster (o & 255)
512*512 image patch
Raw Pixel data
Label Pixel data
DL ALGORITHM U-NET
TRAINED MODEL
Classification image with Labeled & Unlabeled data
TEST MODEL
Prediction Mask of the Original image
Conversion to Vector data (Shape file)
UPDATE GIS MAPS

Figure 4. Flow chart of the proposal model.

Figure 5. Polygon to raster and reclassify raster.

Methodology

The numerous components that make up the proposed model are discussed in this section. Figure 4 presents the methodology of the proposed system, which consists of data preparation, co-registration, pre-processing, semantic segmentation, post-processing, geo-referencing, converting raster data to vector data, change detection and updating of GIS maps.

Noise removal: For noise removal, a Gaussian filter is used. By reducing or eliminating noise from the satellite data, Gaussian noise reduction techniques attempt to improve the image quality and its usability for a variety of applications. These methods efficiently reduce noise while retaining crucial image information by utilizing statistical aspects and filtering strategies. The image is successfully blurred during the convolution process, which also reduces high-frequency noise.

Labelling of data: This is a crucial step in preparing datasets for deep learning models. The size, complexity, resource availability and required level of accuracy are only a few examples of the variables that influence the choice of the labelling approach. QGIS is used to label data to prepare datasets for deep learning applications, even though it was mainly developed for spatial data analysis and mapping.

Polygon to raster and reclassify raster: Buildings were manually digitized, and the resulting footprints were used to turn the polygons into raster images, which were then used to create environments for the satellite image. In this phase, the geographic extent, coordinate system, cell size and spatial extent were all guaranteed. The rasterized building footprints were classed as ‘0’ or ‘255’, where ‘0’ denotes a non-building class and ‘255’ denotes a building class. This resulted in the generation of data marked with references to the building and non-building classes (Figure 5).

Image splitting: Image splitting in building extraction tasks enables more effective analysis of large and complex scenes, improves model performance and enhances the extraction of detailed building information. A feature or library called Patchify enables users to automatically divide an image into smaller patches or tiles. It offers a practical method to make patches of a specific size from the input image.

Semantic segmentation: Segmenting buildings is done using a modified U-Net design. The model was first trained using a 27-layer U-Net architecture. However, while forecasting the results, land and buildings were taken into account as buildings, which was reflected in the anticipated binary masks. ResNet50 is therefore considered as the encoder block for the U-Net architecture, and the model was trained as described above in order to enhance feature extraction and operate on a deeper network.

Regularize building footprint: The post-processing procedure known as regularizing building footprints after deep learning attempts to hone and enhance the precision of building footprint predictions produced from deep learning models. Although deep learning models may successfully identify and extract building footprints from aerial or satellite data, the resultant footprints would still need to be regularized to guarantee geometric correctness and consistency. The presence or absence of buildings in the input data is frequently indicated by pixel-level predictions or segmentation masks produced by deep learning algorithms for building footprint extraction. However, for various reasons, including data noise, model limits or complicated building geometries, these projections could have abnormalities, fractured borders or errors. Post-processing building boundaries using Douglas–Peucker algorithm involves simplifying detected outlines by strategically applying the algorithm with a selected threshold.

Geo-referencing: Giving geographical coordinates to digital photographs or maps is a crucial step in the GIS process, known as georeferencing. Buildings may be precisely identified and delineated using these models, which offer useful information for disaster preparedness, infrastructure management and urban planning.

Raster to vector conversion: Raster data, such as satellite images or the output of deep learning models, must be converted into vector format as part of the geospatial analysis process known as 'raster to vector conversion'. In the case of building extraction, where the objective is to transform pixel-level predictions or segmentation masks derived from deep learning models into geometrically precise vector representations of building footprints, this conversion is crucial. The conversion of raster to vector format is done using GDAL (Geospatial Data Abstraction Library).

Change detection: This classifies the changes in various object footprints over time. Bi-temporal vector images are the result of the conversion to vector data. Different set operations are used for both images to identify changes, which are then used to categorize structures as new, destroyed or unmodified.

Updation of GIS maps: For the GIS database to appropriately reflect the condition of the built environment, GIS maps for various objects must be updated. Upgrading GIS maps for building extraction includes adding the outcomes of building extraction operations to the current GIS map layers. The new GIS maps display the buildings precisely, aiding in the analysis and decision-making process across a range of field by providing valuable spatial information and insights.

Algorithms

Algorithm-1: Pan sharpening: Although numerous pan-sharpening methods exist, we employed the popular Brovey transform in this study. On dividing each band of the multispectral image by the total number of bands, it adjusts the intensity of each pixel individually. The resultant image is then scaled by the panchromatic image. The Brovey transform modifies the intensity pixel by pixel. Assume we have a panchromatic image P and a multispectral image with bands R, G and B, abbreviated as M_R, M_G and M_B respectively.

The fused image F_R, F_G and F_B can be determined as follows:

$$\text{F\_R} = \text{M\_R} * (P/(\text{M\_R} + \text{M\_G} + \text{M\_B})), \tag{1}$$

$$\text{F\_G} = \text{M\_G} * (P/(\text{M\_R} + \text{M\_G} + \text{M\_B})), \tag{2}$$

$$\text{F\_B} = \text{M\_B} * (P/(\text{M\_R} + \text{M\_G} + \text{M\_B})). \tag{3}$$

Algorithm-2: Image splitting: To enable deep learning models to be processed, evaluated and trained more rapidly involves dividing a large image into smaller patches or tiles as shown in eqs (4) and (5).

$$\begin{aligned}\text{num\_cols} = {} & (\text{width}/(\text{output\_width})) \\ & + (\text{width} \,\%\, \text{output\_width} > 0),\end{aligned} \tag{4}$$

$$\begin{aligned}\text{num\_rows} = {} & (\text{height}/(\text{output\_height})) \\ & + (\text{height} \,\%\, \text{output\_height} > 0).\end{aligned} \tag{5}$$

Algorithm-3: Regularize building footprint: In order to increase the precision and smoothness of the identified building footprints, regularization technique is frequently used. Depending on the method and regularization strategy used, several sets of individual equations may be employed as given below.

$$R(f) = \lambda * \sum \|\nabla f(i, j)\|, \tag{6}$$

where $f(i, j)$ denotes the gradient of the estimated building footprints at pixel (i, j), $R(f)$ signifies the regularization term and λ is the regularization parameter that determines the intensity of the regularization.

Algorithm-4: Conversion from raster to vector: The following algorithm describes how the raster data is transformed into suitable vector data for additional processing:

Step 1: Read the image in 'TIF' format 1 to get the CRS data.
Step 2: Polygonalize the raster image developed from the CRS data using GDal.
Step 3: Change the coordinate reference system for all polygonized raster geometries in an active geometry column.
Step 4: The vector representation of the raster image is thus obtained.

Evaluation metrics

A range of measures, including precision (P), recall (R), crossroads over union (IoU, Jaccard index), and F1 score (Dice coefficient), can be used to assess the effectiveness of a model. The mathematical equations for some popular deep learning assessment metrics are given below

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}), \tag{7}$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}), \tag{8}$$

$$\text{IOU} = \text{TP}/(\text{TP} + \text{FP} + \text{FN}), \tag{9}$$

$$\begin{aligned}\text{F1 score} = {} & (2{*}\text{Precision}{*}\text{recall})/ \\ & ((\text{Precision} + \text{recall})).\end{aligned} \tag{10}$$

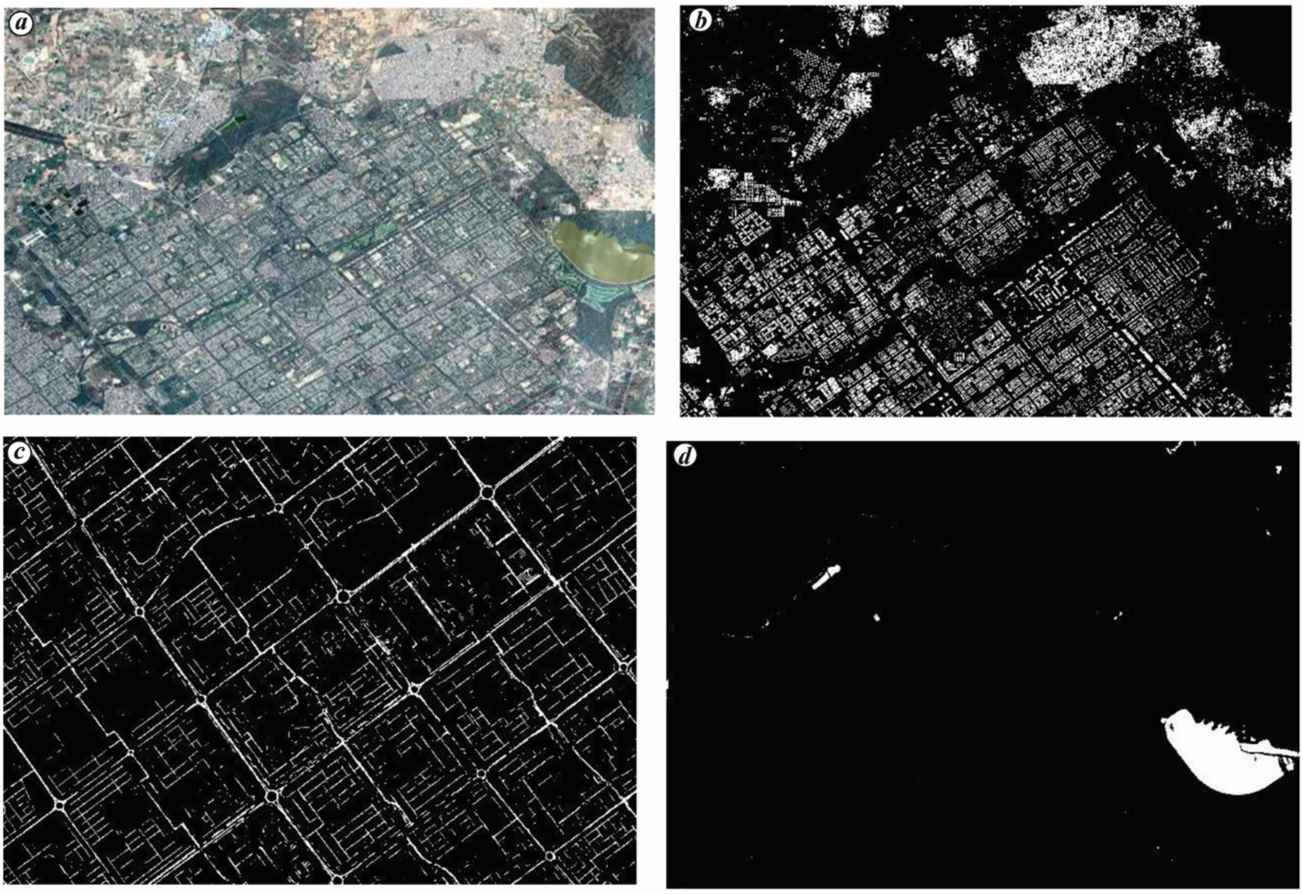

Figure 6. (***a***) RGB image, (***b***) predicted image of building, (***c***) predicted image of road and (***d***) predicted image of water bodies.

where TP is the true positive, FP the false positive and FN is the false negative.

Results and discussion

The successful implementation of the proposed system yields several positive outcomes. The system is designed to accept input from bi-temporal images of Chandigarh (2020 and 2022). The enhanced ResU-Net model successfully detected and classified changes in the bi-temporal images of Chandigarh city with an accuracy of 97% on the test dataset as shown in Figure 6. This highlights the ability of the model to recognize and record architectural changes. Additionally, the validation accuracy of 97% validates the robustness and generalizability of the model in capturing changes in unseen data, according to the study. The binary mask developed for an individual image was combined with an image from the 2022 dataset. This process included the post-processing stage, during which morphological techniques like dilation were applied after the prediction was completed. The goal of these post-processing step was to automatically recognize and categorize changes in building footprints over time, as illustrated in Figures 7 and 8.

Better outcomes can be obtained using powerful computing resources and training the model for longer epochs. The output of change detection of buildings results in these borders that resemble structures since the bi-temporal images under consideration are of various perspectives. The accuracy of the model increases as the number of epochs increases. The model begins with random weights throughout the early epochs and gradually modifies them to reduce the training loss. As the model is trained, it learns more about the patterns in the data and adjusts its parameters to provide more accurate results.

When epochs begin providing accurate findings, the loss continues to decrease. A dropping loss indicates that the model successfully identifies and learns the underlying relationships and patterns in the training data. The model improves its ability to predict outcomes by repeatedly adjusting its weights and biases during each epoch.

Figure 9 shows how the accuracy of the model increases as the number of epochs increases. Throughout the early epochs, the model starts with random weights and gradually changes them to decrease training loss. The model learns more about the patterns in the data as it is trained and adjusts its parameters to provide more accurate results. The loss graph shows that as the epochs provide proper results,

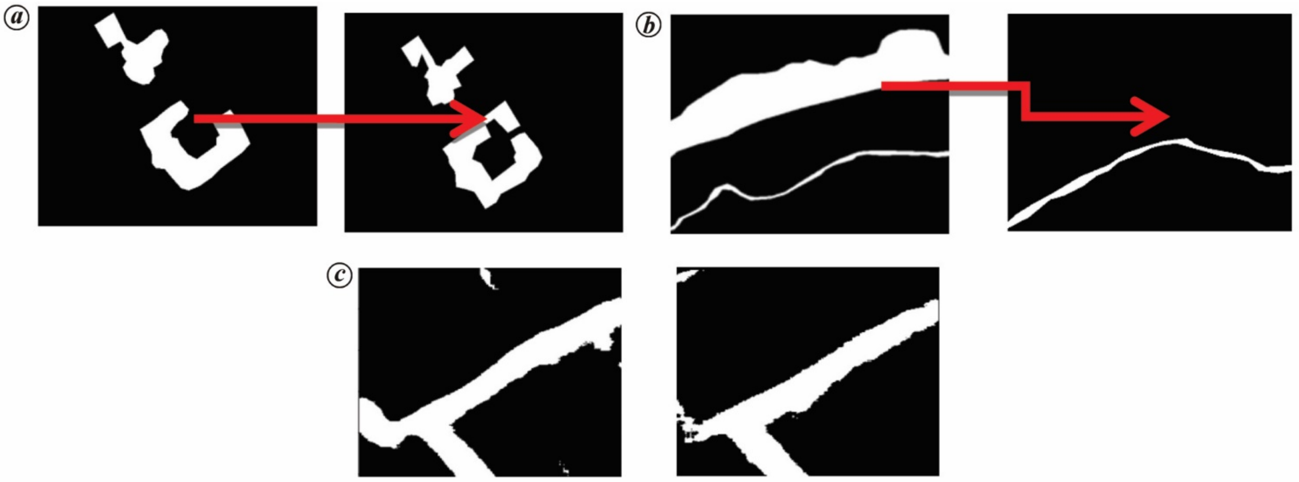

Figure 7. (***a***) Building change detection, (***b***) water body change detection, and (***c***) road change detection of predicted images in 2020 and 2022.

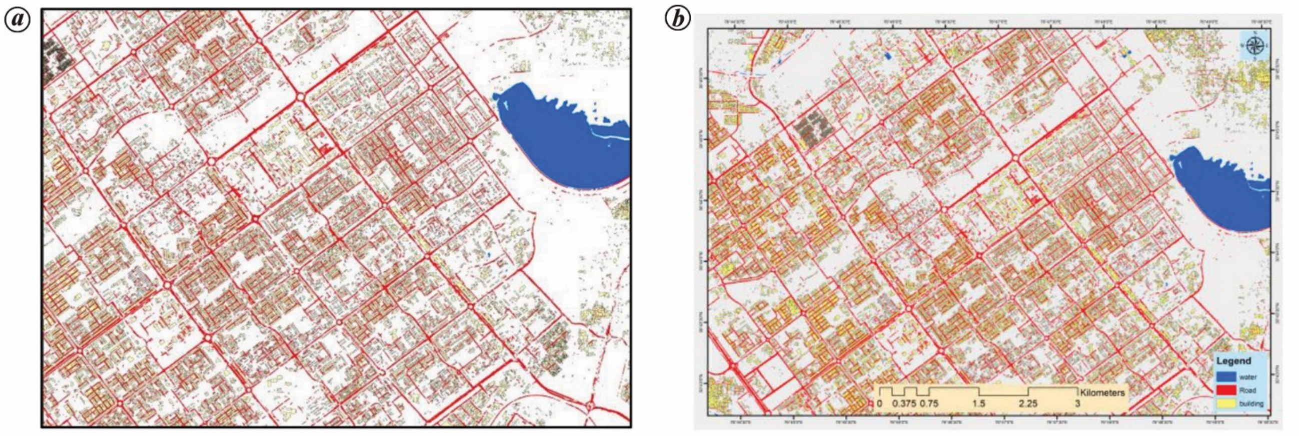

Figure 8. ***a***, Predicted buildings, roads, water bodies (regularized) and thereafter. ***b***, Updated GIS map.

Table 3. Evaluation results

Metrics	Testing on Chandigarh dataset	Validation on Hyderabad dataset
Buildings		
Loss	0.15	0.21
Accuracy (%)	0.92	0.91
Precision (%)	0.85	0.79
Recall (%)	0.75	0.73
IOU (%)	85	83
Roads		
Loss	0.19	0.20
Accuracy (%)	0.91	0.89
Precision (%)	0.73	0.75
Recall (%)	0.79	0.75
IOU (%)	85	84
Water bodies		
Loss	0.18	0.25
Accuracy (%)	0.91	0.89
Precision (%)	0.83	0.81
Recall (%)	0.85	0.81
IOU (%)	87	83

the loss decreases. A reducing loss implies that the model successfully identifies and learns the underlying relationships and patterns in the training data.

Table 3 shows the performance metrics of the proposed system for both Chandigarh and Hyderabad cities. These metrics offer a thorough assessment of the performance of the model in terms of both general accuracy and particular characteristics like precision, recall and IOU. The model is able to accurately categorize the bulk of samples, as evidenced by its high accuracy of 97% as shown in Figure 9. According to precision at 89%, 89% of all the anticipated positive samples include genuine positives. The ability of the model to accurately identify the positive samples from among all the real positive samples in the dataset is demonstrated by its 77% recall. A decent degree of spatial agreement between the anticipated and real building footprints is shown by the IOU metric, which measures the overlap between the expected and ground-truth zones, which is at 70%. Table 4 shows a comparison of the proposed system with existing models.

Table 4. Comparison of building detection with other models

Reference	Methodology	Accuracy metric	Accuracy	Dataset
Building				
21	FCCDN	Mean IOU	0.85	LEVIR-CD
22	O-GAN	Mean IOU	0.88	Tibet
23	U-Net	Mean IOU	0.92	Khartom
24	EGCTNet	Mean IOU	0.82	WHU-CD
25	MSCnet+ Few-Shot CNN	Mean IOU	0.92	LEVIR-CD
Proposed work	U-Net with ResNet Encoder	Mean IOU	0.92	Cartosat-3 (Chandigarh and Hyderabad)
Water bodies				
26	RetinaNet	Mean IOU	0.98	World view-2
27	MSFENet	Mean IOU	0.91	GF-2
28	U-Net	Mean IOU	0.89	Sentinel-1
29	NFANet	Mean IOU	0.90	Gaofen
30	NDWI-DeepLabV3+	Mean IOU	0.88	Sentinel-2A
Proposed work	U-Net with ResNet Encoder	Mean IOU	0.89	Cartosat-3 (Chandigarh and Hyderabad)
Road				
31	U-NET	Mean IOU	0.89	Siegfried map
32	Mnih-CNN	Mean IOU	0.88	Massachusetts data
33	Saito-CNN	Mean IOU	0.90	Massachusetts data
34	Coord-dense-global	Mean IOU	0.92	Gaofen-2
35	Deep Res UNet	Mean IOU	0.91	Massachusetts data
Proposed work	U-Net with ResNet Encoder	Mean IOU	0.93	Cartosat-3 (Chandigarh and Hyderabad)

Conclusion and future work

The study shows that deep learning approaches are highly suitable for precisely identifying and updating building footprints on GIS maps. Bi-temporal images of Chandigarh were used as input for the proposed system, which worked well. It allowed the ResU-Net model to attain a high accuracy of 97% on the test dataset. This shows that the model can recognize and categorize changes in building structures within the study region with accuracy. The robustness and generalizability of the model were further validated by the validation accuracy of 93%.

The findings of this study demonstrate the potential of deep learning models, particularly ResU-Net, to enhance and automate GIS map maintenance procedures. The accuracy and effectiveness of map maintenance operations are improved by the capacity of the model to automatically detect and update building changes. Numerous applications, including urban planning, infrastructure management and disaster response, can benefit from this.

Although these findings are encouraging, it is vital to recognize certain constraints. The characteristics of the study area, the accuracy of the data and the specifics of the implementation could affect how effectively the ResU-Net model works. More research and testing are necessary to examine the generalizability of the method to different geographical areas and datasets.

Conflict of interest: The authors declare that there is no conflict of interest.

1. Wieland, M., Martinis, S., Kiefl, R. and Gstaiger, V., Semantic segmentation of water bodies in very high-resolution satellite and aerial images. *Remote Sensing Environ.*, 2023, **287**, 113452; https://doi.org/10.1016/j.rse.2023.113452.
2. Mattyus, G., Luo, M. and Urtasun, R., DeepRoadMapper: extracting road topology from aerial images. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3438–3446.
3. Vasavi, S., Priyadarshini, N. K. and Harshavaradhan, K., Invariant feature-based Darknet Architecture for moving object classification. *IEEE Sensors J.*, 2021, **21**(10), 11417–11426; doi:10.1109/JSEN.2020.3007883.
4. Chandigarh Administration; https://chandigarh.gov.in/ (accessed on 12 March 2023).
5. Daudt, R. C., Le Saux, B., Boulch, A. and Gousseau, Y., Urban change detection for multispectral earth observation using convolutional neural networks. In IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 2018, pp. 2115–2118; doi:10.1109/IGARSS.2018.8518015.
6. Saito, S., Yamashita, T. and Aoki, Y., Multiple object extraction from aerial imagery with convolutional neural networks. *J. Imaging Sci. Technol.*, 2016, **60**(1); https://doi.org/10.2352/J.ImagingSci.Technol.2016.60.1.010402.
7. Volpi, M. and Tuia, D., Dense semantic labeling of sub-decimeter resolution images with convolutional neural networks, 2016; https://doi.org/10.48550/arXiv.1608.00775.
8. He, K., Zhang, X., Ren, S. and Sun, J., Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778; doi:10.1109/CVPR.2016.90.
9. He, K., Zhang, X., Ren, S. and Sun, J., Identity Mappings in Deep Residual Networks, Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, Springer, Cham, 2016, vol. 9908; https://doi.org/10.1007/978-3-319-46493-0_38.
10. Alam, M., Wang, J. F., Guangpei, C., Yunrong, L. V. and Chen, Y., Convolutional neural network for the semantic segmentation of remote sensing images. *Mobile Netw. Appl.*, 2021, **26**, 200–215; https://doi.org/10.1007/s11036-020-01703-3.
11. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L., DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018, **40**(4), 834–848; doi:10.1109/TPAMI.2017.2699184.

12. Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H., Encoder–Decoder with Atrous Separable Convolution for Semantic Image Segmentation. Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Springer, Cham, 2018, vol. 11211; https://doi.org/10.1007/978-3-030-01234-2_49.
13. Chen, L. C., Papandreou, G., Schroff, F. and Adam, H., Rethinking atrous convolution for semantic image segmentation, arXiv:1706.05587v3; https://doi.org/10.48550/arXiv.1706.05587.
14. Goodfellow, I., Bengio, Y. and Courville, A., *Deep Learning*, MIT Press, 2016.
15. An, S. and Rui, X., A high-precision water body extraction method based on improved lightweight U-Net. *Remote Sensing*, 2022, **14**(17), 4127; https://doi.org/10.3390/rs14174127.
16. Isikdogan, S., Bovik, A. and Passalacqua, P., Surface water mapping by deep learning. *IEEE J. Selec. Top. Appl. Earth Observ. Remote Sensing*; 10.1109/JSTARS.2017.2735443.
17. Moradkhani, K. and Fathi, A., Segmentation of water bodies in remote sensing images using deep stacked ensemble model. *Appl. Soft Comput.*, 2022, **124**, 109038; https://doi.org/10.1016/j.asoc.2022.109038.
18. Zhang, Y., Li, X., Wang, Z. and Chen, H., Deep learning-based water body extraction from remote sensing images using multi-modal data. *IEEE J. Select. Top. Appl. Earth Observ. Remote Sensing*, 2021, **14**, 3315–3327; doi:10.1109/JSTARS.2021.3063114.
19. Xu, Y., Xie, Z., Feng, Y. and Chen, Z., Road extraction from high-resolution remote sensing imagery using deep learning. *Remote Sensing*, 2018, **10**(9), 1461; https://doi.org/10.3390/rs10091461.
20. Xia, L., Chen, J., Luo, J., Zhang, J., Yang, D. and Shen, Z., Building change detection based on an edge-guided convolutional neural network combined with a transformer. *Remote Sensing*, 2022, **14**(18), 4524; https://doi.org/10.3390/rs14184524.
21. Chen, P., Hong, D., Chen, Z., Yang, X., Li, B. and Zhang, B., FCCDN: feature constraint network for VHR image change detection, *ISPRS J. Photogram. Remote Sensing*, 2022, **187**, 101–119; https://doi.org/10.1016/j.isprsjprs.2022.02.021.
22. Sun, S., Mu, L., Wang, L., Liu, P., Liu, X. and Zhang, Y., Semantic segmentation for buildings of large intra-class variation in remote sensing images with O-GAN. *Remote Sensing*, 2021, **13**(3), 475; https://doi.org/10.3390/rs13030475.
23. Li, W., He, C., Fang, J., Zheng, J., Fu, H. and Yu, L., Semantic segmentation-based building footprint extraction using very high-resolution satellite images and multi-source GIS data. *Remote Sensing*, 2019, **11**(4), 403; https://doi.org/10.3390/rs11040403.
24. Daudt, R. C., Le Saux, B., Boulch, A. and Gousseau, Y., Urban change detection for multispectral earth observation using convolutional neural networks. In IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 2018, pp. 2115–2118; doi:10.1109/IGARSS.2018.8518015.
25. Khoshboresh-Masouleh, M. and Shah-Hosseini, R., Deep few-shot learning for bi-temporal building change detection, arXiv, 2021; doi:10.48550/ARXIV.2108.11262.
26. el Hoummaidi, Lala, Abdelkader, L. and Khan, A., Automatic extraction of water bodies from high-resolution remote sensing imagery using deep learning and GIS in Dubai. *SSRN Electronic J.*, 2021, 10.2139/ssrn.3914039.
27. Liu, B., Du, S., Bai, L., Ouyang, S., Wang, H. and Zhang, X., Water extraction from optical high-resolution remote sensing imagery: a multi-scale feature extraction network with contrastive learning. *GIScience Remote Sensing*, 2023, **60**, 1; doi:10.1080/15481603.2023.2166396.
28. Jeon, H., Kim, D.-J. and Kim, J., Water body detection using deep learning with Sentinel-1 SAR satellite data and land cover maps. In IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 2021, pp. 8495–8498.
29. Lu, M., Fang, L., Li, M., Zhang, B., Zhang, Y. and Ghamisi, P., NFANet: A Novel Method for Weakly Supervised Water Extraction from High-Resolution Remote Sensing Imagery, arXiv: 2201.03686; https://doi.org/10.48550/arXiv.2201.0368.
30. Peng, Y., Su, H. C., Xu, C., Feng, A. and Liu, T., NDWI-Deep-Labv3+: high-precision extraction of water bodies from remote sensing images. In Proceedings of the 3rd International Conference on Machine Learning and Machine Intelligence, 2020, pp. 110–116; https://doi.org/10.1145/3426826.3426847.
31. Jiao, C., Heitzler, M. and Hurni, L., A fast and effective deep learning approach for road extraction from historical maps by automatically generating training data with symbol reconstruction. *Int. J. Appl. Earth Observ. Geoinf.*, 2022, **113**, 102980.
32. Mnih, V. and Hinton, G. E., Learning to detect roads in high-resolution aerial images. Computer Vision – ECCV 2010. Lecture Notes in Computer Science, Springer, 2010, vol. 6316; https://doi.org/10.1007/978-3-642-15567-3_16.
33. Saito, S., Yamashita, T. and Aoki, Y., Multiple object extraction from aerial imagery with convolutional neural networks. *J. Imaging Sci. Technol.*, 2016, **60**(1); https://doi.org/10.2352/J.ImagingSci.Technol.2016.60.1.010402.
34. Wang, S., Yang, H., Wu, Q., Zheng, Z., Wu, Y. and Li, J., An improved method for road extraction from high-resolution remote-sensing images that enhances boundary information. *Sensors*, 2020, **20**(7), 2064; https://doi.org/10.3390/s20072064.
35. Zhang, Z., Liu, Q. and Wang, Y., Road extraction by deep residual U-Net. *IEEE Geosci. Remote Sensing Lett.*, https://doi.org/10.48550/arXiv.1711.10684.

Received 28 June 2023; revised accepted 10 October 2023

doi: 10.18520/cs/v126/i3/326-335