J. Vis. Commun. Image R. 77 (2021) 103141

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Full length article

Deep multi-feature learning architecture for water body segmentation from satellite images☆

Rishikesh G. Tambe [a,*], Sanjay N. Talbar [a], Satishkumar S. Chavan [b]

[a] SGGS Institute of Engineering and Technology, Nanded, Maharashtra, 431606, India
[b] Don Bosco Institute of Technology, Kurla (W), Mumbai, Maharashtra, 400070, India

ARTICLE INFO

Keywords:
Convolutional neural network
Deep learning
Refinement modules
Satellite image analysis
Water body extraction

ABSTRACT

Automatic water body extraction from satellite images of various scenes is a classical and challenging task in remote sensing and image interpretation. Convolutional neural network (CNN) has become prominent option for performing image segmentation task in remote sensing applications. However, CNN-based networks have non-trivial issues for segmenting such as: (1) blurring boundary pixels; (2) large number of trainable parameters; and (3) huge number of training samples. In this paper, we propose an end-to-end multi-feature based CNN architecture, called as W-Net, to perform water body segmentation. W-Net consists of contracting/expanding networks and inception layers. W-Net takes advantage of contracting network to capture context information while localization is achieved with expanding network. With these networks, W-Net is able to train on less number of images and extract water pixels accurately. Use of inception layers reduces computational burden within the network by decreasing total number of trainable parameters. W-Net incorporated two refinement modules to enhance predicted results which mitigate blurring effect and to inspect continuity of boundary pixels. Dataset consisting 2671 images with manually annotated ground truths are built to validate performance and effectiveness of our proposed method. In addition, we evaluated our method on crack detection dataset where W-Net achieved competitive performance with Deepcrack. W-Net accomplished excellent performance on the water body dataset ($I/U = 0.9434$ and $F - score = 0.9509$).

1. Introduction

Water resources such as ocean, rivers, lakes, streams and reservoirs are important in preserving and controlling various life sources in and around ecosystem [1]. Rapid urbanization is accelerating change and damage to available water resources. To conserve and take benefit from available water resources, constant monitoring and proper survey is necessary for getting information about water bodies [1]. Manual survey provides trustworthy information but is costly in terms of man-power and is time consuming process. In recent years, advancement in remote sensing has made it possible to use information recorded by different sensors on-board of satellite in timely manner. These factual knowledge is widely used for various applications including extracting information about water bodies.

For remote sensing imagery, water body extraction is aimed to discriminate water bodies from other non-water body structures. Satellite images are more complex in nature which consist of other information including man-made structures, forest, snow, barren lands and so on which makes water body extraction difficult and challenging. As a result, traditional methods based on water body indexes [2–5] lead to problem of misclassifying water pixels as non-water pixels and inaccurately identify boundary pixels due to tedious task of selecting threshold. In last few years, use of convolutional neural networks (CNN) has tremendously increased for various application in remote sensing [6–11]. Many researchers have worked on salient object detection (SOD) to locate noticeable and eye-catching object regions in images [12–16] and videos [17,18]. Recently, the accuracy of SOD has been improved extensively due to advancement and use of deep CNN. SOD aims at identifying individual object instance in the detected salient region while segmentation of water body is performed on whole image. CNN have been widely used and have achieved great success in SOD [12–18].

Recently, there have been use of CNN in extracting water bodies using satellite images as those reported in [19–23]. As disparate water bodies exhibit different spectral characteristics, it is difficult to design adaptable method to deal with water bodies in various scenes. Li et al. [22] worked only with images having sea and land structures

☆ This paper has been recommended for acceptance by Zicheng Liu.
* Corresponding author.
E-mail address: rishitambe@gmail.com (R.G. Tambe).

https://doi.org/10.1016/j.jvcir.2021.103141
Received 21 November 2020; Received in revised form 5 February 2021; Accepted 1 May 2021
Available online 5 May 2021

in same scene whereas [23] deals with raw Landsat images at once which required high-end computing facility. CNN-based architectures like [24–26] often resulted in blurred output and generally failed in discriminating objects which resulted in degraded segmentation. These architectures have huge amount of trainable parameters. Water body extraction methods based on CNN provide high accuracy but are still not in practical use because of huge requirement of relevant datasets and high computational complexity due to large number of trainable parameters. CNN based methods performed better than index based methods but extraction of water body accurately and preserving boundary pixels remains a challenge.

In this paper, we propose a water body extraction system (W-Net) based on deep CNNs which accurately extracts water pixels and preserves continuous boundary pixels. To verify and enhance the water body extraction and boundary pixels, two refinement modules are implemented on predicted images. W-Net requires less number of training images and produces accurate segmentation. To deal with computational complexity of the architecture, asymmetric convolutions are used to reduce trainable parameters. The main contributions of our proposed method are summarized as follows.

1. We propose an end-to-end multi-feature CNN architecture for extracting water bodies, named W-Net, for segmenting water pixels from non-water pixels.
2. Inception blocks are added at both contracting (encoders) path and expanding (decoders) path so as to aggregate features extracted at different scales using asymmetric convolutions.
3. Computational complexity within the network hugely depends on total number of trainable parameters. To reduce these trainable parameters, we use asymmetric convolutions which made our network as sparse and more refined architecture.
4. The refinement modules are used not only for enhancing predicted images but also assist in examining continuity as well as discontinuity of boundary pixels of water body. In addition, refinement modules pull out non-water pixels which are predicted as water pixels and are not identified prior.
5. W-Net shows its potency and robustness by showing impressive results on differing datasets. We validated W-Net on crack detection dataset without fine-tuning.

The remainder of this paper is organized as follows. In Section 2, we review recent related work. The proposed W-Net architecture is described in Section 3. Section 4 gives details about experiments, including implementation steps, dataset, metrics and cross-dataset evaluation. We conclude our work in Section 5.

2. Related work

Over the past decades, a series of approaches have been published for extracting water bodies for remote sensing data. Most commonly used approach for water body extraction is based on water spectral indices. One of the most well-known approach is normalized difference water index (NDWI) [2,3] which makes use of near-infrared band (NIR) and visible green band of Landsat imagery to extract water bodies. However, this approach limits itself in extracting water bodies within the complex scenes. Xu [4], proposed modified NDWI (MNDWI) index in which NIR band was replaced by mid-infrared (MIR) band of Landsat imagery. After enhancement of water features, water bodies were separated by using threshold value. The selection of the optimal threshold value is very difficult and varies from less complex to more complex scenes. Feyisa et al. [27], presented automated water extraction index (AWEI) which makes use of multiple bands. However, result for complex scenes were not up to the mark and manually optimal threshold value was selected based on commission and omission error rate. To overcome these limitations of above index-based methods, Guo et al. [5], recommended use of weighted NDWI (WNDWI) which provided higher overall accuracy (OA) by adopting two different ways of selecting the optimal threshold value. All above index-based methods for extracting water bodies show good results but still selecting optimal threshold value remains a challenging task.

To overcome drawbacks of these index-based methods researchers [28,29] had worked on spatial features for extracting water body by using neighbourhood features of the centre pixel. Huang et al. [30], presented two level machine learning framework for extracting water bodies in high-resolution (HR) remote sensing images. Here, at first level, water bodies were extracted at pixel level, at second level, water body types were identified by using geometrical and texture features. However, pixel level extraction of water bodies may lead to errors due to manually selected threshold. Yao et al. [31], discussed automated urban water extraction method (UWEM) which combined water index with building shadow detection method. Wu et al. [32], preferred combining urban water index (UWI) for water body detection and urban shadow index (USI) for removing the shadow pixels from extracted water bodies. In both the cases [31,32] threshold was decided by using support vector machine (SVM), statistical learning technique followed by building shadow detecting method. These approaches were restricted to extract water bodies from urban scenes and cannot be treated as general method for scenes exhibiting different spectral and spatial characteristics.

In recent years, deep learning frameworks (DLF) has obtained a great attention in research community. DLF has been successfully used in various applications of remote sensing [33]. Long et al. [26], performed semantic segmentation based on fully convolutional networks (FCN). In FCN, each decoder up-sampled its feature map which was combined with analogous encoder feature map to yield the input to next decoder. Due to large size of FCN, training was performed at different levels. As number of convolution and pooling operations was large, at different levels, convolution layers obtained variety of relevant features. Zheng et al. [34] suggested use of conditional random fields (CRF) with re-current neural network (RNN), collectively called as CRF-RNN. The predictive performance of FCNs is enhanced by CRF-RNN at the cost of fine-tuning on huge dataset [35]. Further, Lin et al. [36] tried to improve FCNs by introducing task partitioning model for ship detection while in [37] multi-scale semantic labelling scheme was introduced for sea–land–ship segmentation. Li et al. [20], presented FCNs for water body extraction in very high resolution images by analysing thirty six combination of various parameters and selected the best-FCN model. However, this best-FCN model was restricted in many ways such as: extraction of narrow rivers, tested only on Gaofen-2 images with 0.8 metre spatial resolution, and use of scenes which had only flat terrain.

All above FCN networks under-segments boundary pixels of water body. To deal with misclassification of boundary pixels, Miao et al. [19] suggested restricted respective field deconvolutional network (RRF DeconvNet) for water body segmentation. Iskdogan et al. [38], recommended use of DeepWaterMap architecture which can discriminate water body from snow/ice, shadow and clouds. This model was trained on Landsat imagery and showed significant results. However, DeepWaterMap failed in segmenting very high resolution images of urban areas. Iskdogan et al. [23], proposed modified version of DeepWaterMap, namely DeepWaterMapV2 which was memory efficient for larger size inputs but totally ignored computational burden (total number of trainable parameters are over 37 million) within the network. In addition, it was tested and reviewed only on Landsat images. Cheng et al. [39], advocated structured edge network (SeNet) for sea–land segmentation by integrating DeconvNet [40] with structured edge network and showed impressive segmentation resulted along with edge map. However, SeNet greatly depended on large number of annotated data. Badrinarayanan et al. [25], proposed SegNet for image segmentation which was based on encoder (down-sampling) and decoder (up-sampling) architecture and reduced trainable parameters to great extent. Ronneberger et al. [24], came up with network (Unet) constructed on encoder–decoder architecture and was used for different

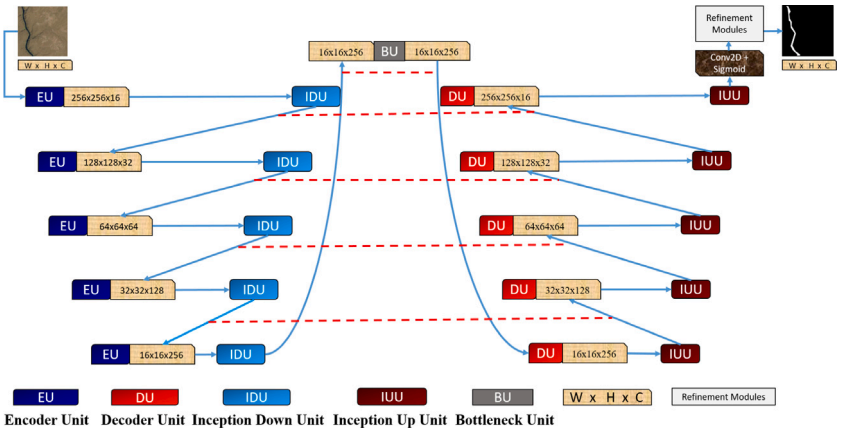

Fig. 1. An illustration of the W-Net architecture. In this architecture, there are five encoder unit (EU, dark blue), five inception down unit (IDU, light blue), bottleneck unit (BU, grey), five decoder unit (DU, light red), five inception up unit (IUU, dark red) followed by convolutional layer. Refinement modules are applied on predicted images to improve its quality.

biomedical segmentation applications based on semantic pixel-wise labelling. In Unet, data augmentation was performed excessively by applying elastic deformation on training images which permitted the network to learn invariance to deformations. Therefore, precise segmentation results were achieved even though Unet was trained on very few images.

Further, many researchers came up with different variants of U-net for various application in remote sensing such as classification [41,42], building detection [43–45], road detection [46–48] and water body extraction [21,23,49]. Feng et al. [21], proposed deep convolutional encoder–decoder (DCED) architecture for water body extraction from very high resolution images. However, Deepunet model predicted results which suffered from blurring boundary pixels of water body due to up-sampling. To mitigate this blurring effect, fully connected CRF (FCCRF) model was employed [50]. Due to smoothing phenomena, FC-CRF failed in preserving local structure information of water body and its boundary pixels [50]. To overcome this problem, regional restriction (RR) was used which improved prediction with enhanced water body boundaries. Gonzalez et al. [49], suggested water body segmentation by using TernausNet [43]. In which knowledge transfer-based model was used in-order to map high resolution labelled images with very high resolution images. Due to different distribution of spectral and spatial features of high resolution and very high resolution images, prediction results are not desirable. Apart from this issue, model failed to perform segmentation at several occasion including improper segmentation of scenes with water body having different spectral features. Further, knowledge-transfer from high resolution to very high resolution images did not improve the efficiency and performance of the segmentation. Most of the above mentioned architecture showed admirable performance but at the cost of large number of trainable parameters which increased computational complexity. Therefore, the proposed W-Net is motivated by Ronneberger et al. [24], Szegedy et al. [51], Szegedy et al. [52] and successfully compete with other methods by its excellent performance.

3. Proposed method

In this section, we present detail architectural description of our proposed method. Further, we will explore inception layers which improved the performance and mitigated computational complexity within the network.

3.1. W-Net architecture

In this section, we formulate water body segmentation (W-Net) as a binary image labelling task, where “1” and “0” refer to “water pixel” and “non-water pixel”, respectively. This task requires high level and low level features [53]. Fig. 1, illustrates the overall architecture of the proposed W-Net. It performs two tasks: convolution and deconvolution, which resembles encoder–decoder model. W-Net is computationally efficient and aggregates hierarchical attributes obtained from multiple convolutional layers.

As shown in Fig. 1, encoder unit (EU) and inception down unit (IDU) are more or less symmetric to the decoder unit (DU) and inception up unit (IUU), respectively. The network is W-shaped architecture and represents water body extraction, hence, named as ’W-Net’. $W \times H \times C$ (width, height and number of channels) represent dimensions of the image. Each EU and DU are connected to its corresponding inception unit (IDU and IUU), respectively through solid blue lines. The skip connections are also added from the IDU to IUU and denoted by solid-dash red lines. Fifth IDU is connected to corresponding DU through bottleneck unit (BU). The EU down-samples the input image and the IDU is inserted after each EU. BU outputs feature maps whose dimensions are same as its input. DU up-samples the input image and IUU block is added after each DU preserving the size of feature maps throughout the network. The final output of IUU is passed through sigmoid followed by refinement modules for final binary output. To improve quality of predicted image, we incorporated two refinement modules.

3.2. Structure of EU, IDU and BU

Fig. 2(a), (b) and (c) illustrate detail structure of EU, IDU and BU, respectively. Each EU consists of two 3×3 filter convolutional layers followed by a max pooling layer with 2×2 filter and stride of two (Fig. 2(a)). Here, each convolutional layer is comprised of convolutional operation (conv2D), batch normalization (BN) and Rectified Linear Unit (ReLU) [54,55]. BN is used to mitigate internal covariant shift and avoid overfitting problem [55]. To learn a non-linear task, avoid saturation and induce sparsity during learning process [56], ReLU (activation function $f(x) = max(0, x)$) is used. The pooling layer with 2×2 pixel filters is used to carry out spatial pooling along with down-sampling the input image by a stride of two. The filter channels are increased by factor of two after each EU. In [43–45], variants of U-net using VGGNet [57] were proposed for image segmentation. All these variants have fascinating feature of architectural simplicity but at the cost of high computational complexity and memory requirements. Li et al. [58,59] tried to alleviate computational complexity but reducing total number of trainable parameters remained as a challenge. This is due to use of 3×3 receptive field size in convolutional layers. In W-Net, we introduced use of inception block i.e. IDU and IUU after each

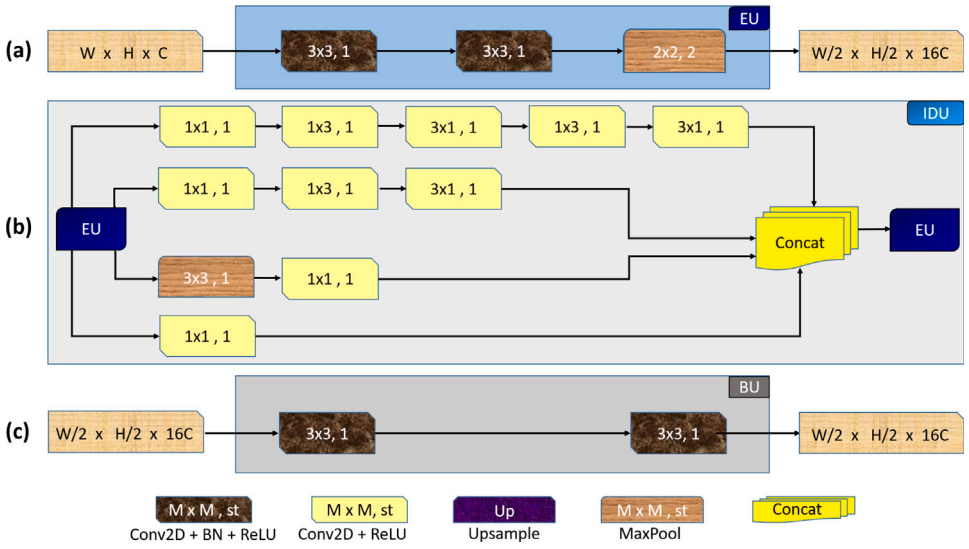

Fig. 2. An illustration of: (a) EU, (b) IDU, and (c) BU.

EU and DU to reduce number of trainable parameters which relieved computational burden during training of the network.

The proposed W-Net consists of two different inception units: five IDU corresponding to five EU and five IUU corresponding to DU, respectively as shown in Fig. 1. The feature map yield by EU is given as input to its analogous IDU. Fig. 2(b) shows detail structure of IDU. Each convolutional layer in IDU consists of conv2D and ReLU. The max pooling layer with 3×3 filter is for spatial pooling with stride of one followed by conv2D with 1×1 filter. Therefore, we performed spatial factorization of 3×3 filter conv2D into 1×3 filter conv2D followed by 3×1 filter conv2D using asymmetric convolutions [52]. Similarly, 5×5 filter conv2D is replaced by two 3×3 filter conv2D and then depth-wise separable convolutions on each 3×3 is performed as shown in Fig. 2(b). Use of depth-wise separable convolutions reduces computational cost within the network which results in less number of trainable parameters and faster training of the network [52]. In our case number of trainable parameters during training are 10.33 million which is significantly less as compared to [22,24,59]. Here, 1×1 convolution has dual-purpose, one for dimensionality reduction for efficient computing before asymmetric convolutions and second use of non-linear activation (ReLU) which allows W-Net to learn more complex function. Then the feature maps from multiple branches (at different scales) are aggregated to form single-output feature map and given as input to next EU. The output of fifth IDU is input to BU whose detail structure is shown in Fig. 2(c). BU consists of two 3×3 filter convolution layers with same spatial resolution followed by first DU.

3.3. Structure of DU and IUU

Fig. 3(a) illustrates detail structure of DU. Each DU up-samples the feature map to that of input image followed by 3×3 filter transpose convolutional layer. Here, each transpose convolutional layer consists of convolutional operation (conv2DT) and ReLU. Then the up-sampled feature map is given as input to IUU followed by next DU. The only difference between IDU and IUU is the use of transpose convolutional layer instead of convolutional layer as shown in Fig. 3(b). The final feature map generated by fifth IUU, same size to that of input image, is applied to output convolutional layer with 1×1 filter conv2D and sigmoid activation function.

3.4. Asymmetric convolutions

Fig. 4 illustrates the total number of parameters generated by different convolutions with variable filter size. Values marked with red colour represents parameters at individual convolutions while green colour denotes total number of parameters of the layers. $256 \times 256 \times 3$ is the size of the input image while $128 \times 128 \times 3$ is the image after applying variable filters (f) with stride (s). Fig. 4(a) demonstrates use of 5×5 convolution which will result in total 608 learnable elements. Learnable parameters are calculated using $((m \times n \times t + 1) \times k)$. where, m and n are width and height of the filter, respectively. t and k are the number of filters used in previous layer and current layer while 1 is the bias term for each filter. With suitable factorization of convolutions, we can achieve disentangled parameters which will allow our network to train faster [52]. Also, only increasing the network depth-wise will not ensure the performance of the network. Hence, balancing the number of filters per stage and the depth of the network can contribute towards improved performance of the network. Therefore, we replace 5×5 filter convolution with two 3×3 convolutions as shown in Fig. 4(b) (expanding the network width-wise). It can be observed that the learnable elements at individual convolutions (3×3) are 224 and total number of parameters are 448. This clearly show that, factorizing the convolutions can decrease number of trainable parameters to increase computational efficiency within the network. Further, in Fig. 4(c) we factorize each 3×3 convolutions by using asymmetric i.e. 1×3 and 3×1 convolutions. The number of trainable parameters at individual convolutions using asymmetric convolution is 80 while total number of trainable parameters are 320. This setup clearly reduces the total number of parameters by almost 52.63% with two prior setup of 5×5 and 3×3 convolutions.

3.5. Refinement modules

Two refinement modules are applied to predicted images: contrast stretching for enhancement and Gaussian filtering (refinement module one) followed by Canny edge detection [60] to obtain edge information (refinement module two). Fig. 5(a), (b), and (c) show input image, ground truth, and edge mask, respectively.

Fig. 5(d) shows predicted image obtained using W-Net which is enhanced using refinement module one to get filtering output (Fig. 5(e)). The enhancement factor for refinement one was set to 4.0 while the sharpness factor was set to 50.0 for all the test images. Thereafter, these enhanced images are smoothened by using filter of size 3×3 and 5×5. Further, filtering output is given as input to refinement module two to get the edge output image (Fig. 5(f)). Identifying boundary pixels for water body is crucial and important. During refinement module two, non-relevant pixels are flagged as weak boundary pixels while

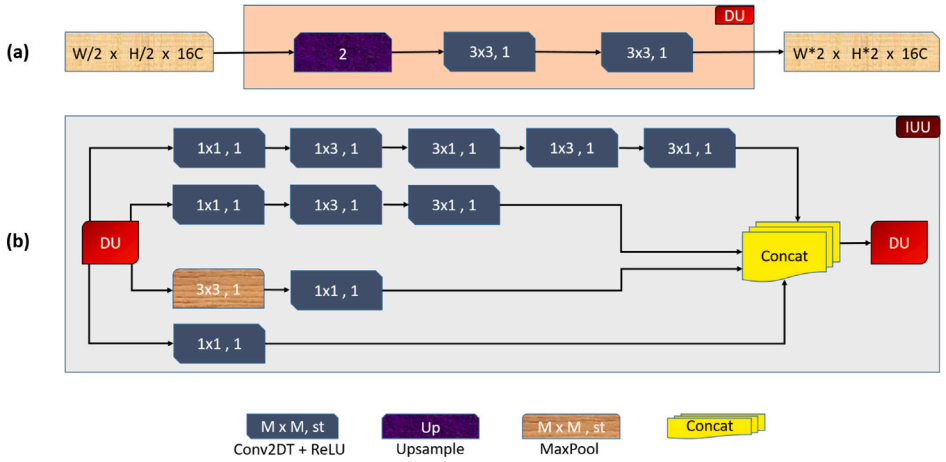

Fig. 3. An illustration of: (a) DU and (b) IUU.

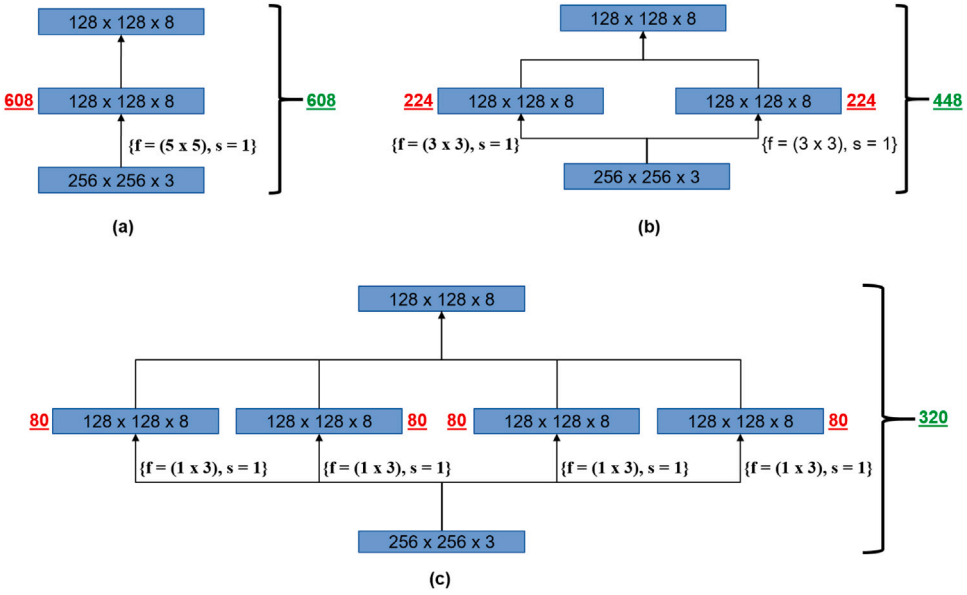

Fig. 4. An illustration of alternatives for variable size of the filters and its effect on the total number of trainable parameters: (a) 5 × 5 filter, (b) 3 × 3 filter and (c) Asymmetric convolutions (1 × 3 and 3 × 1 filters). f and s denotes filter size (width and height) and stride, respectively.

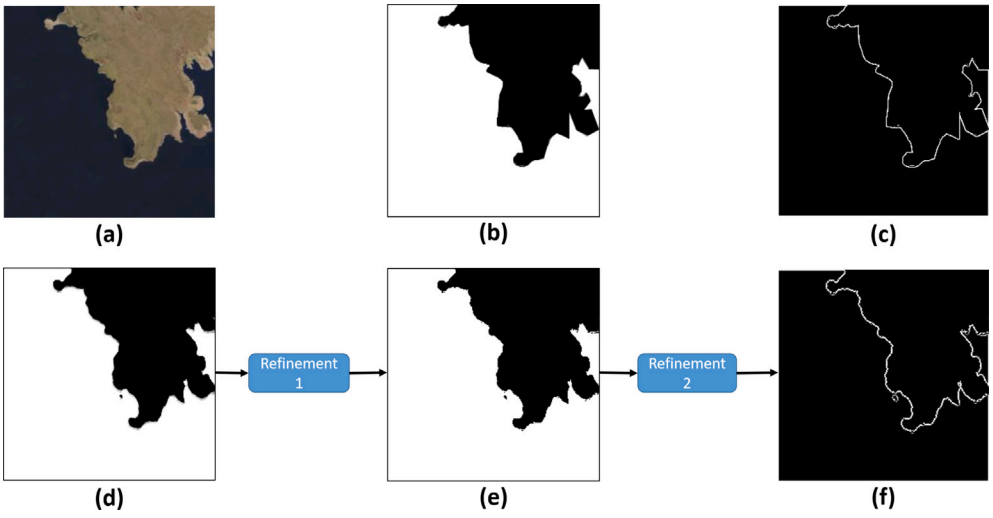

Fig. 5. Refinement Modules: (a) input image, (b) ground truth, (c) edge mask, (d) predicted image is given as input to refinement module one, (e) filtered output, and (f) refinement module two takes filtered output as input to produce edge output.

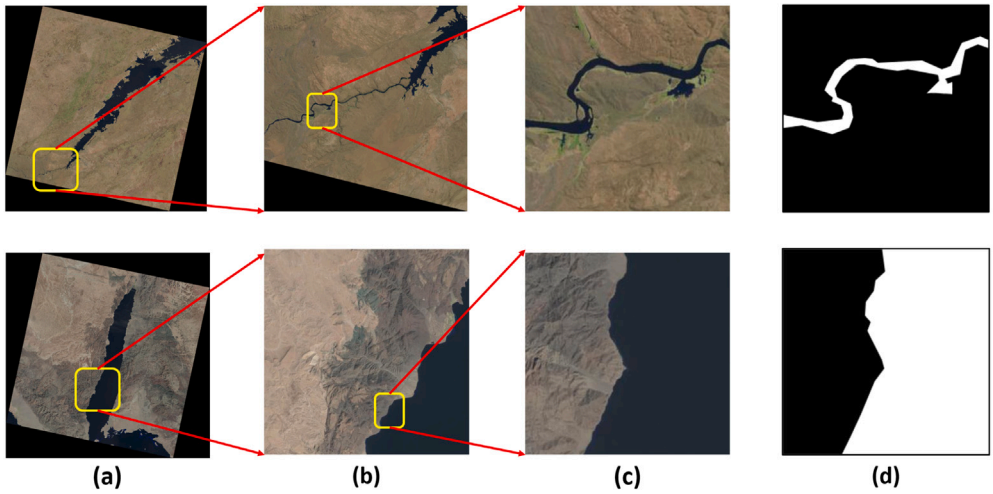

Fig. 6. Illustration of generating image patches and their ground truths from raw Landsat −8 images. Each column presents: (a) raw input images, (b) cropped images with region containing water body, (c) image patch with dimension 256 × 256, and (d) ground truth images.

relevant pixels are flagged as strong boundary pixels for identifying the boundary of the water body. The advantage of refinement modules is to make the predicted image visually appealing and sharpen the image to investigate edge information for further analysis. In edge output image we can effortlessly identify and audit misclassified boundary pixels.

4. Experiments

To demonstrate the efficiency and performance of the proposed W-Net, we evaluated it qualitatively as well as quantitatively on Landsat −8 images. Furthermore, we validated robustness of the presented W-Net by conducting cross-dataset evaluation. In addition, we also examined W-Net through ablation study.

4.1. Implementation

W-Net is trained using publicly available tensorflow [61]. The model parameters that are tuned for W-Net: input image of size $(256 \times 256 \times 3)$, the ground truth $(256 \times 256 \times 1)$, batch size (2) with number of epochs (200) and optimizer (Adam) [62] with the learning rate $(1e-4)$. First, we performed data augmentation like image translation, rotation, flipping, etc. on each patch [63]. We trained W-Net with augmented data having $(256 \times 256 \times 3)$ dimensions. Convolutional and deconvolutional layers of IDU and IUU take a random initialization weight based on [64]. Most of the existing networks perform semantic segmentation with multiclass labelling [35,43,44,65]. Isikdogan et al. [23], Yang et al. [48], Liu et al. [58] aimed to distinguish two classes but with distinct datasets, which are different than our dataset. On account of this reasons we trained W-Net efficiently without exploiting any pre-trained models. All experimental work is carried out using Intel(R) Xeon(R) CPU E5-2695V4 at 2.10*GHz* with 64GB RAM and NVIDIA TITAN Xp with 12GB GDDR5X (Micron) memory, CUDA 9.1 edition.

4.2. Data

Many existing networks were trained and evaluated on specific dataset recorded from various sensors to prove its efficiency. Isikdogan et al. [23,38] used Landsat −8 images for training and testing model, Cheng et al. [39] used natural-colour images from Google Earth, Kim et al. [42] used aerial images, Gonzalez et al. [49] used Sentineal-2 images. Therefore, it is very difficult to compare different models on same dataset. To make the comparison easy, we collected several images from USGS[1] of Landsat −8 (Operational Land Imagery (OLI) sensor) satellite which are with 30 metre multi-spectral, spatial resolutions along a 185 km swath. Landsat −8 OLI sensor produces nine bands in the spectrum of visible light and Near Infrared. The reflectance spectra for water body is exhibited highly in band 2, 3 and 4 of Landsat −8 OLI sensor. Therefore, we selected combination of three bands which include band 2, 3 and 4 for our experimentation. The original Landsat −8 images are of high dimensions i.e. typically of 7751 × 7891 × 3 and require high-end computation for processing. To avoid high computations within the network, we created patches of 256 × 256 dimensions. We selected region (yellow bounded box) within raw images which comprises variety structures of water bodies as shown in Fig. 6(a). Then we cropped the selected region containing water body as shown in Fig. 6(b). Finally, non-overlapping patch of dimension 256 × 256 (Fig. 6(c)) is cropped which is given as input to the network. In the similar way, total 2671 non-overlapping patches are created along with their ground truths (Fig. 6(d)) using GIMP tool.[2] Many images are purposefully selected which consist of shadows and mountain ridges to make the dataset more challenging. The dataset also included images with narrow and wider river-structures and sea–land structures. The dataset consists of non-overlapping 2071 training images, 500 validation images and 100 testing images.

Fig. 7 and Fig. 8 shows some representative samples from the dataset along with ground truth and edge mask, respectively. We purposefully selected these samples such that scene contains river-like (narrow and wider) structures and non-river structures such as sea–land scenes, island scenes, in-island scenes, and river-bed scenes. All the ground truths are manually annotated (Fig. 7(b), (d), and (f)) while edge masks are created from ground truths using [60] (Fig. 8(b), (d), and (f)).

4.3. Evaluation metrics

We evaluated performance of the proposed W-Net on our established water body dataset. To measure the performance of W-Net, we introduced three common metrics which are used for semantic segmentation: global accuracy (G), class average accuracy (C) and mean intersection over union (I/U) [26]. G measures the percentage of the pixels predicted correctly and C denotes predictive accuracy over all classes calculated using (1), (2), (3).

$$G = \sum_i m_{ii} / \sum_i n_i \tag{1}$$

[1] USGS: https://earthexplorer.usgs.gov/

[2] GIMP Tool: https://www.gimp.org/

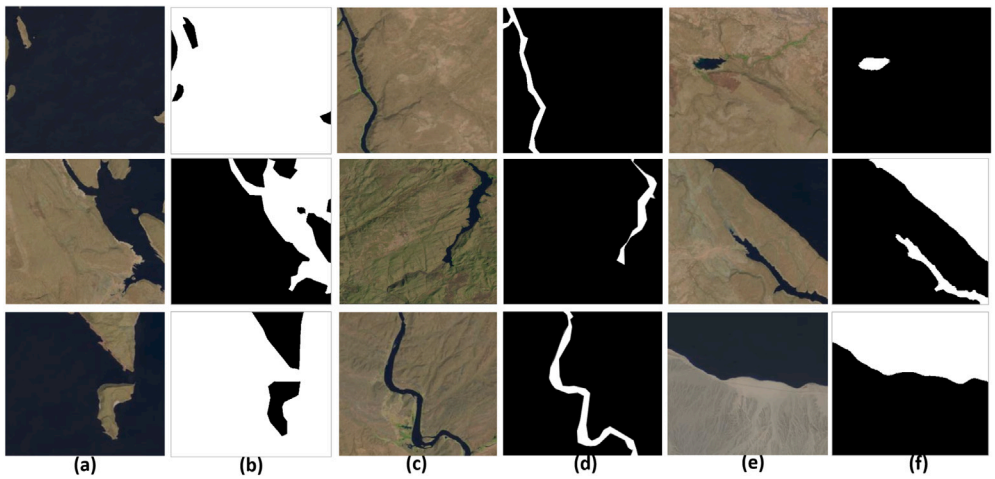

Fig. 7. Sample input images and their ground truths. In each column we present: (a), (c), and (e) with input images, (b), (d), and (f) with manually annotated ground truths. Samples are chosen carefully so that they contain both non-river structures (columns (a) and (e)) and river-like structures (column (c)).

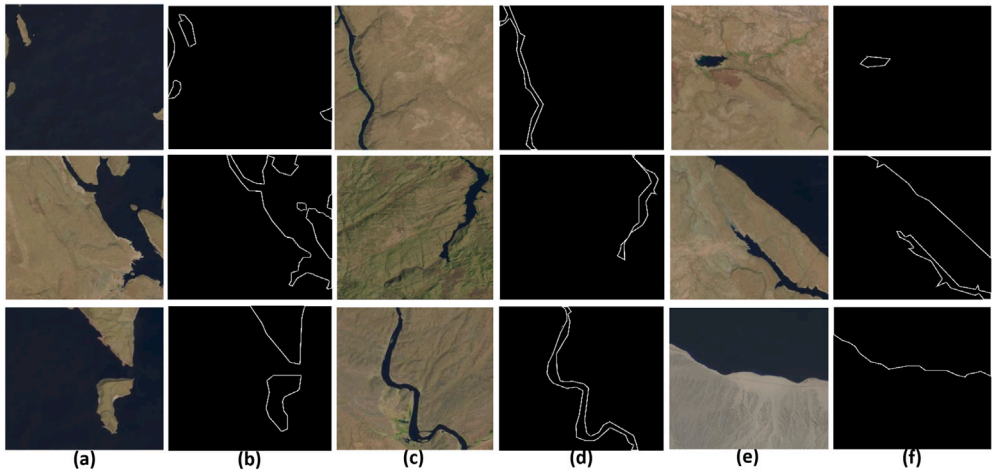

Fig. 8. Sample input images and edge masks. In each column we present: (a), (c), and (e) with input images, (b), (d), and (f) with edge masks. All the edge masks are created using ground truths.

$$C = \left(\frac{1}{m_c}\right) \sum_i \frac{m_{ii}}{n_i} \tag{2}$$

$$I/U = \left(\frac{1}{m_c}\right) \sum_i \frac{m_{ii}}{n_i + \sum_i m_{ji} - m_{ii}} \tag{3}$$

Here, m_{ij} is the number of pixels of the class i predicted to be the class j, m_c are different classes, and n_i is number of pixels of the class i including both true positive (TP) and false positive (FP).

In addition, three common metrics preferred in water body extraction field are also used to assess the semantic segmentation which are computed using (4), (5), (6).

$$Precision(P) = \frac{TP}{TP + FP} \tag{4}$$

$$Recall(R) = \frac{TP}{TP + FN} \tag{5}$$

$$F - score(F) = \frac{2PR}{R + R} \tag{6}$$

Here, TP, FP and FN represents the number of true positives, false positives and false negatives, respectively.

4.4. Evaluation

The proposed W-Net is compared with Unet [24], Deepunet [22], DeepWaterMapV2 i.e. DeepWMV2 [23] and Deepcrack [59]. All networks used for comparison are fine-tuned on water body dataset.

Fig. 9 demonstrates comparison of state-of-the-art methods with proposed W-Net. Compared with other architectures, W-Net shows prominent performance. The predicted images obtained using Unet appear blur as compared to other methods. Deepunet performs slightly better than Unet but leads to pixel misclassification. However, Unet works better with scenes having non-river structure leading to less misclassification of pixels than scenes consisting river-like structures. Deepcrack is originally modelled for crack detection which leads to the facts that the river-like structures are segmented but shows very poor performance in identifying scenes having non-river structures. These indicates that low-level layers represents local features with smaller receptive fields while deeper layers increases false positives (non-water pixels) in scenes containing non-river structures. DeepWMV2 works better with images holding non-river structures while shows inferior performance with scenes having river-like structures. It fails to discriminate shadow and elevated mountain ridge (yellow bounded box) from water pixels as seen in Fig. 10. These pixels (shadow and mountain ridge) are either treated as water pixel or non-water pixels, in above case it is treated as non-water pixels. For scenes comprising of non-river structures, DeepWMV2 misclassified the land pixels as water pixels as shown in Fig. 11(a) while W-Net captures land pixels as non-water pixels as shown in Fig. 11(b).

Fig. 12 shows predicted images, filtering outputs and edge outputs after applying refinement one and two. Here, we have considered two

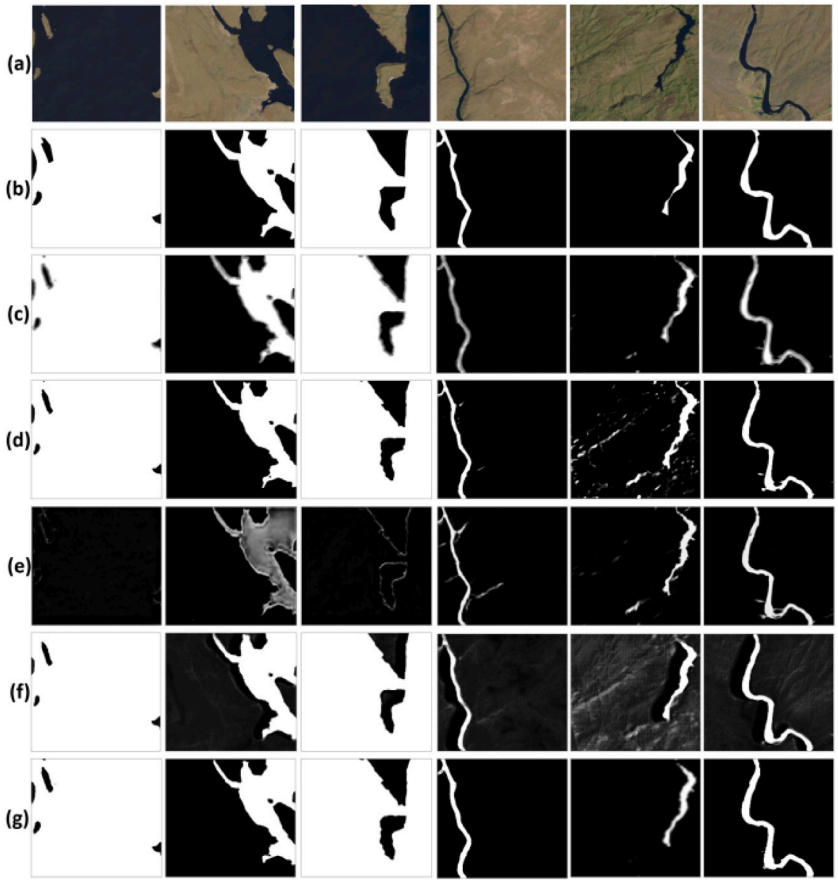

Fig. 9. Results on several samples. Each row presents: (a) input image, (b) ground truth, (c) Unet [24], (d) Deepunet [22], (e) Deepcrack [59], (f) DeepWMV2 [23], and (g) proposed W-Net.

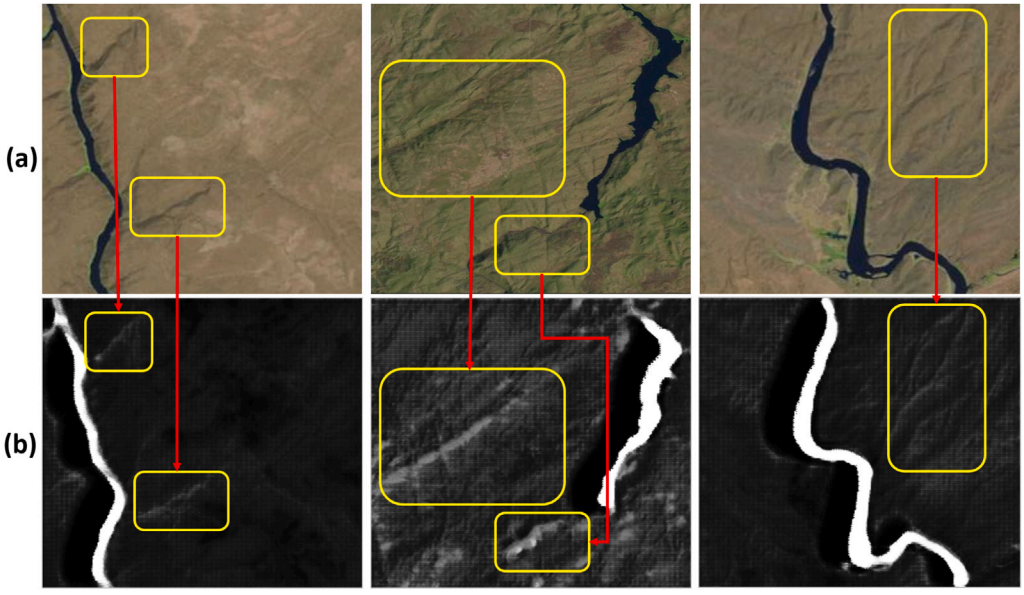

Fig. 10. Results on three samples with river-like structures. In each row we present: (a) input images and (b) predicted images obtained by DeepWMV2. Yellow bound boxes show elevated mountain ridge in both input images and predicted images.

sample images i.e. input image 1 and input image 2 randomly selected from dataset. First two columns represent predicted images, third and fourth columns denote output of refinement one (filtered output), fifth and sixth columns show edge output after applying refinement two.

In Fig. 12, each row (a)–(e) corresponds to Unet, Deepunet, Deepcrack, DeepWMV2 and proposed W-Net, respectively. It is evident that after applying refinement one blurriness is considerably decreased in case of Unet. For Deepunet and W-Net, predicted images are visually

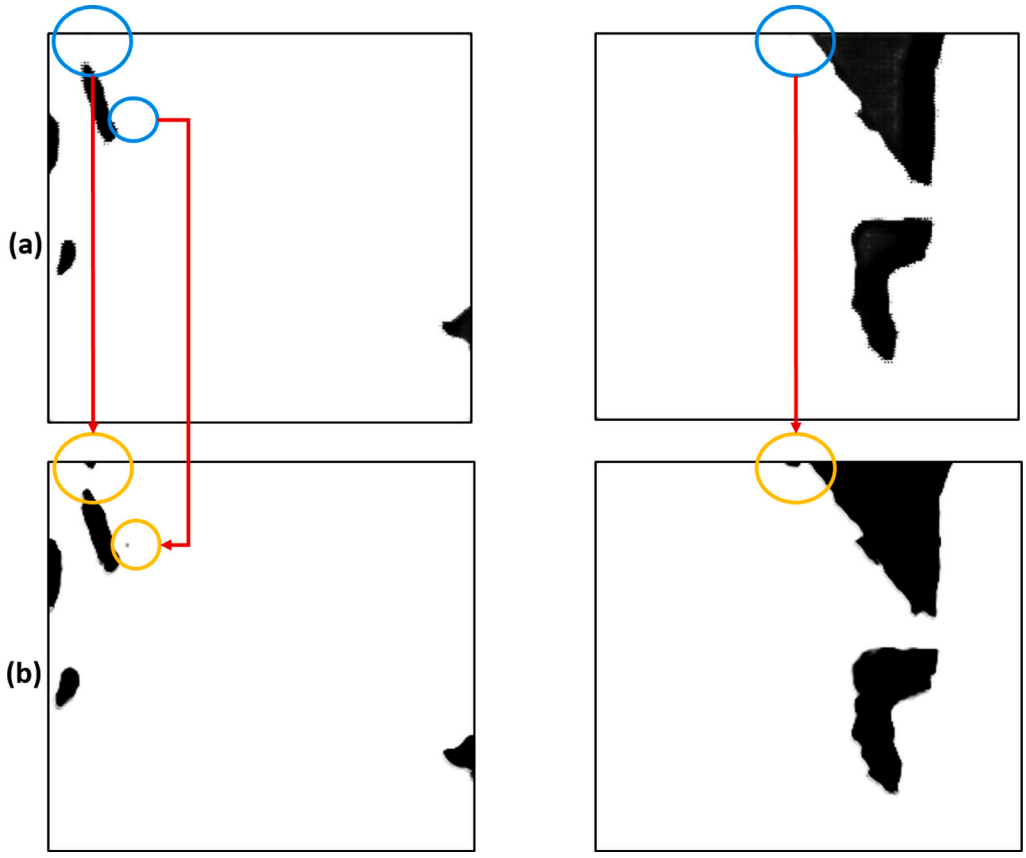

Fig. 11. Results on two samples with non-river structure. Blue circle shows misclassified land pixels as water pixels while orange circle shows correctly classified non-water pixels. In each row we present: (a) predicted images acquired from DeepWMV2 and (b) predicted images obtained by proposed W-Net.

Table 1
Comparison of methods with number of layers and trainable parameters.

Networks	Layers	Trainable parameters	Running time
Unet [24]	23	31.03	236
Deepunet [22]	32	124.44	469
Deepcrack [59]	23	14.72	176
DeepWMV2 [23]	20	37.21	259
W-Net (proposed)	23	**10.33**	**103**

Note: Trainable parameters are in million while running time in milliseconds. DeepWMV2 denotes DeepWaterMapV2. Bold black font denotes best values.

Table 2
Comparison of methods based on metrics for predicted images.

Methods	Metrics					
	G	C	I/U	P	R	F
Unet	0.9465	0.9693	0.8223	0.7061	0.6824	0.8277
Deepunet	0.9717	0.9838	0.8937	0.8199	0.8404	0.9010
Deepcrack	0.5522	0.7431	0.3546	0.2231	0.2113	0.3648
DeepWMV2	0.7592	0.8618	0.5359	0.3481	0.3835	0.5164
W-Net	**0.9847**	**0.9912**	**0.9381**	**0.8881**	**0.8991**	**0.9469**

Note: Best performances are highlighted by bold black font. W-Net is the proposed network.

appealing and boundary pixels are sharp. For Deepcrack and DeepWMV2, refinement one achieved higher rate of identifying non-water pixels as water pixels which resulted in increased number of false negatives. As seen in Fig. 12(c), Deepcrack works well with images having river-like structure (predicted image 2 and filtered output 2) as compared to predicted image 1 and filtering output 1. Here, number of false negatives are less than [23]. Refinement two is applied on filtered output to get boundary information as shown in Fig. 12 (fifth and sixth columns). Fig. 13 shows edge output and we can visually identify continuous boundary pixels except for Unet (shown by red circle). Yellow bounded boxes show misclassified non-water pixels as water pixels in case of Unet and Deepunet which are very difficult to identify from predicted images. Fig. 13(c) shows no misclassification for proposed W-Net. Above reasoning shows that both the refinement modules help in performing in-depth analysis for misclassified pixels and identifying boundary pixels.

Table 1 presents comparison on the basis of number of convolutional layers used and total trainable parameters with running time. DeepWMV2 uses less number of layers (20) as compared to other networks but the total number of trainable parameters generated are still huge (37.21 million). This is because of its input image (Landsat −8) which consist over 300 million pixels per scene. Table 1 indicates that W-Net is not only light weighted but also fast architecture. Table 2 shows comparison for performance metrics of different networks on our testing dataset. Here, we calculated the G, C, I/U, P, R and F for predicted images. Table 2 clearly specifies that the pixel accuracy for predicted images of proposed W-Net exceeds by 0.4325 when compared with Deepcrack while Deepunet shows good performance with minimum difference of 0.0074. The predictive accuracy over all classes is observed highest for W-Net i.e. 0.9912 when compared to Unet, Deepcrack and DeepWMV2 with difference of 0.0382, 0.2481 and 0.1294, respectively. Deepcrack performs worse as it is designed for crack detection and fails in segmenting non-river structures while capturing only river-like structures which are similar to those of crack structures.

Table 3 presents the quantitative analysis after applying refinement modules one. For Unet and Deepunet slight improvement is observed but after refinement one, misclassified non-water pixels are identified as water pixels. Deepcrack and DeepWMV2 deteriorates their performance as both networks work better either with non-river structures

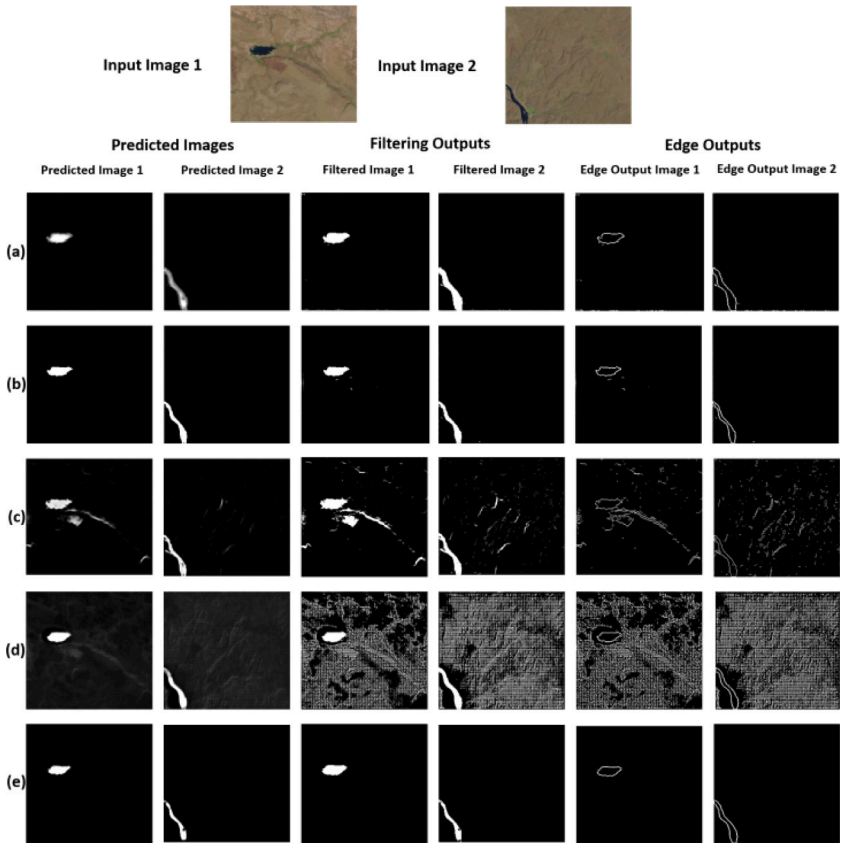

Fig. 12. Results on two samples, input image 1 and input image 2. First and second columns show predicted images, third and fourth columns show filtering outputs (refinement one) and last two columns show edge outputs (refinement two). Each row present : (a) Unet, (b) Deepunet, (c) Deepcrack, (d) DeepWMV2, and (e) proposed W-Net.

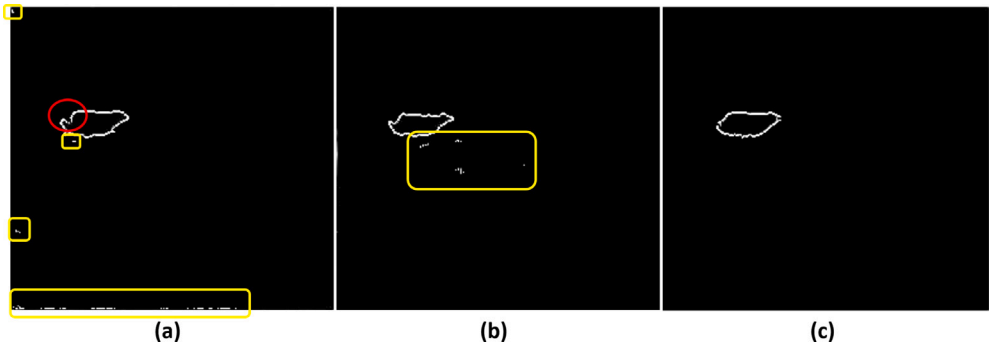

Fig. 13. Results of refinement module two. Yellow bounded boxes show misclassified land pixels as water pixels while red circle shows discontinuous boundary pixels. (a) edge output generated by Unet, (b) edge output generated by Deepunet , and (c) edge output obtained by proposed W-Net.

or river-like structures. While comparing predicted images after refinement one, noticeable changes are observed. When predicted images are compared with filtering output for DeepWMV2, significant difference of 0.3714, 0.4531 and 0.2057 is observed for metrics G, C and I/U, respectively. Quantitative values (P, R and F) provided by Table 3 confirm that the higher non-water pixels were predicted to be water pixels which endorse our qualitative analysis for DeepWMV2. The predicted image is compared with filtering output for proposed W-Net, prominent improvement is reflected in statistical information ($G = 0.9899$, $C = 0.9964$, $I/U = 0.9434$, $P = 0.8901$, $R = 0.9018$, $F = 0.9509$). The quantitative investigation indicated in Table 3 compliments qualitative analysis made through Fig. 12. Statistics presented by Tables 2 and 3 clearly recommend that except for Deepcrack and DeepWMV2 all other methods show significant improvement and take full advantage of refinement modules. Hence, with qualitative and quantitative analysis, it is proved that our proposed W-Net is superior to other methods.

Table 3
Comparison of methods based on metrics for images after refinement modules.

Methods	Metrics					
	G	C	I/U	P	R	F
Unet	0.9899	0.9731	0.8519	0.7063	0.6998	0.8298
Deepunet	0.9801	0.9899	0.9003	0.8214	0.8497	0.9099
Deepcrack	0.7634	0.8699	0.5398	0.3491	0.3799	0.5113
DeepWMV2	0.3878	0.4087	0.3302	0.1865	0.1926	0.2388
W-Net	**0.9971**	**0.9964**	**0.9434**	**0.8901**	**0.9018**	**0.9509**

Note: W-Net is the proposed network.

4.5. Cross-dataset evaluation

We have validated proposed architecture on crack detection dataset used in [59]. Fig. 14 shows sample images from water body dataset

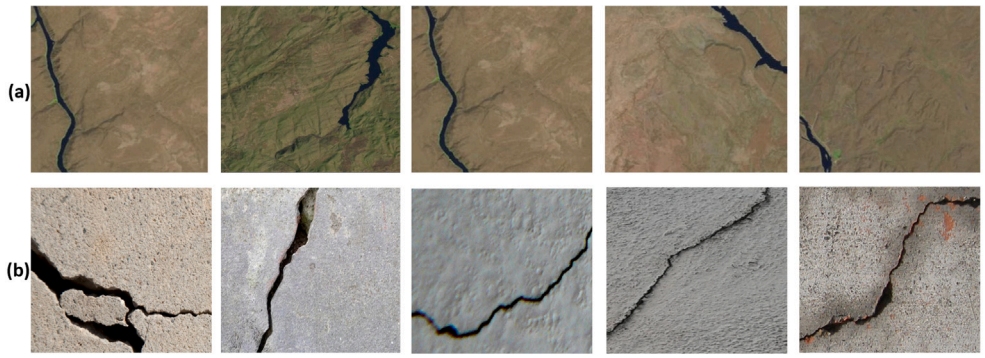

Fig. 14. Sample input images for cross-dataset evaluation from: (a) water body dataset and (b) crack detection dataset.

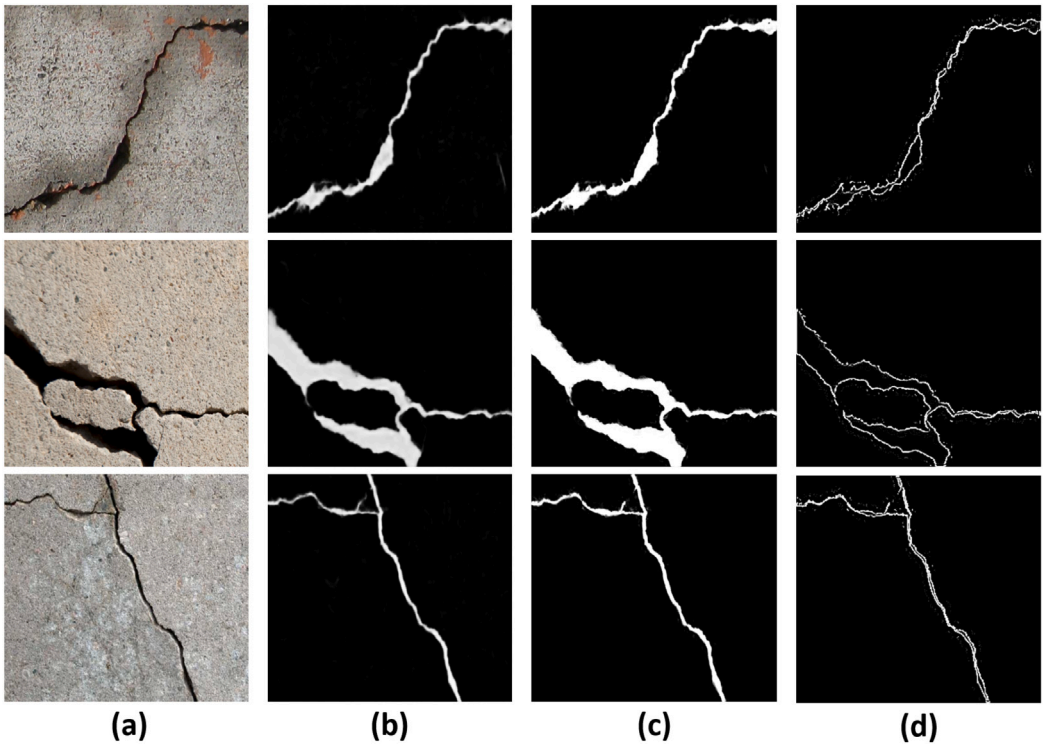

Fig. 15. Result of Deepcrack on sample crack detection images. Each column present: (a) random input images chosen from crack detection dataset, (b) predicted images of Deepcrack, (c) guided filtered output of Deepcrack, and (d) edge output from refinement module two generated from guided filtered output.

(Fig. 14(a)) and crack detection dataset (Fig. 14(b)). Fig. 15 displays output of deepcrack when applied on crack dataset. Fig. 15(b), (c) and (d), show predicted images, guided filter outputs and edge outputs, respectively for Deepcrack. As we can see that narrow and wider river-like structure are very much identical to that of thin and broader crack images. Our proposed W-Net architecture is not fine-tuned with crack detection dataset but trained with river-like structure images which are similar to crack images. Taking advantage of such images, W-Net is able to predict cracks which resemble to predicted images by Liu et al. [59] as shown in Fig. 15. Fig. 16(d) and Fig. 16(e) display filtered output of refinement module one and edge output of refinement module two, respectively. With very few water pixels predicted as non-water pixels, W-Net achieves competitive performance with [59].

4.6. Ablation study

We have proposed W-Net which includes inception units (IDU and IUU). Here, we compare three variants of W-Net architecture with U-Net. W-Net* with inception blocks (IDU) at encoder path without inception blocks (IUU) at decoding path, W-Net** with inception blocks (IUU) at decoder path without inception blocks (IDU) at encoder path and W-Net with inception blocks (IDU) at encoder path as well as inception blocks (IUU) at decoding path. Table 4 gives details about number of layers and trainable parameters. It is obvious from Table 4 that number of convolution layers are same for both variants of W-Net and U-Net. However, the number of trainable parameters in W-Net is significantly less as compared to W-Net*, W-Net** and U-Net. This is achievable due to use of asymmetric convolutions [52]. The proposed W-Net architecture with inception layers at both encoding and decoding path is light weighted than without inception layers.

Fig. 17 shows the comparison between variants of W-Net and U-Net. The first row presents two input images (input image 1 and 2) randomly chosen from dataset with their ground truths. Fig. 17(a) (second row) speaks for U-Net which corresponds to output images produced after refinement one and two. Fig. 17(b) (third row) and Fig. 17(c) (fourth row) shows the output images produced after refinement module (refinement one and two) for W-Net* and W-Net**, respectively. Fig. 17(d) (last row) represents W-Net showing output

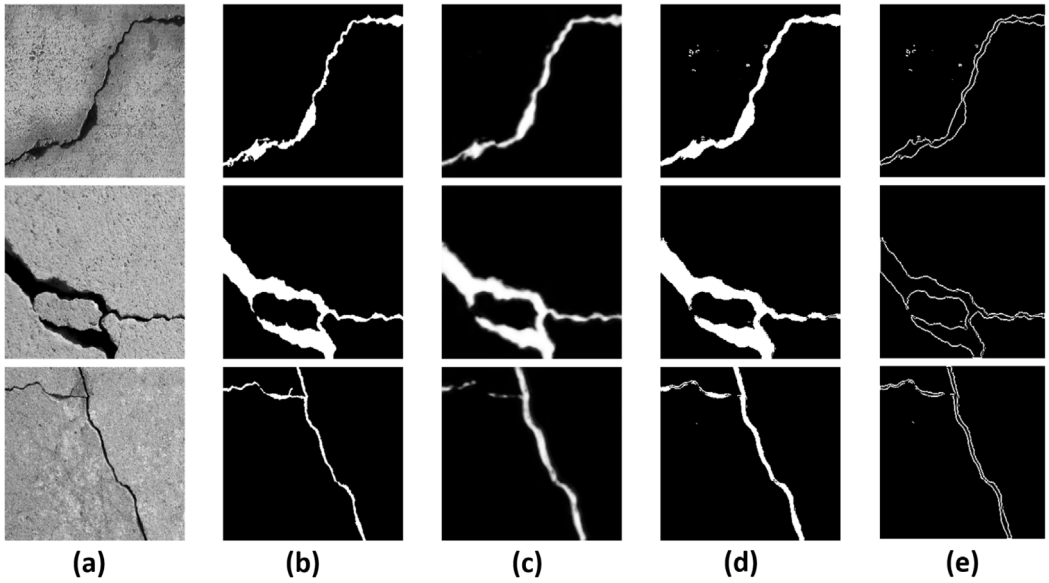

Fig. 16. Result of proposed W-Net architecture on cross-dataset evaluation on sample crack detection images. Each column present: (a) random input images chosen from crack detection dataset, (b) ground truths, (c) predicted images of W-Net, (d) filtered outputs from refinement module one, and (e) edge outputs from refinement module two.

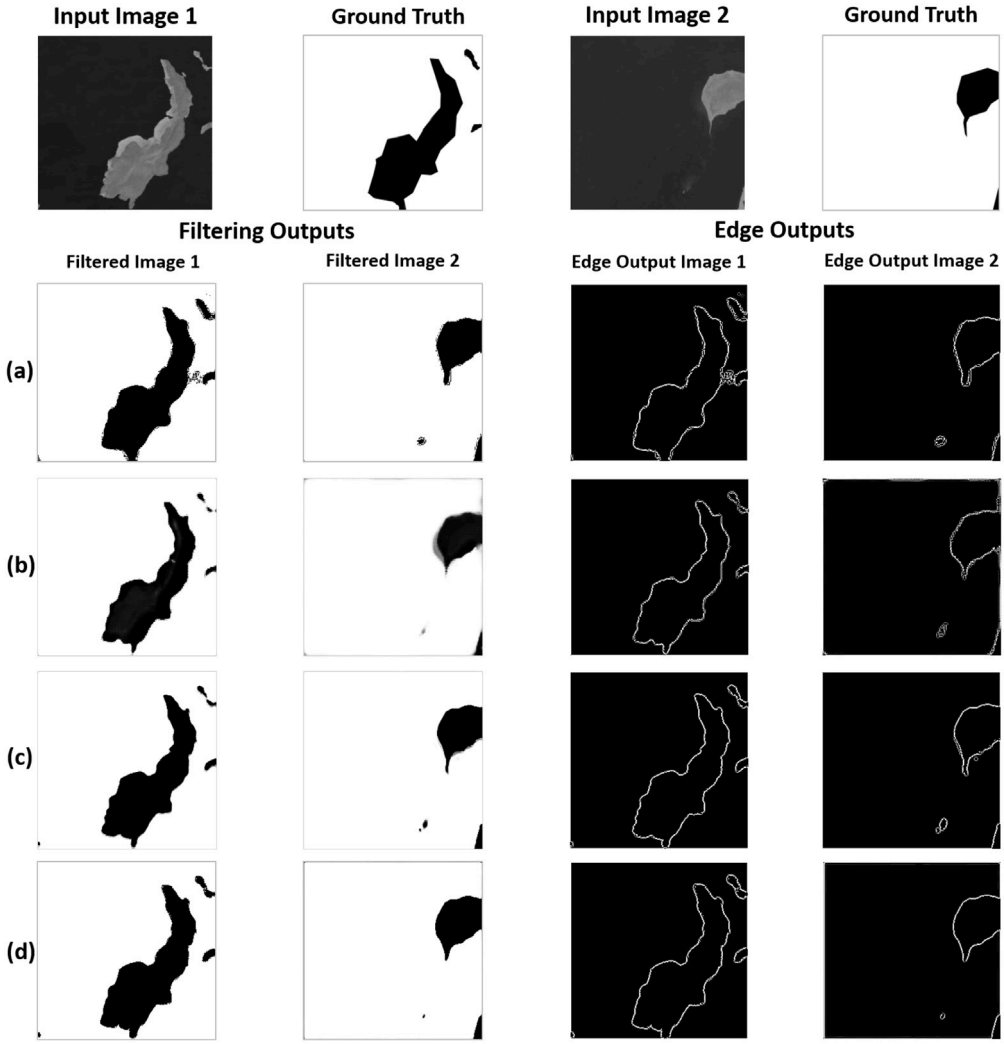

Fig. 17. Results on two samples, input image 1 and input image 2. First and second columns show filtering outputs (refinement one) and last two columns show edge outputs (refinement two). Each row present: (a) result images using U-Net, (b) result images using W-Net*, (c) result images using W-Net** and (d) result images using W-Net.

Table 4
Comparison of W-Net and U-Net for total number of trainable parameters.

Networks	Layers	Trainable parameters	Running time
U-Net	23	31.03	236
W-Net*	23	15.43	165
W-Net**	23	15.43	165
W-Net	23	**10.33**	**103**

Note: Trainable parameters are in million while running time in milliseconds. W-Net* and W-Net** are variants of W-Net while W-Net is the proposed network.

Table 5
Comparison of W-Net and U-Net based on performance metrics.

Methods	Metrics					
	G	*C*	*I/U*	*P*	*R*	*F*
U-Net	0.8918	0.9178	0.8715	0.8314	0.8457	0.8912
W-Net*	0.9411	0.9561	0.8943	0.8501	0.8666	0.9102
W-Net**	0.9623	0.9773	0.9149	0.8676	0.8789	0.9332
W-Net	**0.9899**	**0.9964**	**0.9434**	**0.8901**	**0.9018**	**0.9509**

Note: W-Net is the proposed network.

images after applying refinement one and two on both input images. Fig. 17 indicates that W-Net is close to expected ground truth and does not suffer from pixel misclassification. Pixel misclassification is observed in U-Net, W-Net* and W-Net** (Fig. 17(a), (b) and (c)). Boundary pixels are clearly identified in W-Net whereas U-Net, W-Net* and W-Net** do not determine boundary pixels efficiently. Table 5 shows that W-Net increases the pixel prediction and predictive accuracy by 0.0981 and 0.0786, respectively. W-Net has also amortized number of false negatives comparative to U-Net. Tables 4 and 5 present the comparison between W-Net with and without inception blocks. By introducing inception layers, W-Net not only reduces the computational complexity but also improves the overall prediction results. The qualitative and quantitative analysis demonstrate that W-Net is competent and computationally cheaper than U-Net and its two variants.

5. Conclusion

We present an end-to-end multi-feature based architecture, named W-Net, for extracting water pixels through segmentation. W-Net took advantage of encoding and decoding path to hold context information and localization, respectively. Large number of feature channels in up-sampling layers allow the network to transmit context information to high resolution layers which maintain consistency in segmentation. In addition, we incorporated inception layers after each encoder and decoder units which resulted in reducing computations within the network. Therefore, total numbers of trainable parameters are reduced dramatically in W-Net. W-Net enforced two refinement modules. Refinement one has improved quality of predicted images by reducing blurring effect while refinement two provides edge information (locating continuity and discontinuity of boundary pixels). We also evaluated W-Net on crack detection dataset where it showed competitive performance by achieving results that resembles to annotated maps. The qualitative and quantitative comparison reveals the superiority of W-Net over the state-of-the-art methods.

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.

References

[1] P. Smith, M.R. Ashmore, H.I. Black, P.J. Burgess, C.D. Evans, T.A. Quine, A.M. Thomson, K. Hicks, H.G. Orr, The role of ecosystems and their management in regulating climate, and soil, water and air quality, J. Appl. Ecol. 50 (4) (2013) 812–829.

[2] S.K. McFeeters, The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features, Int. J. Remote Sens. 17 (7) (1996) 1425–1432.

[3] B.-C. Gao, NDWI-A normalized difference water index for remote sensing of vegetation liquid water from space, Remote Sens. Environ. 58 (3) (1996) 257–266.

[4] H. Xu, Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery, Int. J. Remote Sens. 27 (14) (2006) 3025–3033.

[5] Q. Guo, R. Pu, J. Li, J. Cheng, A weighted normalized difference water index for water extraction using landsat imagery, Int. J. Remote Sens. 38 (19) (2017) 5430–5445.

[6] H. Song, Q. Liu, G. Wang, R. Hang, B. Huang, Spatiotemporal satellite image fusion using deep convolutional neural networks, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (3) (2018) 821–829.

[7] E. Maggiori, G. Charpiat, Y. Tarabalka, P. Alliez, Recurrent neural networks to correct satellite image classification maps, IEEE Trans. Geosci. Remote Sens. 55 (9) (2017) 4962–4971.

[8] P. Zhang, X. Niu, Y. Dou, F. Xia, Airport detection on optical satellite images using deep convolutional neural networks, IEEE Geosci. Remote Sens. Lett. 14 (8) (2017) 1183–1187.

[9] M. Vakalopoulou, S. Christodoulidis, M. Sahasrabudhe, S. Mougiakakou, N. Paragios, Image registration of satellite imagery with deep convolutional neural networks, in: IGARSS 2019–2019 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2019, pp. 4939–4942.

[10] M. Segal-Rozenhaimer, A. Li, K. Das, V. Chirayath, Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (CNN), Remote Sens. Environ. 237 (2020) 111446.

[11] S. Ghassemi, A. Fiandrotti, G. Francini, E. Magli, Learning and adapting robust features for satellite image segmentation on heterogeneous data sets, IEEE Trans. Geosci. Remote Sens. 57 (9) (2019) 6517–6529.

[12] Z. Wu, S. Li, C. Chen, A. Hao, H. Qin, A deeper look at image salient object detection: Bi-stream network with a small training dataset, IEEE Trans. Multimed. (2020).

[13] X. Wang, S. Li, C. Chen, Y. Fang, A. Hao, H. Qin, Data-level recombination and lightweight fusion scheme for RGB-D salient object detection, IEEE Trans. Image Process. 30 (2020) 458–471.

[14] G. Ma, S. Li, C. Chen, A. Hao, H. Qin, Stage-wise salient object detection in 360° omnidirectional image via object-level semantical saliency ranking, IEEE Trans. Vis. Comput. Graphics 26 (12) (2020) 3535–3545.

[15] C. Chen, J. Wei, C. Peng, W. Zhang, H. Qin, Improved saliency detection in RGB-D images using two-phase depth estimation and selective deep fusion, IEEE Trans. Image Process. 29 (2020) 4296–4307.

[16] G. Ma, C. Chen, S. Li, C. Peng, A. Hao, H. Qin, Salient object detection via multiple instance joint re-learning, IEEE Trans. Multimed. 22 (2) (2019) 324–336.

[17] Y. Li, S. Li, C. Chen, A. Hao, H. Qin, A plug-and-play scheme to adapt image saliency deep model for video data, IEEE Trans. Circuits Syst. Video Technol. (2020).

[18] C. Chen, S. Li, Y. Wang, H. Qin, A. Hao, Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion, IEEE Trans. Image Process. 26 (7) (2017) 3156–3170.

[19] Z. Miao, K. Fu, H. Sun, X. Sun, M. Yan, Automatic water-body segmentation from high-resolution satellite images via deep networks, IEEE Geosci. Remote Sens. Lett. 15 (4) (2018) 602–606.

[20] L. Li, Z. Yan, Q. Shen, G. Cheng, L. Gao, B. Zhang, Water body extraction from very high spatial resolution remote sensing data based on fully convolutional networks, Remote Sens. 11 (10) (2019) 1162.

[21] W. Feng, H. Sui, W. Huang, C. Xu, K. An, Water body extraction from very high-resolution remote sensing imagery using deep U-Net and a superpixel-based conditional random field model, IEEE Geosci. Remote Sens. Lett. 16 (4) (2018) 618–622.

[22] R. Li, W. Liu, L. Yang, S. Sun, W. Hu, F. Zhang, W. Li, Deepunet: A deep fully convolutional network for pixel-level sea-land segmentation, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (11) (2018) 3954–3962.

[23] L.F. Isikdogan, A. Bovik, P. Passalacqua, Seeing through the clouds with deepwatermap, IEEE Geosci. Remote Sens. Lett. (2019).

[24] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[25] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.

[26] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[27] G.L. Feyisa, H. Meilby, R. Fensholt, S.R. Proud, Automated Water Extraction Index: A new technique for surface water mapping using Landsat imagery, Remote Sens. Environ. 140 (2014) 23–35.

[28] A. Essa, P. Sidike, V. Asari, Volumetric directional pattern for spatial feature extraction in hyperspectral imagery, IEEE Geosci. Remote Sens. Lett. 14 (7) (2017) 1056–1060.
[29] R. Qin, A mean shift vector-based shape feature for classification of high spatial resolution remotely sensed imagery, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 8 (5) (2014) 1974–1985.
[30] X. Huang, C. Xie, X. Fang, L. Zhang, Combining pixel-and object-based machine learning for identification of water-body types from urban high-resolution remote-sensing imagery, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 8 (5) (2015) 2097–2110.
[31] F. Yao, C. Wang, D. Dong, J. Luo, Z. Shen, K. Yang, High-resolution mapping of urban surface water using ZY-3 multi-spectral imagery, Remote Sens. 7 (9) (2015) 12336–12355.
[32] W. Wu, Q. Li, Y. Zhang, X. Du, H. Wang, Two-step urban water index (TSUWI): A new technique for high-resolution mapping of urban surface water, Remote Sens. 10 (11) (2018) 1704.
[33] X.X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, F. Fraundorfer, Deep learning in remote sensing: A comprehensive review and list of resources, IEEE Geosci. Remote Sens. Mag. 5 (4) (2017) 8–36.
[34] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P.H. Torr, Conditional random fields as recurrent neural networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1529–1537.
[35] M. Everingham, S.A. Eslami, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, Int. J. Comput. Vis. 111 (1) (2015) 98–136.
[36] H. Lin, Z. Shi, Z. Zou, Fully convolutional network with task partitioning for inshore ship detection in optical remote sensing images, IEEE Geosci. Remote Sens. Lett. 14 (10) (2017) 1665–1669.
[37] H. Lin, Z. Shi, Z. Zau, Maritime semantic labeling of optical remote sensing images with multi-scale fully convolutional network, Remote Sens. 9 (5) (2017) 480.
[38] F. Isikdogan, A.C. Bovik, P. Passalacqua, Surface water mapping by deep learning, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 10 (11) (2017) 4909–4918.
[39] D. Cheng, G. Meng, G. Cheng, C. Pan, SeNet: Structured edge network for sea-land segmentation, IEEE Geosci. Remote Sens. Lett. 14 (2) (2016) 247–251.
[40] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1520–1528.
[41] A. Rakhlin, A. Davydow, S.I. Nikolenko, Land cover classification from satellite imagery with U-Net and lovasz-softmax loss, in: CVPR Workshops, 2018, pp. 262–266.
[42] J.H. Kim, H. Lee, S.J. Hong, S. Kim, J. Park, J.Y. Hwang, J.P. Choi, Objects segmentation from high-resolution aerial images using U-Net with pyramid pooling layers, IEEE Geosci. Remote Sens. Lett. 16 (1) (2018) 115–119.
[43] V. Iglovikov, A. Shvets, Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation, 2018, arXiv preprint arXiv:1801.05746.
[44] V. Iglovikov, S.S. Seferbekov, A. Buslaev, A. Shvets, TernausNetV2: Fully convolutional network for instance segmentation, in: CVPR Workshops, Vol. 233, 2018, p. 237.
[45] V. Khryashchev, R. Larionov, A. Ostrovskaya, A. Semenov, Modification of U-Net neural network in the task of multichannel satellite images segmentation, in: 2019 IEEE East-West Design & Test Symposium, EWDTS, IEEE, 2019, pp. 1–4.
[46] Z. Zhang, Q. Liu, Y. Wang, Road extraction by deep residual u-net, IEEE Geosci. Remote Sens. Lett. 15 (5) (2018) 749–753.
[47] A. Buslaev, S.S. Seferbekov, V. Iglovikov, A. Shvets, Fully convolutional network for automatic road extraction from satellite imagery, in: CVPR Workshops, 2018, pp. 207–210.
[48] X. Yang, X. Li, Y. Ye, R.Y. Lau, X. Zhang, X. Huang, Road detection and centerline extraction via deep recurrent convolutional neural network U-net, IEEE Trans. Geosci. Remote Sens. 57 (9) (2019) 7209–7220.
[49] J. Gonzalez, D. Bhowmick, C. Beltran, K. Sankaran, Y. Bengio, Applying knowledge transfer for water body segmentation in peru, 2019, arXiv preprint arXiv:1912.00957.
[50] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.
[51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[52] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[53] J. Yang, B. Price, S. Cohen, H. Lee, M.-H. Yang, Object contour detection with a fully convolutional encoder-decoder network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 193–202.
[54] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning, ICML-10, 2010, pp. 807–814.
[55] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015, arXiv preprint arXiv:1502.03167.
[56] D. Pouliot, R. Latifovic, J. Pasher, J. Duffe, Landsat super-resolution enhancement using convolution neural networks and Sentinel-2 for training, Remote Sens. 10 (3) (2018) 394.
[57] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[58] Y. Liu, J. Yao, X. Lu, M. Xia, X. Wang, Y. Liu, Roadnet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images, IEEE Trans. Geosci. Remote Sens. 57 (4) (2018) 2043–2056.
[59] Y. Liu, J. Yao, X. Lu, R. Xie, L. Li, DeepCrack: A deep hierarchical feature learning architecture for crack segmentation, Neurocomputing 338 (2019) 139–153.
[60] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. (6) (1986) 679–698.
[61] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} Symposium on Operating Systems Design and Implementation, {OSDI} 16, 2016, pp. 265–283.
[62] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[63] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[64] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
[65] A. Rakhlin, A. Davydow, S.I. Nikolenko, Land cover classification from satellite imagery with U-Net and lovasz-softmax loss, in: CVPR Workshops, 2018, pp. 262–266.