



# Optimized Deep Network for Precise Digital Terrain Model Extraction from Light Detection and Ranging Data

A'kif Al-Fugara<sup>a</sup>, Ali Nouh Mabdeh<sup>b</sup>, Abdel Rahman Al-Shabeeb<sup>b</sup>, Rami Al shawabkeh<sup>c</sup>,  
Laith Abualigah<sup>d,e,f,g,h</sup>

<sup>a</sup>Dept. of Surveying Engineering, Faculty of Engineering, Al al-Bayt University, Mafraq 25113, Jordan

<sup>b</sup>Dept. of Geographic Information Systems and Remote Sensing, Faculty of Earth and Environmental Sciences, Al al-Bayt University, Mafraq 25113, Jordan

<sup>c</sup>Dept. of Architecture Engineering, Faculty of Engineering, Al al-Bayt University, Mafraq 25113, Jordan

<sup>d</sup>Computer Science Department, Al al-Bayt University, Mafraq 25113, Jordan

<sup>e</sup>MEU Research Unit, Middle East University, Amman 11831, Jordan

<sup>f</sup>Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

<sup>g</sup>Jadara Research Center, Jadara University, Irbid 21110, Jordan

<sup>h</sup>Centre for Research Impact & Outcome, Chitkara University, Punjab

## ARTICLE HISTORY

Received January 15, 2024  
Accepted April 16, 2024  
Published Online 6 July 2024

## KEYWORDS

Digital terrain model  
Digital surface model  
LiDAR  
Hierarchical deep network  
Particle swarm optimization

## ABSTRACT

Classifying and filtering non-ground objects from the point cloud data are among the major challenges to the development of a digital terrain model (DTM). This paper proposes a hierarchical deep network to filter and classify non-ground objects from the point cloud data. The proposed network is mainly based on a deep encoder-decoder network with effective convolutional connections for extracting and fusing the features of shallow and deep layers to detect the objects better. In the proposed encoder-decoder network, a feature extraction block was designed to extract various features at different levels. The network also adopts a global-local feature fusion strategy. The proposed hierarchical deep network is based on the extraction and gradual fusion of features on different scales to extract objects of various dimensions and densities in urban areas with different topographical conditions. Evaluation of the proposed deep network in US and ISPRS datasets indicated its high accuracy of object detection in complicated and dense areas. In the US dataset, our model reduced total error by 0.055 and increased kappa by 20.18% on average compared to the second-best method. According to the results of a performance analysis through the ISPRS data, the proposed deep network outperformed the other methods and reduced the total error by 0.130 compared to the existing methods.

## 1. Introduction

Photogrammetry and remote sensing are powerful tools to gather information about the Earth from a distance using sensors on satellites, airplanes, or drones to collect data in the form of light, radar, or other electromagnetic waves (Zhu et al., 2018; Hong et al., 2023; Zhao and Sun, 2023). Given the emergence of different technologies such as LiDAR (light detection and ranging) or various techniques such as dense matching of aerial imagery, it is now possible to quickly collect the accurate 3D point cloud from

the Earth's surface. Represented as the point cloud, these data include 3D information about the Earth's surface and all of its objects. The point cloud is called the Digital Surface Model (DSM). Digital Terrain Model (DTM) generation is crucial for various applications such as urban planning, flood modeling, line-of-sight analysis, and landscape modeling (Pingel et al., 2013; Gevaert et al., 2018). It provides valuable information about the topography of an area, aiding in the understanding of surface features, elevation changes, and slope characteristics. This data is essential for making informed decisions in fields like

**CORRESPONDENCE** Laith Abualigah ✉ [aligah.2020@gmail.com](mailto:aligah.2020@gmail.com) ☒ Computer Science Department, Al al-Bayt University, Mafraq 25113, Jordan; Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan; MEU Research Unit, Middle East University, Amman 11831, Jordan; Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

© 2024 Korean Society of Civil Engineers

engineering, environmental assessment, and infrastructure development. Converting a DSM into a DTM is among the important applications of photogrammetry and remote sensing. It necessitates filtering or detecting and eliminating the non-ground objects. The difference between DSM and DTM resulted in the development of the normalized Digital Surface Model (nDSM), which includes the heights of non-ground objects. In fact, an nDSM plays a key role in detecting and classifying objects and developing their 3D models (Gevaert et al., 2018; Bigdeli et al., 2018b). The process of extracting a DTM from a DSM usually consists of two phases, in the first of which non-ground points are detected and eliminated. In the next phase, the DTM is developed by interpolating and retrieving the height information about the eliminated areas. Filtering methods are faced with various challenges and problems that can be classified as two categories (Gevaert et al., 2018), the first of which is due to the inherent characteristic of adopting or collecting the point cloud. For instance, shades can cause problems in the matching procedure or create noise in the outputs when the point cloud is obtained from the dense matching of aerial imagery. Moreover, the insufficiency of spectral information or calibration of cameras can sometimes cause noise or false data in the point cloud. In addition, the point cloud developed through LiDAR might have false data and noise due to dispersal or multiple reflections from different surfaces. The features of the study areas pose the second category of challenges to filtering algorithms. For instance, it is difficult to detect non-ground objects in sloped areas. Moreover, high-dimensional objects can adversely affect the detection accuracy (Pingel et al., 2013; Chen et al., 2016a; Gevaert et al., 2018). The algorithms proposed for point cloud filtering can be classified as four categories: methods based on progressive densification, morphology-based operators, surface-based methods, and segmentation-based methods.

In the methods based on progressive densification, a few points are considered the initial points which must belong to the ground. They are then densified and multiplied through a progressive process. Axelsson (2000) used a grid to select the points with the lowest height and develop a Triangulated Irregular Network (TIN). In fact, an initial network was employed with the points of the lowest height to eliminate the edge-cutting phenomenon. The developed TIN would be densified progressively by adding the points with shorter distances from three neighboring node points in the network than a threshold (Axelsson, 2000). Sohn and Dowman (2002) proposed an algorithm with triangulation downward and divide-and-conquer upward trends as well as the progressive DTM enhancement. For this purpose, an initial coarse TIN was first developed with a specific number of points. In a triangulation downward trend, the points with lower heights than the threshold were detected to update the DTM level. The locational relationship between the rest of the nodes and the TIN was then analyzed in the divide-and-conquer trend (Sohn and Dowman, 2002). In fact, the model was mainly based on the assumption to find the candidate points that could be added to the model and divide the surface locally to further flat ground

surfaces. The distance criterion is employed to select a closer point as the candidate point when the number of candidate points is larger than a specific number. The process of detecting and adding points to the surface would continue until no new points were found (Sohn and Downman, 2002). Zhang and Lin (2013) employed the progressive TIN enhancement and segmentation based on the softness criterion for DTM extraction. First, the initial points were selected, and a region growing trend was then used with respect to the initial points to filter the non-ground points and TIN densification. Guan et al. (2014) proposed a filtering method based on the Cross-Section Plane (CSP) by first placing the point cloud inside a 3D grid and then using a multi-directional CSP, which can extract the features of a complicated 3D object as 2D features, to extract the features of every grid cell. This algorithm focused on the forest areas. Since the laser pulses can penetrate vegetation, the points with only one return were considered the initial ground points. The other potential ground points were extracted by considering certain rules based on the number of neighboring ground points as well as the height difference and signal intensity of each point in comparison with the neighboring points. All of the potential ground points were then filtered inside each cell through segmentation and point extraction with the lowest height. Ultimately, the final ground points were obtained from each direction of the CSP by extracting and crossing the ground point set (Guan et al., 2014). Chen et al. (2016a, 2016b) emphasized that the TIN-based method would mainly face problems in the steep slope mountain regions, for which they proposed three strategies. First, they used the triangular relationships to detect the triangles placed on ridges and discontinuity areas. The points with the lowest local heights and the ridge points were then detected. Finally, a relationship was employed to control the iteration number and achieve the best accuracy.

The morphology-based operators are potentially capable of eliminating non-ground objects and developing DTM. Vosselman (2000) proposed a filtering method based on mathematical morphology to calculate the surface details by analyzing the height differences of neighboring points (Vosselman, 2000). Sithole (2001) proposed a local operator, the parameters of which can change as a function of the region slope, to minimize the negative effects of changes in the surface height. Roggero (2001) employed a morphology-based operator whose parameters were tuned in proportion to the region surface (Roggero, 2001). Kobler et al. (2007) considered a slope-based operator for the DTM extraction and analyze the upward height difference and the downward height difference separately. Shao and Chen (2008) proposed a climbing and sliding method and tried to develop the local search technique by considering the general features. For this purpose, geometrical features such as erosion, dilation, and performance improvement slope of the steep slope regions were used (Shao and Chen, 2008). Lu et al. (2009) employed a hybrid conditional random field method of the DTM extraction, in which a supervised technique was proposed to classify ground and non-ground points on continuous and discrete surfaces. They used the height values of the detected ground

points, which can be extracted from LiDAR data, to create the ground. The height values of the ground surface around the detected non-ground points were determined through the Gaussian random field (Lu et al., 2009). Sampath and Shan (2004) detected the non-ground points by applying 1D and 2D filters (Shan and Aparajithan, 2005), a technique which was also developed as a multi-directional method (Meng et al., 2009). Wang and Tseng (2014) divided the conventional 1D coin operator into two separate operators, the results of which were fused to extract the non-ground objects. These operators can be applied horizontally and vertically. They are suitable for detecting non-ground objects in urban areas due to their capability of extracting drastic surface changes and sudden stair changes (Wang and Tseng, 2014). Hu et al. (2015) adopted two strategies for detecting and eliminating non-ground objects. The value of the slope parameter is determined with respect to the surface bulge in the first strategy that is based on slope. In fact, the value of bulge is determined by calculating the differences of the adjacent detected segments. The second strategy is to filter the points by analyzing each line in eight directions (Hu et al., 2015). Li et al. (2014) employed a top-hat filter to detect the non-ground points and then used geometrical parameters such as slope to improve the performance of that filter (Li et al., 2014). Mongus et al. (2014) used a few interconnected operators to better filter the points. For this purpose, they first utilized a grid to connect the points and then eliminate the blunder errors of structural elements. Therefore, some operators were used as the area of the widest interconnected regions, the maximum roughness of an object existing in the scene, and the differential surfaces on which non-ground objects were placed on a higher level than the neighboring points (Mongus et al., 2014). Using several thresholds on different scales would eliminate non-ground objects of different dimensions and prevent the false elimination of sloped regions misidentified as non-ground objects (Mongus and Žalik, 2012, 2013). Hu et al. (2016) employed the morphology-based operator with an iterative structure to extract the DTM, in which the dimensions of the structural element increased progressively, and various objects were eliminated in each step in proportion to the region slope (Hu and Yuan, 2016). Bigdeli et al. (2018a) proposed an iterative procedure based on the geodesic dilation operator to detect the non-ground points, in which the dimensions of the structural element increased progressively, and the detected segments were analyzed separately. The final class of non-ground objects was extracted through the detailed geometrical analysis of the detected segments (Bigdeli et al., 2018b).

It has been very common to use a local minimum to extract a DTM. For this purpose, an initial surface was created by selecting at least three points. The surface was then improved gradually by adding the ground points. In fact, it is fairly suitable to create initial surfaces with the minimum number of points for flat regions; however, this method faces problems and fails to work when the ground topography is slightly complicated. Kraus and Pfeifer (2001) proposed a method in which a primary procedure was developed through a number of control points, which are usually the points with the lowest height in each cell. The residual

values between the points and the estimated surface were then calculated, and a weight was then attributed to each point in proportion to the resultant values. The points with higher weights have greater effects on the process of updating the surface, whereas the points with lower weights have smaller effects on the process. The process of iterating and updating the surface continues until the resultant surface is stabilized or the iteration number reaches the maximum quantity (Kraus and Pfeifer, 2001). Pfeifer et al. (2001) converted the LiDAR point cloud into an image and proposed a hierarchical calculation process to improve the filtering performance and acquire better results. In fact, it is possible to improve the results gradually by comparing the initial DTM to the DTM with the above reference details (Pfeifer et al., 2001). Elmqvist (2002) employed an active model for the DTM estimation. Yielding acceptable outputs by using the dense point cloud, the model would minimize the energy function to extract a DTM in an iterative process (Elmqvist, 2002). Kobler et al. (2007) proposed an iterative interpolation procedure for the DTM extraction. In that method, classical methods were used first to primarily process and eliminate false data and non-ground points. In other words, a few TINs were selected randomly, and the generation of height data was calculated in different DTM positions. The residual non-ground points were detected by comparing the distribution of height data with the total mean of points (Kobler et al., 2007). Chen et al. (2012) proposed an iterative structure for surface retrieval to generation the DTM. In their proposed structure, the final pulse points are first matched and are then classified as different surfaces by dividing LiDAR data. In the next step, the ground points are detected, and the final DTM results are improved from a top layer to a bottom layer (Chen et al., 2012). Maguya et al. (2013) proposed an adaptive algorithm for the DTM extraction in the forest areas. In their proposed algorithm, a set of minimum point would be selected to create an initial surface. Two different linear and quadratic equations were then employed to simulate the surface. A point would be considered a ground point if it met the DTM limits. Zhang et al. (2016) proposed a Cloth Simulation Filter (CSF), in which a procedure would be created on the points, and a gravity function would be utilized to correct and form the procedure properly.

The image processing algorithms used for image classification (Li et al., 2014; Simon et al., 2023) are used extensively for the DTM extraction. In other words, the ground points are used for extraction in the DTM by categorizing the point cloud as different classes through the extracted or determined rules. Generally, the point cloud contains only height information, something which makes the point cloud classification difficult, for the limited spectral or geometrical information can be extracted from LiDAR data. In this regard, certain parameters such as the number of recursive LiDAR pulses or the difference of first and last pulses can be employed to detect non-ground objects such as vegetation (Lee and Lucas, 2007; Chen et al., 2016b). Since the geometrical behavior and signal intensity of the neighboring points are very similar in the point cloud, the object-based

classification methods can be really useful. In the object-based classification process, the point cloud is first interpolated and converted into an image. The point cloud is then divided into some segments with similar features but unspecific labels by using height values, signal intensity, and geometrical features in a segmentation method. Finally, classification is performed by applying some classification rules and considering the extracted features (Blaschke, 2010; Blaschke and Tomljenović, 2012). Antonarakis et al. (2008) used the height vegetation model, signal intensity, skewness, and kurtosis to classify LiDAR point cloud in forest areas in addition to distinguishing between ground and non-ground points (Antonarakis et al., 2008). The texture features of Gray-Level Co-occurrence Matrix (GLCM) such as homogeneity, mean, entropy, correlation, and dissimilarity were employed to classify trees, buildings, and land (Samadzadegan et al., 2010; Huang et al., 2011). Chen and Gao (2014) first adopted a segmentation algorithm to divide images into segments with similar features and then detected pieces of land by calculating distances of segments and applying a few slope operators instead of ground points (Chen and Gao, 2014). The good performance of deep networks in different fields has attracted the attention of researchers in recent years (Li et al., 2018; Mukherji et al., 2022; Mokayed et al., 2023). Hu et al. (2016) employed a deep Convolutional Neural Network (CNN) to classify and differentiate ground points from non-ground points. For this purpose, each point and its neighborhood were extracted and converted into an image in a specified window. In addition to the height values of each point, the height values of neighboring points and the structure of a region at the point of interest were considered in the classification process to improve the results (Hu and Yuan, 2016). Gevaert et al. (2018) used a morphology-based filter first to detect the candidate ground and non-ground points, which they then employed to train a Fully Convolutional Network (FCN). Finally, the trained network was adopted to detect the non-ground points (Gevaert et al., 2018).

Since the available filtering methods are mainly faced with problems in detecting non-ground objects of different dimensions in complicated sloped regions and in extracting the DTM, this paper aims to perform the DTM extraction in such areas with the fewest errors by proposing a framework based on deep learning and Particle Swarm Optimization (PSO) (Shi, 2004; Wang et al., 2023). In this regard, a hierarchical imagery-based deep network was proposed to extract non-ground objects from the point cloud. After the non-ground objects were eliminated, polynomial interpolation was performed to reconstruct the DTM and estimate the eliminated height values. The modified PSO algorithm was employed to extract the optimal polynomial parameters. In designing a PSO, different ranges can be selected for each element of a particle's position and velocity vectors within the  $d$ -dimensional problem space, corresponding to the number of coefficients for the chosen polynomial. Managing the ranges for larger problem spaces can be quite challenging. Equally crucial is defining the maximum velocity change for a particle during an iteration. In our modified PSO proposal, we aimed to address these challenges

by mapping the element values within a specified range in each iteration of computation. The evaluation results of the proposed approach are compared with those of other methods (Chen et al., 2013; Pingel et al., 2013; Mongus et al., 2014; Hui et al., 2016; Bigdeli et al., 2018a) to assess the robustness of the proposed approach. This paper consists of various sections. The proposed methodology is introduced in Section 2, and the implementation results are reported in Section 3. The results are then analyzed and discussed in Section 4. Finally, the research conclusion is drawn in Section 5.

## 2. Methodology

The proposed algorithm for DSM filtering and DTM extraction consists of two main steps, the first of which is to detect non-ground objects and eliminate them from a DSM (Subsection 2.1), and the second step is to interpolate and retrieve the eliminated regions as well as developing a DTM (Subsection 2.2).

### 2.1 Ground Filtering with Hierarchical Deep Network

The proposed deep framework for non-ground object extraction has a hierarchical structure. The main core of this method is a deep network with an encoder-decoder structure consisting of two separate sections for feature extraction in generating the classification map. In the first section called the encoder (Fig. 1: left), various features are extracted from the input image at different levels by using the ResNet-101 structure, in which the dimensions of feature maps decreased to 1.23 of the input images dimensions through five steps and five convolutional blocks based on the residual connection. Shallower levels extract features with more details, whereas deeper levels extract the general features of the image. In fact, the semantic consistency of data is very unsuitable at shallow layers due to the low dimensions of the receptive field and the insufficient context information. By contrast, the semantic consistency of the information is higher at deeper layers due to the higher dimensions of the receptive field; however, the positional information is fairly weak. As a result, each feature extraction block extracts different efficient features that are necessary for the accurate segmentation of an image. Thus, the proposed network was designed to have the capability of extracting and fusing features at various levels for the accurate detection of non-ground objects. Fig. 1 represents the structure of the proposed deep network.

As discussed earlier, the dimensions of the generated features decrease gradually in the first section of the proposed network. However, the dimensions of feature maps increase in the second section (Fig. 1: right), where the non-ground objects are detected and separated. Hence, the structure of the second section should be strong enough to convert the generated features into the final result of detecting objects. In addition, using the features generated at each level of the first section can be effective in detecting non-ground objects more accurately in the second section. For this purpose, the decoder sections of several different blocks were designed alongside the up-sampling layer to generate the feature layers by considering the features extracted from

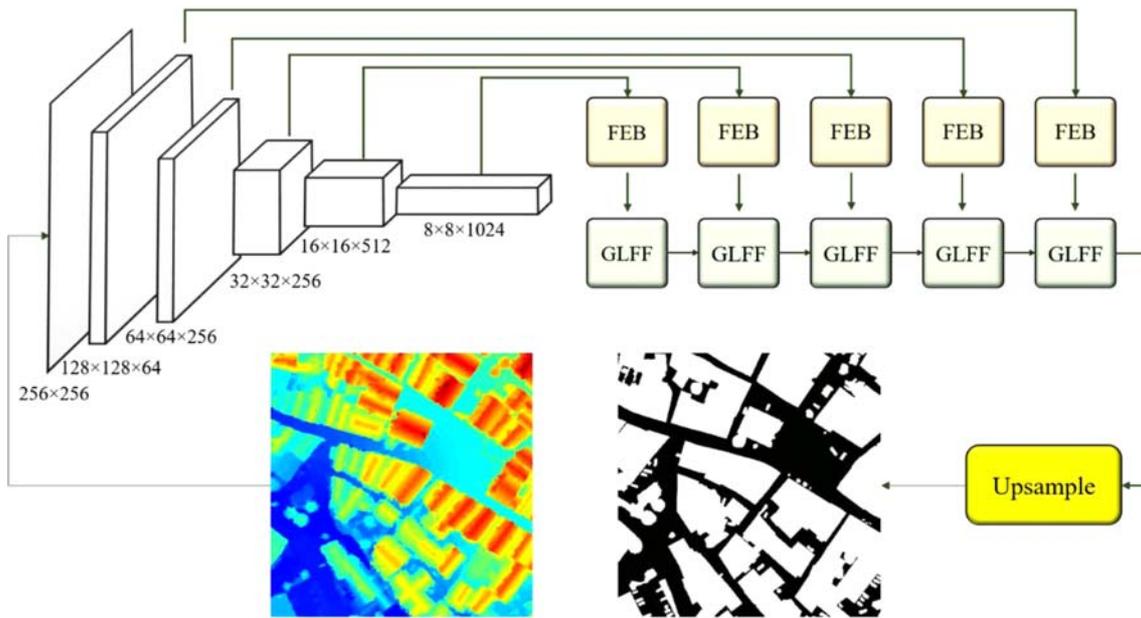


Fig. 1. The Proposed Deep Network with an Encoder-Decoder Structure for the Extraction and Fusion of Features to Detect Non-Ground Objects

previous layers, using the features of the network encoder layers, and fusing them with the layers of the network decoder (Fig. 2).

Inspired by the GoogleNet structure (Ballester and Araujo, 2016), a Feature Extraction Block (FEB) was designed (Fig. 3) to extract features from different levels of the network encoder for fusion with the corresponding levels in the network decoder. For this purpose, four convolutional layers of  $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$  dimensions were employed to extract the context information required at different levels from the block designed in the first section (Figs. 2 and 3). Some points should be taken into account. First, a  $1 \times 1$  convolutional layer was employed to reduce feature dimensions and compress the feature maps. Second,  $1 \times k$  and  $k \times 1$  filters were fused instead of using  $k \times k$  convolutional filters due to preventing the elimination and loss of information in the process of applying convolution. Finally, different

filters were used, and features were extracted from various levels. Moreover, a  $1 \times 1$  filter was applied to fuse the extracted features and correct the number of their channels in order to prepare the final feature map for use in the next steps. Fig. 3 depicts the structure of the designed Feature Extraction Block (FEB).

The features of deep layers provide robust compatible constraints for the positions of objects, whereas shallow layers contain proper local information for pinpointing the objects accurately. The fusion of these data is very useful for acquiring the best accuracy. Therefore, the features extracted from the network encoder should be fused with the corresponding features of the network decoder through an FEB connection within the up-sampling process. For this purpose, a block was designed to fuse local and global features. It is called the Global-Local Feature Fusion (GLFF) block. The designed structure consists of two

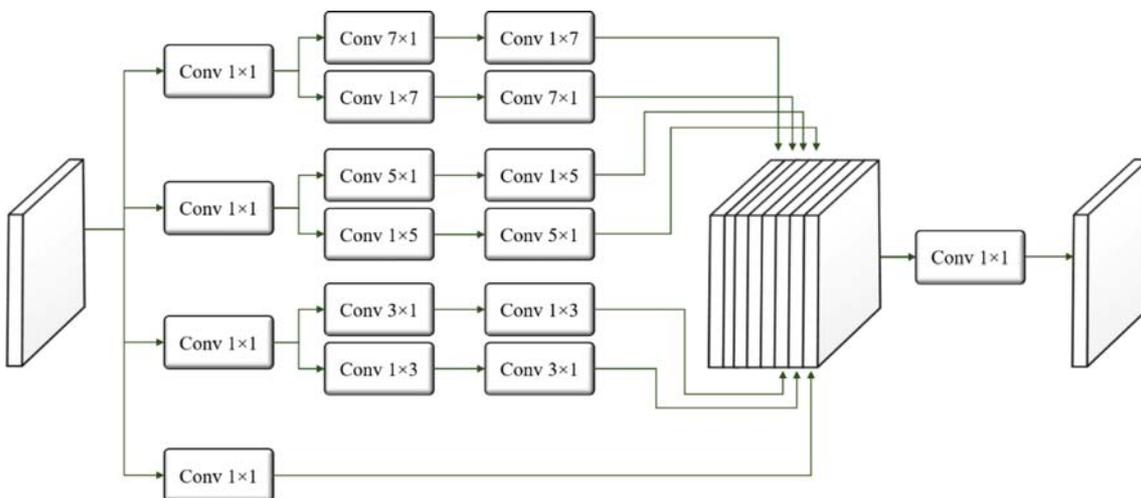


Fig. 2. The Proposed Structure for the Feature Extraction Block (FEB)

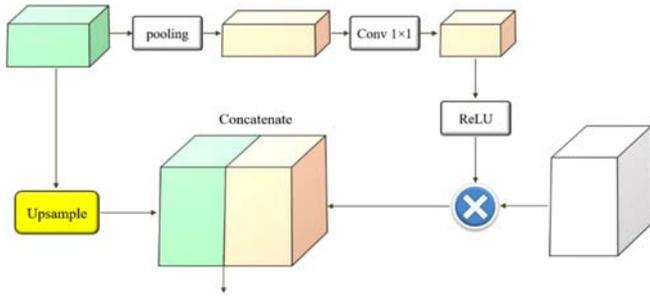


Fig. 3. The Proposed Structure for the GLFF Block

main branches, *i.e.*, weighting data based on the global features and up-sampling data (Fig. 4).

It is very difficult to acquire proper information to extract the relationships of channels only through convolutional operators. In fact, the fusion process should be performed in a way that encodes all the locational information of features in a channel. For this purpose, if it is assumed that  $x_{ij}$  is the numerical value of a feature pixel in a channel  $k$  with  $D_1$ ,  $D_2$ , and  $D_3$  representing the lengths, widths, and number of the feature channels, the following equation can be used for fusion:

$$Z_k = \frac{\sum_{j=1}^{D_1} \sum_{i=1}^{D_2} x_{i,j}}{D_1 \times D_2} \quad (1)$$

In the weighed branch, the result is multiplied by the corresponding features from Section 1 of the network after a  $1 \times 1$  convolutional layer, a batch normalization layer, and an ReLU layer are applied. In fact, this process helps use the data extracted from a deeper layer to direct the information retrieval through a shallower

layer. If  $C$  is assumed a  $1 \times D_1$  column matrix containing the weights pertaining to each channel ( $\omega_i$ ) with  $M$  representing the feature map of interest with  $m_i$  elements, the resultant feature map ( $F$ ) will be a 3D matrix shown as  $D_1 \times D_2 \times D_3$ .

$$F = W \times M = \begin{bmatrix} \omega_1 \\ \dots \\ \omega_c \end{bmatrix} \times \begin{bmatrix} m_1 & \dots & m_{D_3} \end{bmatrix}. \quad (2)$$

In the up-sampling branch, a  $1 \times 1$  convolutional layer is applied to the feature map to set the number of feature map channels. The dimensions of feature maps are then doubled through bilinear interpolation. Finally, the features generated in the first and second branches are merged (Fig. 5). The loss function used for the binary classification and detection on non-ground objects is cross-entropy.

$$e = CrossEntropy(L_e, L_r), \quad (3)$$

where  $L_e$  and  $L_r$  denote the estimated value of the candidate pixel and the reference data value, respectively. The available filtering methods have certain defects such as misidentifying and deleting vegetation or dense urban regions. In many of these algorithms, an iterative-progressive procedure is employed to prevent the abovementioned problem; however, it still persists in challenging regions in practice. The main reason for the failure of these methods would be to use several limited geometrical parameters in the filtering process. By contrast, the proposed method extracts very diverse features within the training process through a deep network. In addition, a vast region is analyzed in each round of detection, something which helps analyze neighborhood in long distances. However, analysis of geometrical parameters through a structural element and slope in the existing methods will lead to the analysis of a limited neighborhood in the point cloud. A hierarchical

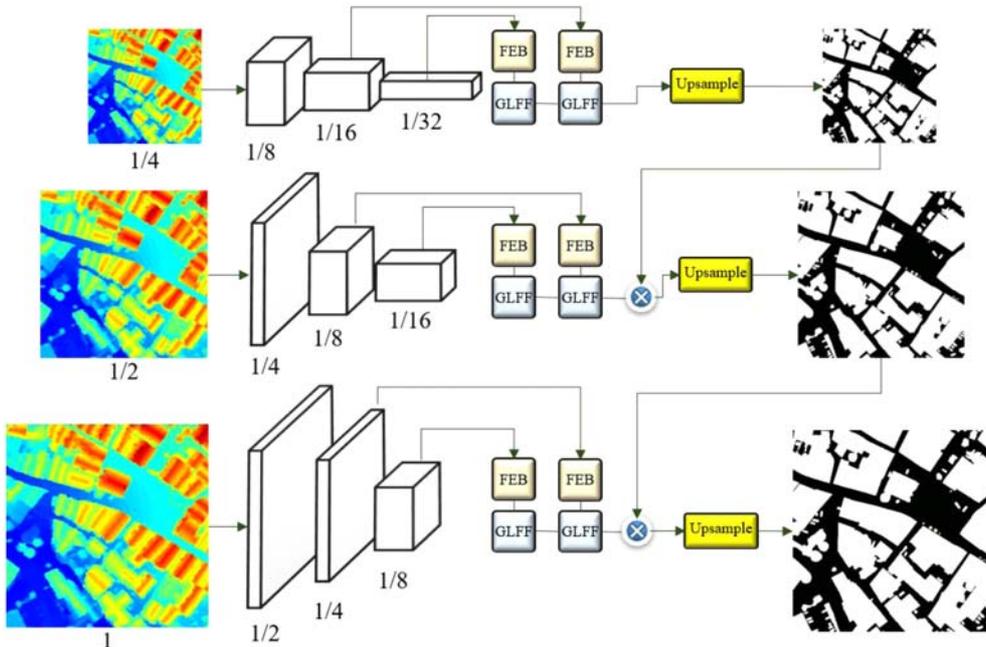


Fig. 4. The Architecture of the Proposed Hierarchical Deep Network for Extracting Non-Ground Objects of Different Dimensions

deep structure was proposed to solve the problem of detecting and eliminating objects with different dimensions (Fig. 5).

## 2.2 Interpolation with Modified PSO

After the non-ground objects are detected and eliminated, the height values of the eliminated regions should be retrieved to extract the final DTM. Since many parts of a surface are sometimes eliminated due to vegetation or dense urban regions, interpolation and retrieval of height values should be performed accurately in order to generate the Earth's surface correctly. In this subsection, the proposed modification is applied to the PSO algorithm to determine the optimal polynomial coefficients for the interpolation and retrieval of the height information. In the proposed PSO algorithm, the particle population is initialized randomly to search for the global optimum in an iterative process. First,  $m_i^x$  and  $n_i^x$  are assumed to be the lower and upper bounds of each  $x_k(i)$ , respectively, whereas  $m_i^v$  and  $n_i^v$  are assumed to be the lower and upper bounds of  $v_k(j)$ , respectively. Moreover,  $l$  and  $t$  represent the number of particles and the number of loop iterations, respectively. Therefore, the position vector and the velocity vector of particle  $k$  ( $k \in \{1, 2, \dots, l\}$ ) in the  $d$ -dimensional space will be as below:

$$\underline{x}_k = [x_k(1) \ x_k(2) \ \dots \ x_k(i)], \quad i \in \{1, 2, \dots, d\}, x_k(i) \in [m_i^x \ n_i^x], \quad (4)$$

$$\underline{v}_k = [v_k(1) \ v_k(2) \ \dots \ v_k(i)], \quad i \in \{1, 2, \dots, d\}, v_k(i) \in [m_i^v \ n_i^v], \quad (5)$$

Due to the topography and complexity of the study area, the quadratic polynomial coefficients were optimized. According to the above equations, different ranges can be selected for dimensions of  $\underline{v}_k$  and  $\underline{x}_k$  based on the problem that should be optimized. Hence, the wider the search space, the more difficult it is to determine the dimension domain of  $\underline{v}_k$ . The next challenge is to determine the maximum changes ( $v_{max}$ ) that a particle might have experienced in an iteration. The range of each element  $\underline{x}_k$  in the PSO algorithm is normalized within  $[-1, 1]$  to overcome the challenges.

$$x_k^N(i) = \frac{x_k(i) - \min(x_k(i)|_{i=1}^d)}{\max(x_k(i)|_{i=1}^d) - \min(x_k(i)|_{i=1}^d)}, \quad (6)$$

$$x_k(i) \in [m_i^x \ n_i^x], x_k^N(i) \in [-1 \ 1],$$

$$\underline{x}_k^N = [x_k^N(1) \ x_k^N(2) \ \dots \ x_k^N(i)]. \quad (7)$$

Furthermore, every element  $\underline{v}_k$  is initialized randomly within  $(-1, 1)$ . Hence, the following equation is employed to determine the velocity of each particle for the next iteration between  $\underline{x}_k^N$  pertaining to the best particle  $k^{th}$  called  $\underline{x}_{pbk}^N(t)$  and  $\underline{x}_k^N$  pertaining to the best particle of all iterations called  $\underline{x}_{gb}^N(t)$ :

$$w(t) = w_{max} - \left( \frac{w_{max} - w_{min}}{t_{max}} \right) t, \quad t \in \{1, 2, \dots, t_{max}\}, \quad (8)$$

$$\underline{v}_k(t+1) = w(t)\underline{v}_k(t) + c_1 r_1 (\underline{x}_{pbk}^N(t) - \underline{x}_k^N(t)) + c_2 r_2 (\underline{x}_{gb}^N(t) - \underline{x}_k^N(t)), \quad (9)$$

$$r_1, r_2 \in U(0, 1),$$

where  $r_1$  and  $r_2$  are random numbers between 0 and 1, whereas  $c_1$  and  $c_2$  are the coefficients of self-recognition and social components. Every  $\underline{x}_k$  is used to calculate the fitness value of each particle in each iteration ( $t$ ). The values of  $\underline{x}_{pbk}^N(t)$  and  $\underline{x}_{gb}^N(t)$  are then calculated through the following equations:

$$\underline{x}_{pbk}^N(t) = \arg \min \{ f(\underline{x}_k(t)) |_{t=1}^{t_{max}} \}, \quad (10)$$

$$\underline{x}_{gb}^N(t) = \arg \min \{ f(\underline{x}_{gb}^N(t-1)), f(\underline{x}_k(t)) \}. \quad (11)$$

Finally, the new position of particle  $k$  is determined through the following equation:

$$\underline{x}_k^N(t+1) = \underline{x}_k^N(t) + \underline{v}_k(t+1); \quad (12)$$

Moreover, RMSE is the evaluation criterion for interpolation in the PSO algorithm. It is obtained from the following equation:

$$E_{rms} = \sqrt{\frac{\sum_{i=1}^n (d_e - d_r)^2}{n}}. \quad (13)$$

where  $d_e$  and  $d_r$  represent the estimated height and the reference height, respectively. The pseudocode of the modified PSO algorithm is presented now to obtain the polynomial coefficients.

**Inputs:**  $n, w_{min}, w_{max}, V_{max}, t_{max}, m_i^x, n_i^x$

**Output:** The optimum values polynomial coefficients

1. Generate a  $d$ -dimensional particle position vector [Eq. (4)], randomly between  $[m_i^x \ n_i^x]$ ;
2. Generate a  $d$ -dimensional particle velocity vector [Eq. (5)], randomly between  $(-1 \ 1)$ ;
3. Compute  $\underline{x}_k^N$  [Eq. (4)] for all particles in iteration  $t=1$ .
4. **for**  $t = 1$  to  $t_{max}$ ;
5. Compute  $w(t)$  [Eq. (8)];
6. **end**
7. **for**  $k = 1$  to  $n$ ;
8. Compute the fitness [Eq. (13)];
9. **end**
10. Compute  $\underline{x}_{pb}^N(1)$  or  $\underline{x}_{gb}^N(1)$ ;
11. **for**  $k = 1$  to  $n$ ;
12. Update the velocity vector [Eq. (8)];
13. **if**  $\underline{v}_k^j > v_{max}$
14.  $\underline{v}_k^j = v_{max}$
15. **elseif**  $\underline{v}_k^j < -v_{max}$
16.  $\underline{v}_k^j = -v_{max}$
17. **end**
18. Update the position vector [Eq. (12)];
19. **end**
20. **while**  $t \leq t_{max}$  **do**
21. **for**  $k = 1$  to  $n$ ;
22. Compute the fitness [Eq. (13)];
23. **end**
24. Compute  $\underline{x}_{pbest}^N(t)$  and  $\underline{x}_{gbest}^N(t)$  [Eqs. (10) and (11)];
25. **for**  $k = 1$  to  $n$ ;
26. Update the velocity vector [Eq. (8)];
27. **if**  $\underline{v}_k^j > v_{max}$
28.  $\underline{v}_k^j = v_{max}$
29. **elseif**  $\underline{v}_k^j < -v_{max}$

```

30.  $v_k^j = -v_{max}$ 
31. end
32. Update the position vector [Eq. (12)];
33.  $t = t + 1$ ;
34. end
35. return  $x_{glt}(t)$ .

```

### 3. Experimentation and Evaluation

In this section, the research dataset is first presented to implement the proposed algorithm (Subsection 3.1). The results of detecting non-ground points through the proposed hierarchical deep network are then reported in Subsection 3.2, and the results of the modified PSO algorithm are ultimately presented in Subsection 3.3 to develop the final DTM.

#### 3.1 Study areas and Data Used

The 3D point cloud obtained from LiDAR was employed to implement the proposed algorithm. The captured elevation data by LiDAR are used for our implementation. The study areas and their features are described in the next paragraph. In fact, the study areas were selected in a way that would allow the thorough analysis of the proposed algorithm for filtering and extracting a DTM from a DSM.

**The US Dataset:** This dataset includes some LiDAR data collected from Lake City, Utah in the US within the 2013 – 2014 period. Covering an area of 1360 km<sup>2</sup>, the data density was nearly 11.14 pts/m<sup>2</sup>. Four regions of this dataset with approximate areas of 22 km<sup>2</sup> were selected to implement and analyze the proposed method (Regions 1 – 4 in Fig. 5).

**ISPRS Dataset:** This dataset is the ISPRS benchmark that is gathered by LiDAR. These data were collected by FOTONOR AS through the OpTech ALTM scanner. Since the collected data had different densities of points and covered various regions with different features, it is very suitable for the evaluation of algorithms proposed by researchers. The study areas of the US are represented in the following figure along with ISPRS samples (Samples 1 – 1 to 7 – 1 in Fig. 5).

#### 3.2 Ground Filtering with Hierarchical Deep Network

The proposed deep network is based on imagery; therefore, interpolation should be applied to convert the input data into an image if these data are given in the point cloud. For this purpose, the modified PSO algorithm was employed to convert the point cloud into a height image with 0.5 m spatial resolution. To this end, the modified PSO computes the optimized polynomial coefficients for generating height images using the point cloud coordinates (Section 2.2). In addition, remote sensing data mainly cover vast areas, in which it is impossible to directly use all regions simultaneously in the deep network. Hence, it is essential to collect the images that match the acceptable input dimensions to train the network (256 × 256). Deep networks have millions of parameters, the tuning of which through the network training procedure requires a large amount of training

data. It is often challenging to collect this amount of training data; therefore, data augmentation methods were used effectively to increase the training data. These methods often include applying certain transforms to the input and target data. The following operations were performed in this paper for data augmentation.

**Flipping:** The input and target data are flipped horizontally and vertically.

**Rotation:** The input and target are rotated with respect to a random value.

**Translation:** Some of the input and target data are cropped randomly.

The US study area covers a vast region with diverse types of vegetation and different buildings. Hence, Area 1 of the US study area was selected to train the network (Fig. 5), and the other areas were utilized to test the network. In this regard, after cropping and applying dataset augmentation about 12000 samples are generated that 80% of them are selected randomly for training and the remained are selected for validation. The number of epochs, batch size, weight decay, learning rate, and moment for both areas were considered 200, 8,  $3 \times 10^{-4}$ ,  $10^{-4}$ , and  $10^{-4}$ , respectively. Fig. 6 reports the results of detecting non-ground objects in the study areas.

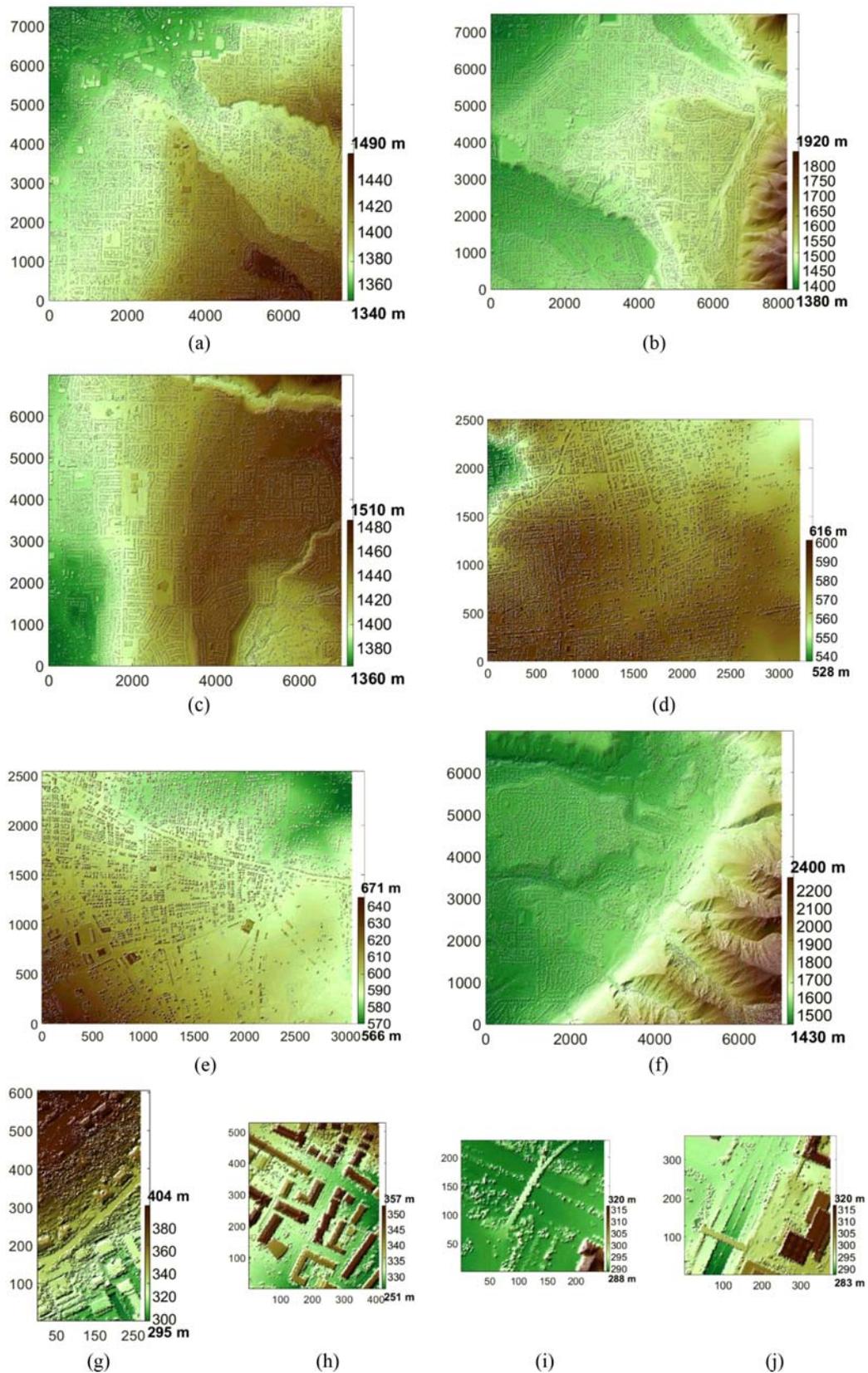
Three parameters called  $E_{T1}$ ,  $E_{T2}$ , and  $E_{Total}$  were employed to evaluate the results of filtering and detecting non-ground objects. These parameters are calculated through the following equations:

$$E_{T1} = \frac{b}{a+b}, \quad (14)$$

$$E_{T2} = \frac{c}{c+d}, \quad (15)$$

$$E_T = \frac{b+c}{a+b+c+d}, \quad (16)$$

where  $a$  indicates the ground points detected correctly, and  $b$  refers to the ground points misidentified as non-ground points. Moreover,  $c$  indicates the non-ground points misidentified as ground points, whereas  $d$  represents the non-ground points extracted correctly. Kappa ( $E_{Kappa}$ ), a widely used standard for object detection and classification, was employed to gauge the accuracy of detection (Houtte et al., 2011).  $E_{Kappa}$  is a metric that measures the agreement between observed and expected classifications, often used to assess the performance of classification models. Accordingly, the results of detecting the non-ground points through the proposed deep network were compared with the results of other famous algorithms such as LAsTools (Axelsson, 2000), gLidar (Mongus et al., 2014), and CFS (Zhang et al., 2016). Based on structural and geometrical characteristics, these algorithms try to detect non-ground objects; thus, they need to determine such parameters. Different values of parameters were analyzed to implement those algorithms, and the best values were selected to develop a suitable DTM. Table 1 reports the



**Fig. 5.** The Study Areas of America (a-d) and ISPRS (e-s), (a) Area 1, (b) Area 2, (c) Area 3, (d) Area 4, (e) Area 5, (f) Area 6, (g) Sample 1-1, (h) Sample 1-2, (i) Sample 2-1, (j) Sample 2-2, (k) Sample 2-3, (l) Sample 2-4, (m) Sample 3-1, (n) Sample 4-1, (o) Sample 4-2, (p) Sample 5-1, (q) Sample 5-2, (r) Sample 5-3, (s) Sample 5-4, (t) Sample 6-1, (u) Sample 7-1

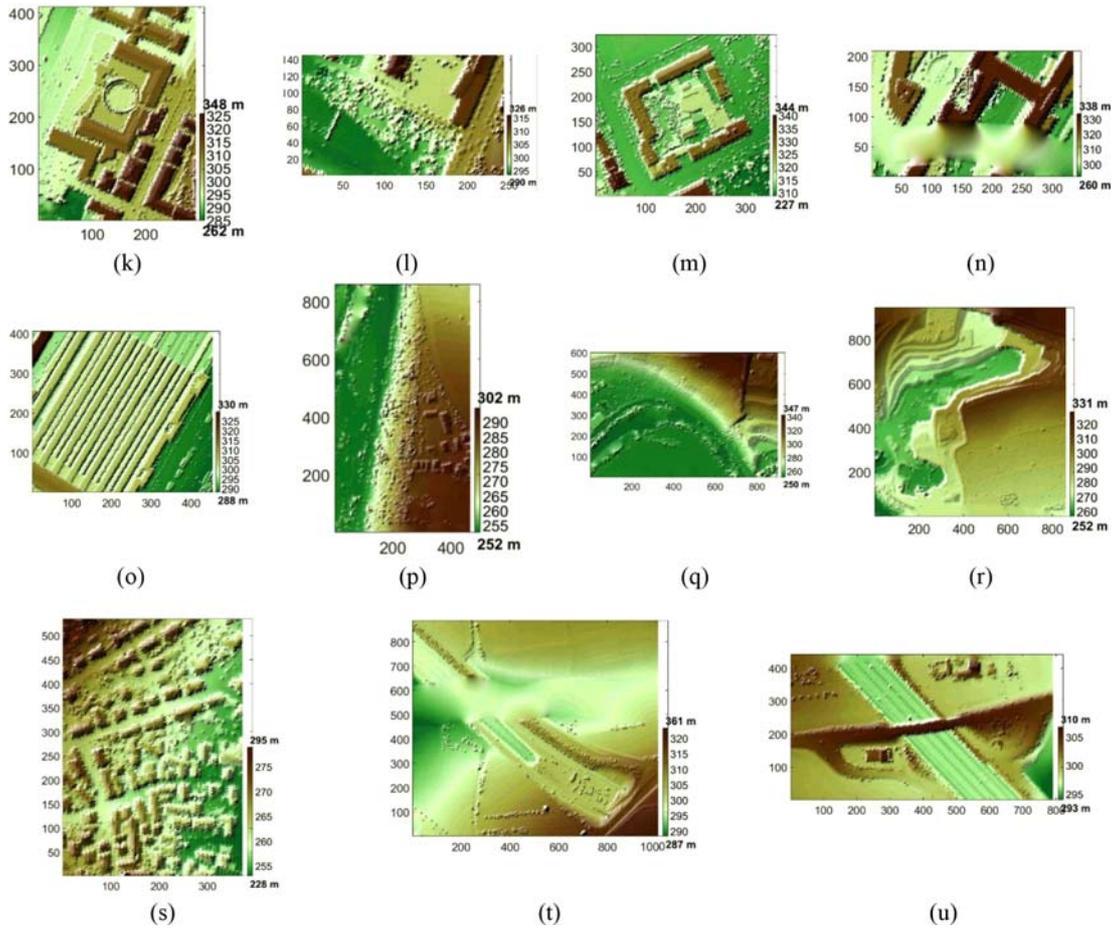


Fig. 5. (continued)

calculated values of  $E_{T1}$ ,  $E_{T2}$ , and  $E_T$  for the evaluation of results obtained from filtering and detecting non-ground objects through the proposed deep algorithm in comparison with LAsTools, gLidar, and CSF algorithms. Moreover, the results of the proposed hierarchical deep network (Fig. 5) are presented along with the proposed encoder-decoder network (Fig. 2) for performance evaluation.

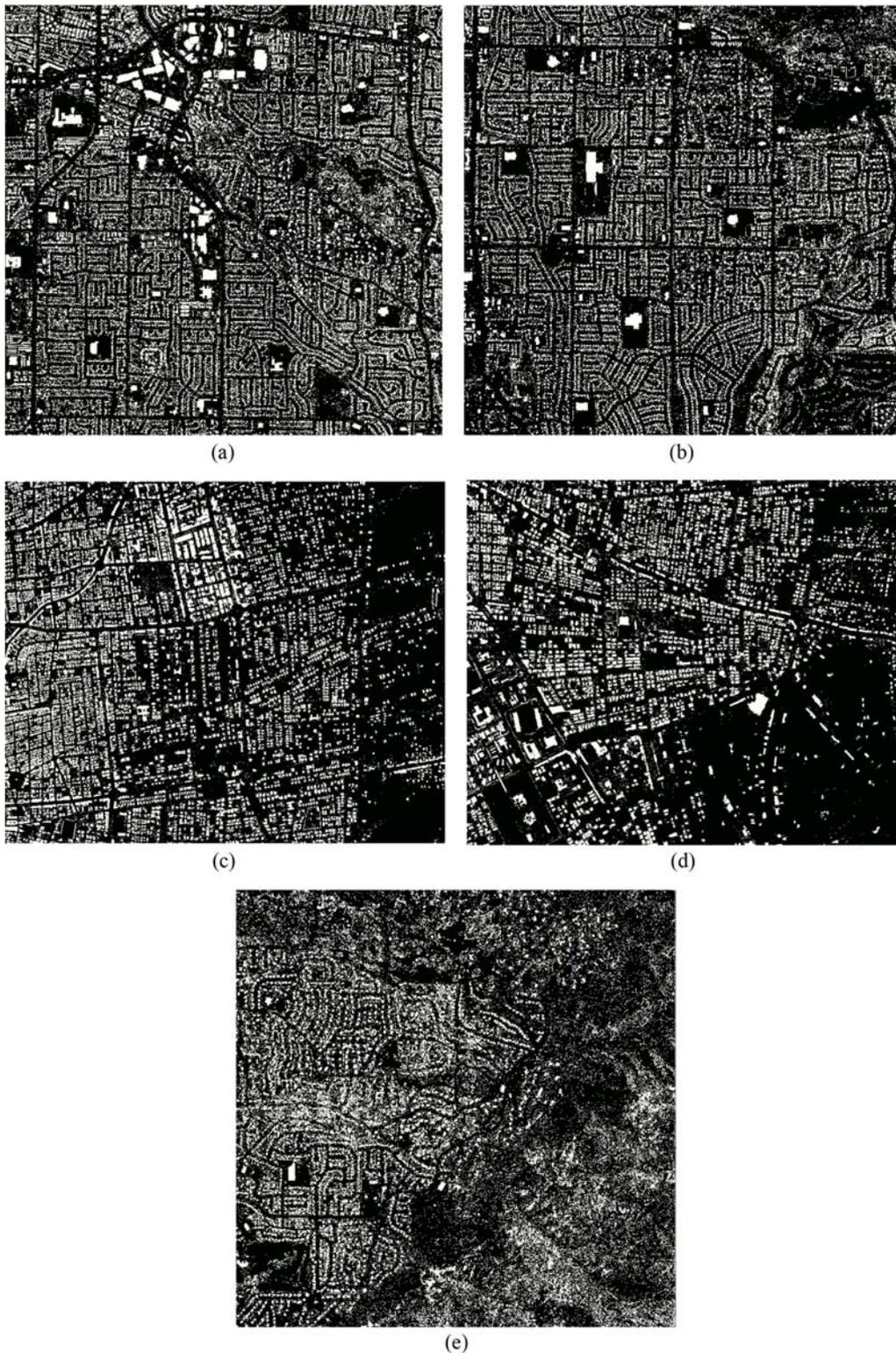
In Table 1, it can be seen that the obtained values of  $E_{T1}$  are higher than other areas, which indicates that more pixels are wrongly identified as non-ground pixels. This area has a rougher topography compared to other areas and has caused relatively severe height changes in small distances. Therefore, it becomes difficult to separate ground pixels that are associated with severe height changes and non-ground objects that also have sudden height changes. Fig. 7 presents the qualitative evaluation results. Each parameter (*i.e.*,  $a$ ,  $b$ ,  $c$ , and  $d$ ) is highlighted in different colors for clarity. Therefore, it is easy to perceive in what areas an algorithm failed to detect non-ground objects and in what areas there were errors in the detection of non-ground objects. Fig. 7 also reports the results of gLidar, CFS, and LAsTools algorithms.

The performance of the proposed network was then evaluated in filtering non-ground points in comparison with ISPRS data.

For this purpose, Table 2 presents the results of filtering through the proposed network as well as the outputs of famous algorithms (Chen et al., 2013; Pingel et al., 2013; Mongus et al., 2014; Hui et al., 2016; Bigdeli et al., 2018a). In addition, Fig. 8 reports the qualitative evaluation results of the proposed network outputs.

### 3.3 Interpolation with Modified PSO

This subsection presents the trend in the results of extracting the final DTM after the non-ground objects were detected and eliminated. As discussed earlier, the point cloud data should be converted into image data through interpolation to train the proposed deep network. It is essential to reach and maintain high accuracy in converting the point cloud data into an image, for the lower the conversion accuracy, the milder the ridges of objects. As a result, the accuracy of detecting non-ground objects will decrease in the filtering process. Hence, the modified PSO algorithm was employed to interpolate and convert the point cloud into an image. To evaluate the performance of the modified PSO algorithm, 60% of points of an area were used randomly to calculate the polynomial coefficients, and the rest of the data were employed to evaluate the estimated height values. Table 3 presents the results of evaluating the modified PSO algorithm in comparison with some other famous interpolation algorithms



**Fig. 6.** The Results of Detecting Non-Ground Objects in the Dataset and the US Through the Proposed Deep Networks: (a) Area 2, (b) Area 3, (c) Area 4, (d) Area 5, (e) Area 6

such as kriging, IDW, spline, bilinear, and cubic methods. Moreover, Table 3 reports the result of using only the PSO algorithm to calculate polynomial coefficients for a better analysis. Eq. (13) ( $E_{RMSE}$ ) was adopted to calculate the DTM error rate.

The modified PSO algorithm was then employed to retrieve the height values of the eliminated regions after the non-ground objects were deleted and the final DTM was developed. The US study areas, which have a reference DTM, were used to evaluate

**Table 1.** The Quantitative Evaluation of Results from Filtering and Detecting Non-ground Objects Through the Proposed Deep Network in Comparison with gLidar, LAStools, and CSF Algorithms

	Method	$E_{T1}$	$E_{T2}$	$E_T$	$E_{\text{Kappa}}$
Area 2	gLidar	0.051	0.002	0.013	96.17
	CSF	0.052	0.021	0.028	87.37
	LAStools	0.029	0.064	0.057	76.29
	Proposed Encoder-decoder network	0.003	0.0156	0.012	96.34
	Proposed multi-scale network	0.001	0.007	0.005	98.36
Area 3	gLidar	0.131	0.007	0.032	89.43
	CSF	0.055	0.019	0.026	86.41
	LAStools	0.034	0.046	0.044	78.04
	Proposed Encoder-decoder network	0.002	0.021	0.018	93.63
	Proposed multi-scale network	0.001	0.012	0.010	96.55
Area 4	gLidar	0.331	0.031	0.138	67.59
	CSF	0.290	0.057	0.127	68.22
	LAStools	0.254	0.013	0.092	72.64
	Proposed Encoder-decoder network	0.063	0.003	0.019	94.42
	Proposed multi-scale network	0.032	0.002	0.014	96.12
Area 5	gLidar	0.406	0.018	0.138	63.87
	CSF	0.248	0.020	0.075	77.96
	LAStools	0.216	0.038	0.102	78.03
	Proposed Encoder-decoder network	0.080	0.030	0.024	94.01
	Proposed multi-scale network	0.093	0.013	0.021	92.44
Area 6	gLidar	0.478	0.083	0.190	47.55
	CSF	0.516	0.037	0.214	49.25
	LAStools	0.032	0.077	0.068	68.10
	Proposed Encoder-decoder network	0.185	0.011	0.052	84.57
	Proposed multi-scale network	0.155	0.012	0.040	90.48

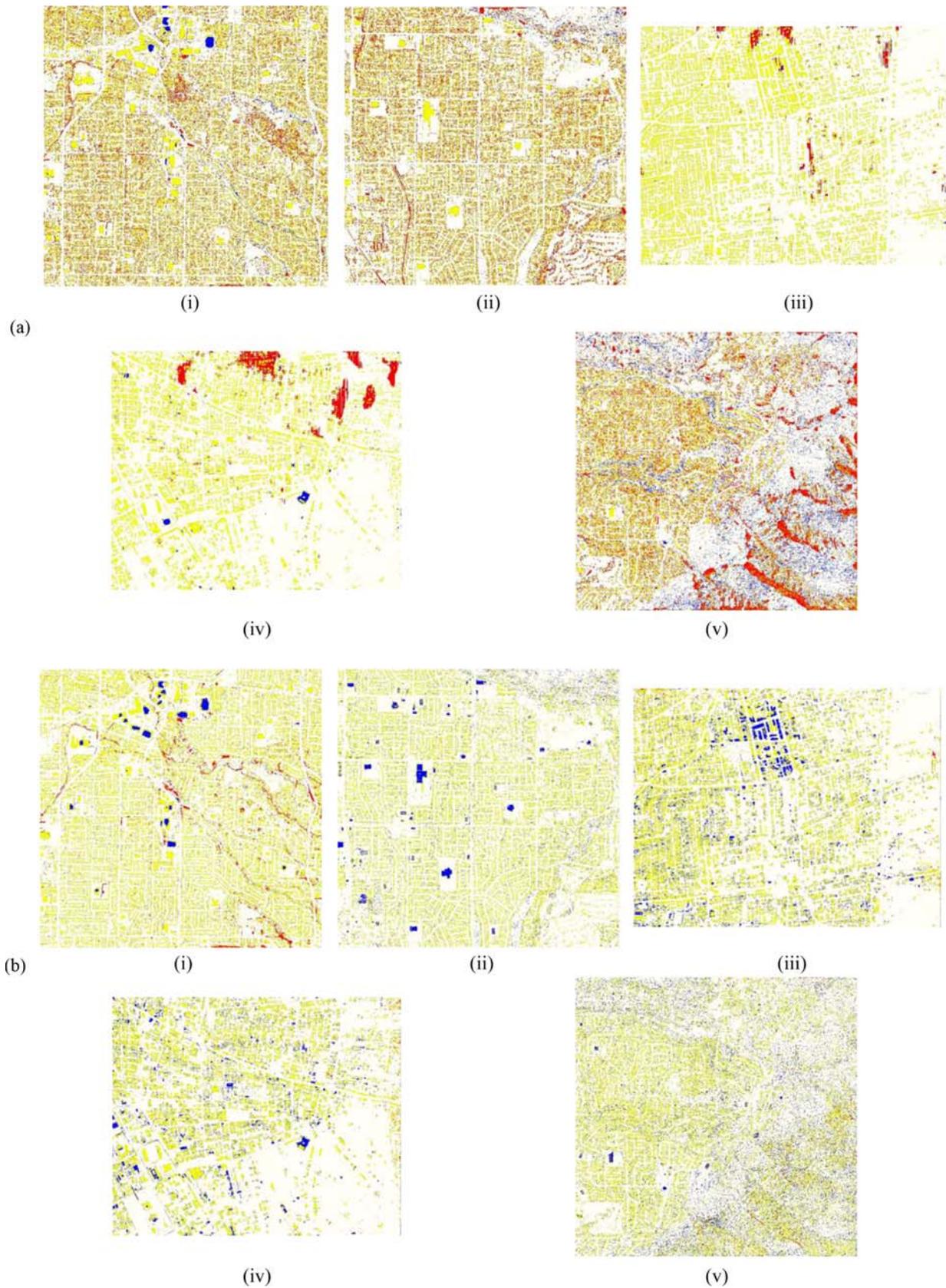
the final DTM. Area 1, Area 2, and Area 3 were characterized by 0.157, 0.120, and 0.245, respectively. Fig. 9 indicates the finally extracted DTMs and the differential DTM obtained from the subtraction of the DTM extracted through the proposed method from the reference DTM.

#### 4. Discussion

As discussed earlier, the proposed deep networks were trained through Area 1 of the study areas (Fig. 5). The other areas were used for testing and evaluating the proposed network. According to Table 1, the proposed encoder-decoder network outperformed LAStools, CSF, and gLidar algorithms by far. In other words, the results of the proposed network had lower values of  $E_{T1}$ ,  $E_{T2}$ , and  $E_T$ ; therefore, they achieved more accurate values of kappa (Table 1). The major weakness of the conventional filtering methods lies in the detection of non-ground objects in sloped regions and ridge lines of mountains that are very uneven (Fig. 7). In fact, many of the ground points are misidentified as non-ground points due to drastic height changes of the ground in these areas (see red regions in Fig. 7). However, the proposed deep network was very efficient in this regard (Figs. 7 and 10).

The proposed hierarchical structure managed to enhance the performance of the proposed deep network. In fact, it improved the final results by eliminating objects with different dimensions, especially objects with large areas, in the conventional filtering methods where it is difficult to delete such objects. According to Fig. 10, the gLidar, CSF, and LAStools algorithms faced problems in deleting buildings with large areas. Changing the parameters did not improve their outputs. In these algorithms, emphasis on the elimination of high-dimensional objects can lead to the selection some ground objects as non-ground objects in steep slopes, something which adversely affects the accuracy of results. The proposed deep encoder-decoder network outperformed the other methods in deleting such objects; however, the objects were not eliminated completely. However, the proposed hierarchical deep network was very successful in this regard (Fig. 11).

The performance of the proposed hierarchical deep network was also noteworthy in comparison with the ISPRS data. Although the network was trained through different methods, the proposed network outperformed the other powerful techniques. Furthermore, the results of the algorithms presented in Table 2 were obtained by selecting the optimal parameters. In other words, the algorithm proposed by Pingel et al. (2013) and the



**Fig. 7.** The Qualitative Evaluation of Results from Filtering and Detecting Non-Ground Objects: (a) gLidar, (b) LAStools, (c) CSF, (d) the Proposed Deep Encoder-Decoder Network, (e) the Proposed Hierarchical Deep Network, (i) Area 2, (ii) Area 3, (iii) Area 4, (iv) Area 5, (v) Area 6 (a = yellow; c = blue; b = red; and d = white)

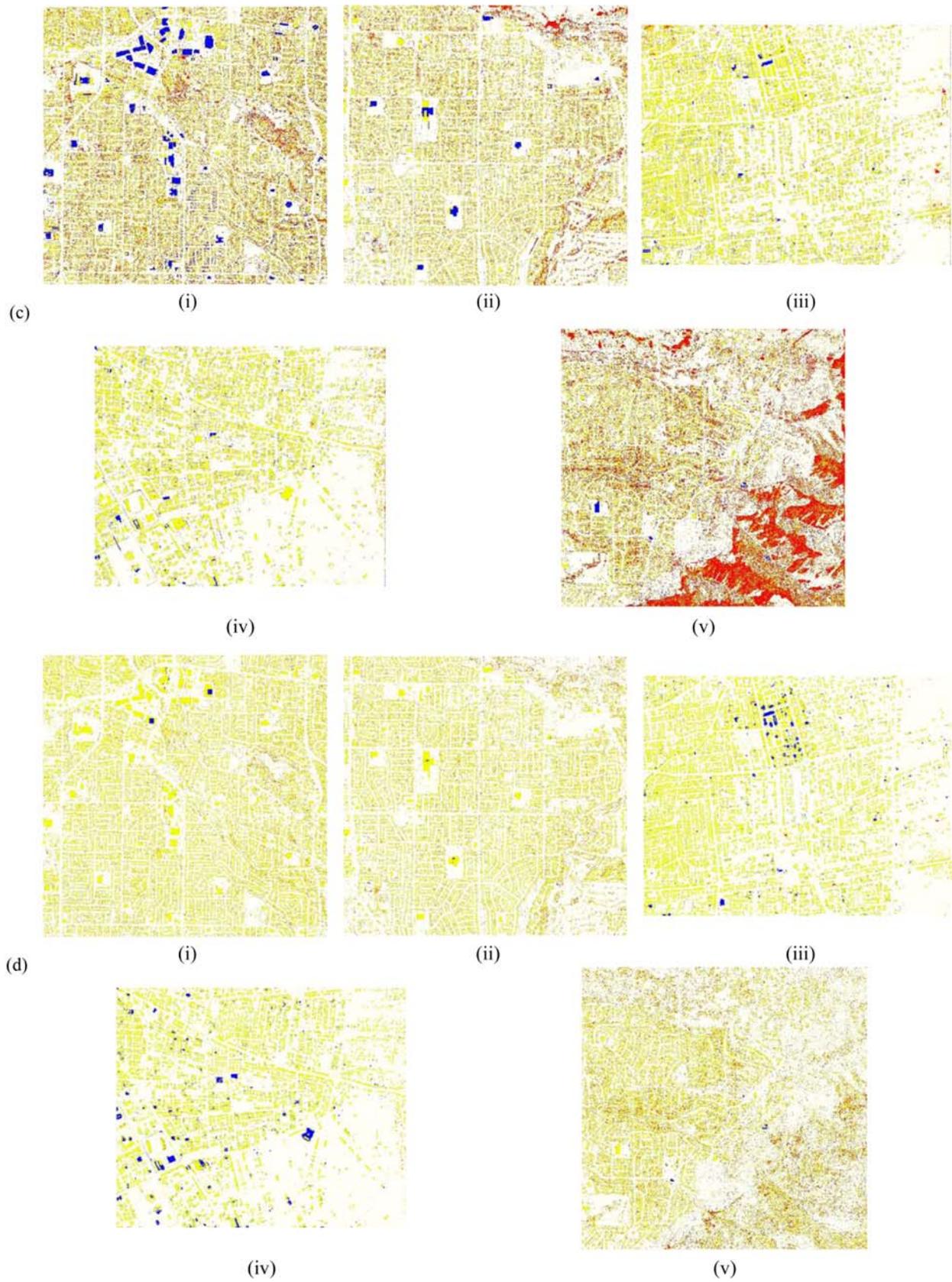


Fig. 7. (continued)

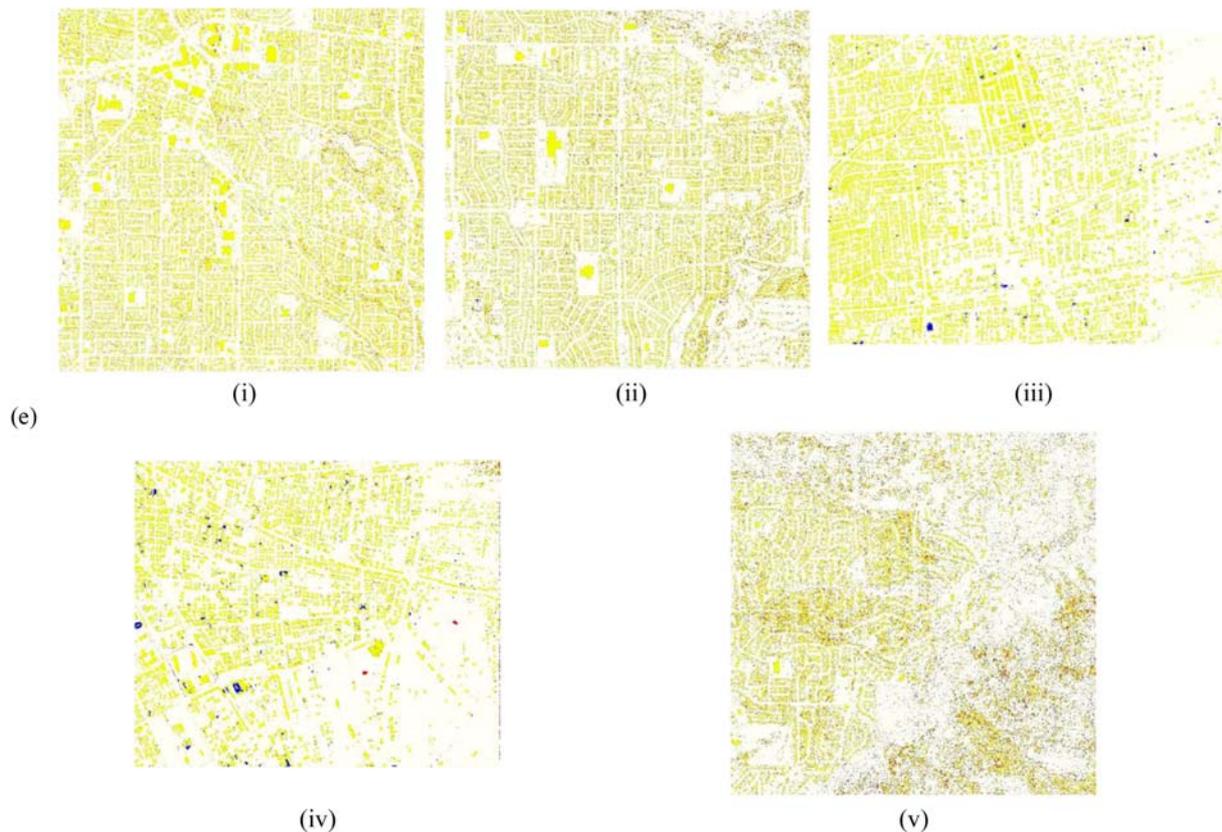


Fig. 7. (continued)

algorithm proposed by Mongus et al. (2014) needed to determine the values of four parameters to achieve optimal results. Moreover, the algorithm proposed by Chen et al. (2013) needed to determine two optimal parameters in the area to acquire the most accurate results. Nevertheless, there were no parameters for optimization based on the topography of the study area after the network was trained properly in the proposed method, something which indicates the high generalizability of the proposed method.

According to Table 3, the performance evaluation of the modified PSO algorithm indicated its much higher accuracy than those of the other interpolation algorithms. In other words, the RMSE of the modified PSO algorithm was 0.150 cm on average, which is significantly better than those of the other methods. The conventional interpolation methods are often inefficient in the areas where there are sudden changes of height. In this case, these methods usually turn sharp ridges into mild ridges and change height gradually, which directly affects the filtering results and decreases their accuracy in detecting non-ground objects.

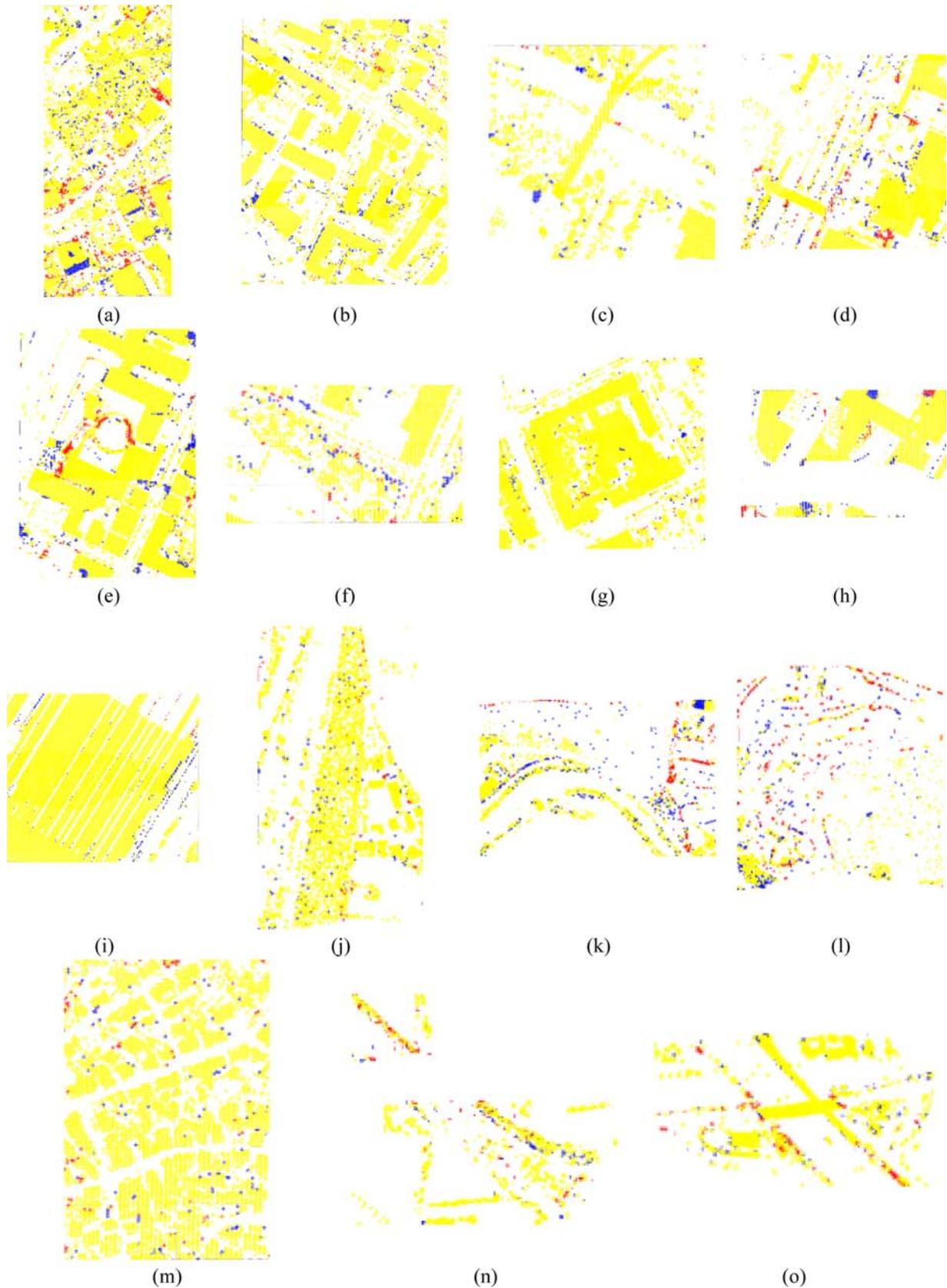
## 5. Conclusions

This paper proposed a hierarchical deep network to filter and detect non-ground objects through the point cloud by using an encoder-decoder structure as the main core consisting of a feature extraction block and a global-local feature fusion block.

The proposed hierarchical deep network was designed to detect non-ground objects in urban areas with drastic height changes, dense vegetation, and complicated topography by extracting global and local features as well as fusing those features gradually. The data of Utah, the US were employed to test the proposed trained network, and indicated the detection of objects with the mean kappa of 94.79. Moreover, the values of  $E_{T1}$ ,  $E_{T2}$ , and  $E_T$  were reported 0.056, 0.009, and 0.018, respectively, which indicated the much lower error rate of the proposed network than those of the CSF, LAsTools, and gLidar algorithms. According to the results, the proposed hierarchical structure improved the performance of the encoder-decoder network (Table 1). In other words, the average values of  $E_{kappa}$ ,  $E_{T1}$ ,  $E_{T2}$ , and  $E_T$  were reported 92.59, 0.066, 0.016, and 0.025, respectively, for the encoder-decoder network. The performance of the proposed network was compared with the outputs of the existing methods by using the ISPRS data, the results of which demonstrated improvements in the detection accuracy as opposed to those of the other methods. After the non-ground objects were detected and deleted, the modified PSO algorithm was employed to calculate the optimal polynomial coefficients for interpolation and development of the final DTM. To evaluate the performance of the modified PSO algorithm in calculating the height values, 60% of the study areas were utilized to calculate the polynomial coefficients, whereas the rest of the data were used for testing. The results indicated that the proposed PSO algorithm was characterized by only a

**Table 2.** Comparing the Proposed Deep Network with the Other Reviewed Algorithms

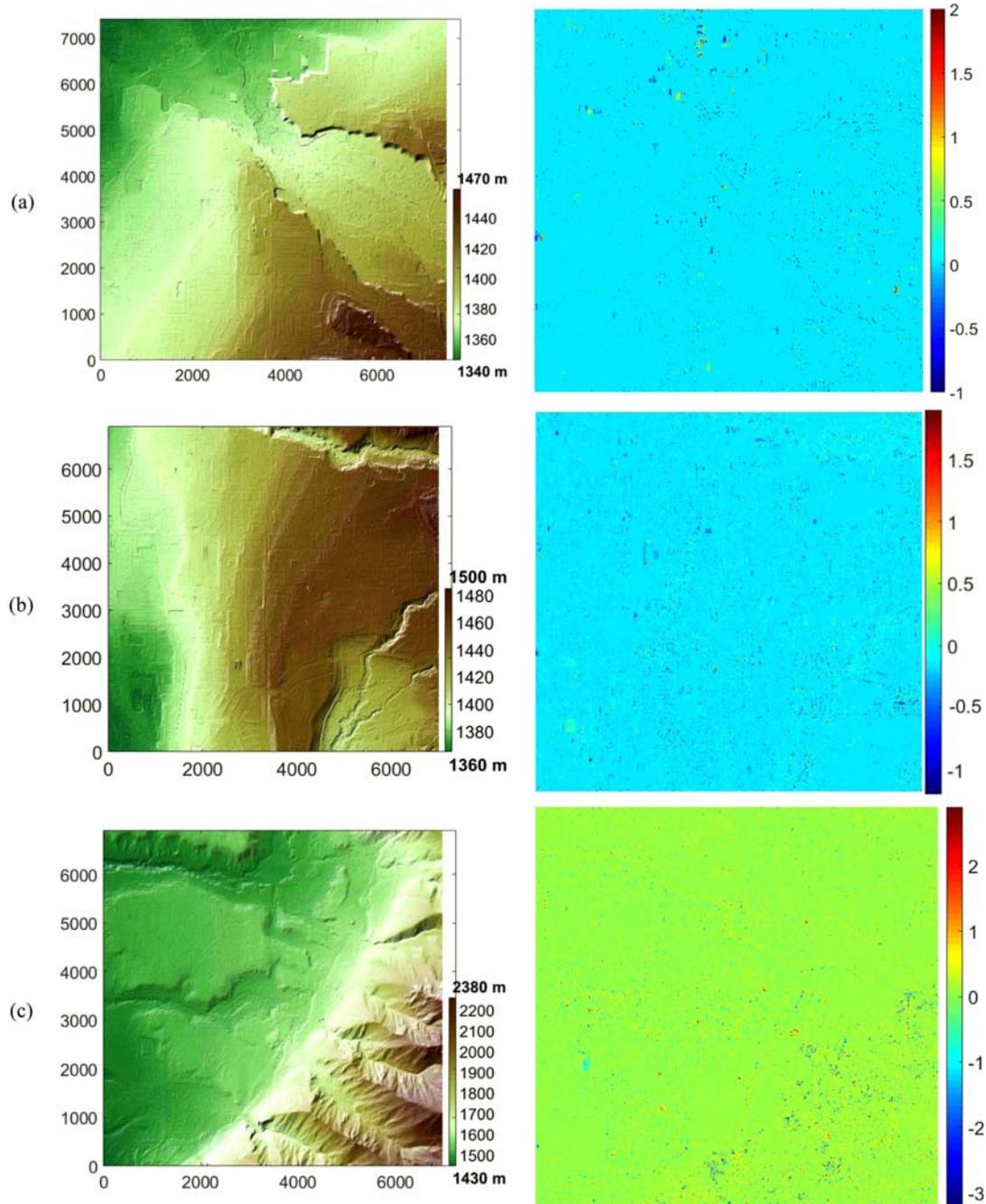
Sample	Error (%)	Pingel	Chen	Mongus	Bigdeli	Hui	Proposed encoder-decoder network	Proposed hierarchical network
1 (1-1)	$E_{T1}$	7.88	19.25	6.17	8.03	13.63	6.65	6.10
	$E_{T2}$	8.81	4.62	9.28	11.98	12.96	11.85	8.51
	$E_T$	8.28	13.01	7.50	9.71	13.34	8.35	7.23
2 (1-2)	$E_{T1}$	2.57	3.99	2.28	3.59	4.86	2.25	2.21
	$E_{T2}$	3.30	2.75	2.84	2.11	2.08	5.08	3.13
	$E_T$	2.92	3.38	2.55	2.84	3.50	3.63	2.73
3 (2-1)	$E_{T1}$	0.26	0.75	0.37	0.60	0.01	0.16	0.15
	$E_{T2}$	4.07	3.41	4.28	0.24	9.95	4.35	3.66
	$E_T$	1.10	1.34	1.23	1.00	2.21	1.09	0.92
4 (2-2)	$E_{T1}$	2.57	2.65	1.79	2.79	5.27	2.60	2.22
	$E_{T2}$	5.07	9.11	5.12	4.10	5.74	4.93	4.13
	$E_T$	3.35	4.67	2.83	3.20	5.41	3.31	2.74
5 (2-3)	$E_{T1}$	3.21	4.45	3.26	2.44	4.00	3.61	3.23
	$E_{T2}$	6.17	6.11	5.55	6.12	6.35	6.18	5.59
	$E_T$	4.61	5.24	4.34	4.18	5.11	4.83	4.01
6 (2-4)	$E_{T1}$	2.25	5.54	2.34	2.63	7.47	2.02	1.92
	$E_{T2}$	6.90	8.26	6.85	8.99	7.48	6.51	6.02
	$E_T$	3.52	6.29	3.58	4.38	7.47	3.35	3.06
7 (3-1)	$E_{T1}$	0.39	0.55	0.31	1.59	0.87	0.46	0.40
	$E_{T2}$	1.52	1.77	1.74	1.98	1.86	1.51	1.44
	$E_T$	0.91	1.11	0.97	1.77	1.33	0.95	0.88
8 (4-1)	$E_{T1}$	3.64	9.07	2.14	5.21	18.17	3.45	3.46
	$E_{T2}$	8.17	2.11	4.21	4.97	3.07	7.62	7.40
	$E_T$	5.91	5.58	3.18	5.09	10.60	5.54	5.51
9 (4-2)	$E_{T1}$	0.27	4.7	0.58	1.06	3.04	0.25	0.24
	$E_{T2}$	1.98	0.48	1.67	1.70	1.45	1.95	1.52
	$E_T$	1.48	1.72	1.35	1.51	1.92	1.44	1.09
10 (5-1)	$E_{T1}$	0.59	0.73	0.72	0.80	1.42	0.65	0.58
	$E_{T2}$	4.44	4.88	9.94	0.87	17.25	4.78	4.23
	$E_T$	1.43	1.64	2.73	2.53	4.88	1.55	1.28
11 (5-2)	$E_{T1}$	3.09	3.06	1.26	3.87	5.59	1.64	1.52
	$E_{T2}$	10.08	13.76	18.88	12.32	14.86	21.97	12.83
	$E_T$	3.82	4.18	3.11	4.76	6.56	3.77	3.20
12 (5-3)	$E_{T1}$	1.18	7.15	0.67	4.08	6.78	1.38	1.35
	$E_{T2}$	31.97	10.51	38.44	23.97	23.90	26.17	20.67
	$E_T$	2.43	7.29	2.19	4.88	7.47	2.25	1.98
13 (5-4)	$E_{T1}$	2.51	3.44	1.76	2.06	4.90	2.23	2.16
	$E_{T2}$	2.05	2.79	2.51	2.81	3.52	2.31	2.26
	$E_T$	2.27	3.09	2.16	2.46	4.16	2.19	2.10
14 (6-1)	$E_{T1}$	0.51	1.70	0.20	0.42	1.54	0.61	0.55
	$E_{T2}$	10.70	4.98	22.39	16.75	24.54	8.12	8.01
	$E_T$	0.86	1.81	0.96	0.98	2.33	0.73	0.75
15 (7-1)	$E_{T1}$	0.99	0.35	0.78	0.38	0.96	1.04	0.99
	$E_{T2}$	6.84	8.98	15.88	15.20	25.42	6.43	6.56
	$E_T$	1.65	1.33	2.49	2.06	3.73	1.62	1.68
Mean	$E_{T1}$	2.13	4.49	1.64	2.63	5.28	1.93	1.24
	$E_{T2}$	4.47	5.63	9.97	7.60	10.70	7.98	5.16
	$E_T$	2.97	4.11	2.74	3.42	5.33	2.93	2.61



**Fig. 8.** The Qualitative Evaluation of the Proposed Network in Filtering ISPRS Data: Sample 1-1, (b) Sample 1-2, (c) Sample 2-1, (d) Sample 2-2, (e) Sample 2-3, (f) Sample 2-4, (g) Sample 3-1, (h) Sample 4-1, (i) Sample 4-2, (j) Sample 5-1, (k) Sample 5-2, (l) Sample 5-3, (m) Sample 5-4, (n) Sample 6-1, (o) Sample 7-1

**Table 3.** The Quantitative Evaluation of Procedures Generated Through the Modified PSO Algorithm and the Other Famous Interpolation Methods

	Bilinear (cm)	Cubic (cm)	Kriging (cm)	IDW (cm)	Spline (cm)	PSO (cm)	Modified PSO (cm)
Area 2	9.34	8.54	10.42	5.23	5.03	1.03	0.10
Area 3	8.89	8.23	10.21	4.88	4.23	0.94	0.09
Area 4	12.32	11.98	14.41	6.69	7.14	1.56	0.114
Area 5	14.43	13.56	15.98	7.56	9.65	3.24	0.156
Area 6	11.32	11.95	14.47	5.90	7.44	4.15	0.294



**Fig. 9.** The Qualitative Evaluation of the Proposed Procedure for Extracting the DTM from the DSM: (a) Area 2, (b) Area 3, (c) Area 4, (d) Area 5, (e) Area 6, (i) the final DTM, (ii) the Differential Map of the Extracted DTMs and the Reference DTMs

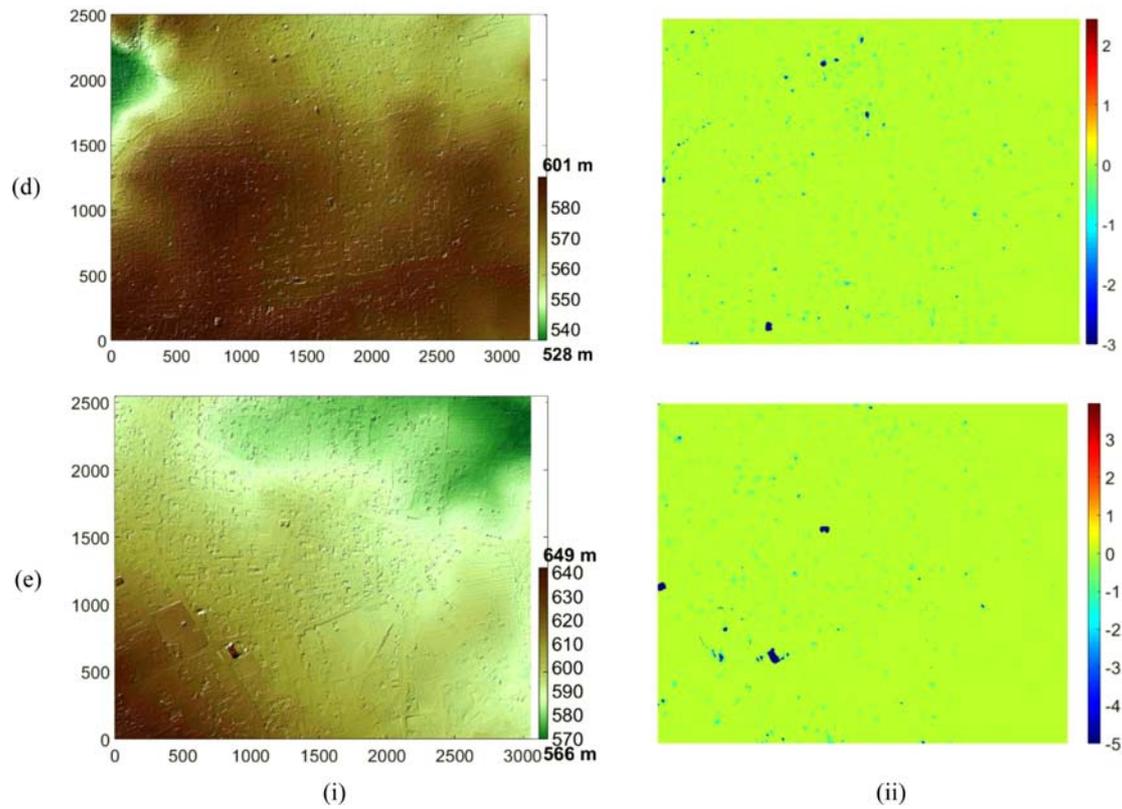


Fig. 9. (continued)

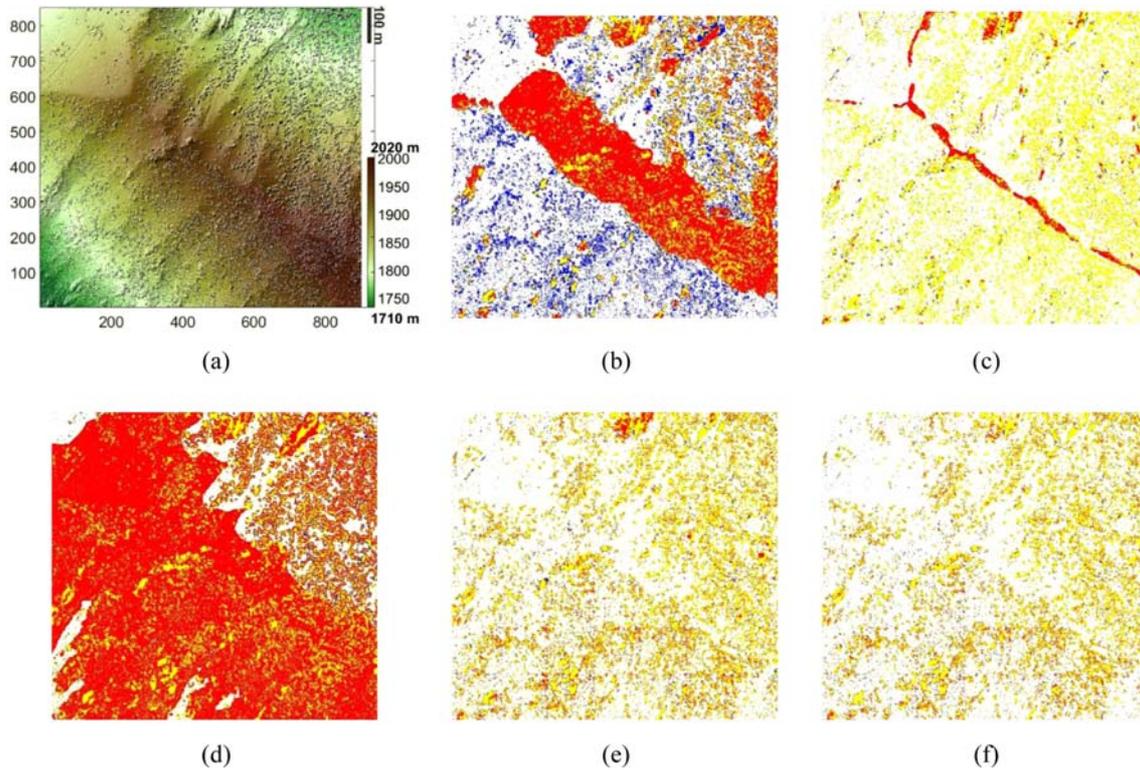
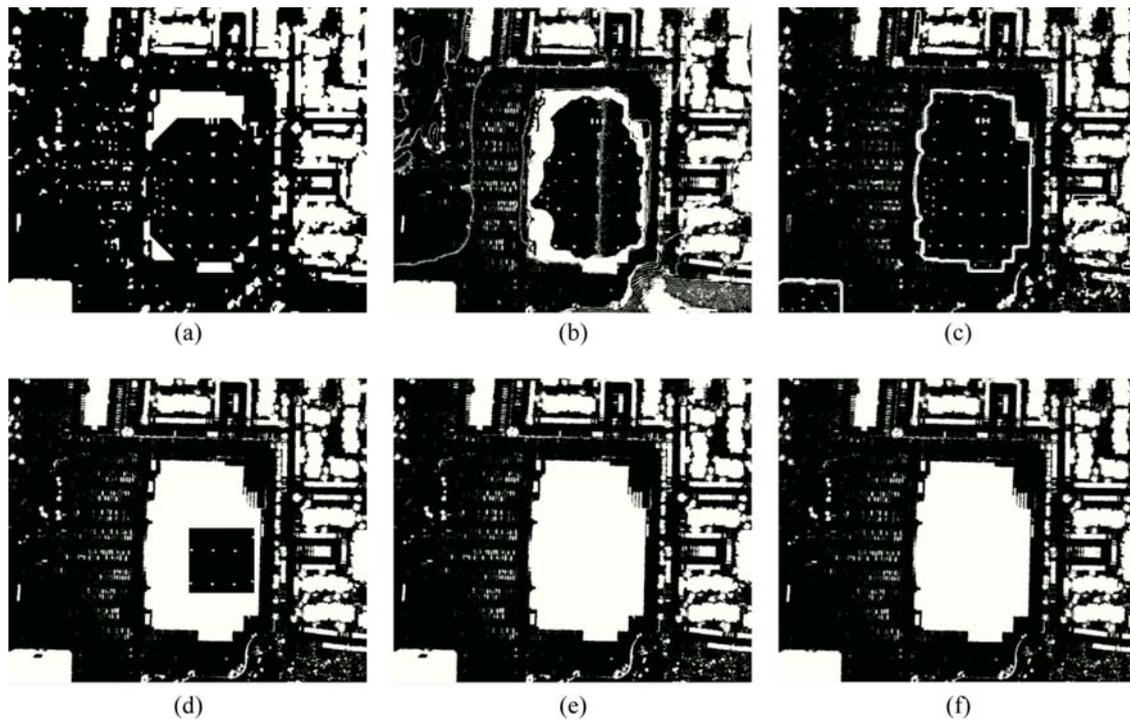


Fig. 10. Comparing Different Filtering Algorithms in Their Results of Dealing with Objects in Areas with Drastic Height Changes: (a) A Part of the DSM, (b) Results of the gLidar Algorithm, (c) Results of the LAsTools Algorithm, (d) Results of the CSF Algorithm, (e) Results of the Proposed Deep Encoder-Decoder Network, (f) Results of the Proposed Hierarchical Deep Networks.



**Fig. 11.** Comparing Different Filtering Algorithms in Dealing with High-dimensional Objects: (a) The gLidar Algorithm, (b) The LAStools Algorithm, (c) The CSF Algorithm, (d) the Proposed Deep Encoder-Decoder Network, (e) the Proposed Hierarchical Deep Network, (f) Reference Data

0.150 cm error rate, which is far better than those of the other interpolation methods.

## Acknowledgments

We would like to express our great thanks to the Deanship of Scientific Research, Al al-Bayt University for the financial supporting for conducting this research within the research project grant number, 4/2020/2021, in 30/5/2021.

## ORCID

Laith Abualigah  <https://orcid.org/0000-0002-2203-4549>

## References

- Antonarakis A, Richards KS, Brasington J (2008) Object-based land cover classification using airborne LiDAR. *Remote Sensing of Environment* 112:2988-2998
- Axelsson P (2000) DEM generation from laser scanner data using adaptive TIN models. *International Archives of Photogrammetry and Remote Sensing* 33:110-117
- Ballester P, Araujo RM (2016) On the performance of GoogLeNet and AlexNet applied to sketches, Thirtieth AAAI Conference on Artificial Intelligence
- Bigdeli B, Amini Amirkolae H, Pahlavani P (2018a) DTM generation from the point cloud using a progressive geodesic morphology and a modified Particle Swarm Optimization. *International Journal of Remote Sensing* 39:8450-8481
- Bigdeli B, Amirkolae HA, Pahlavani P (2018b) DTM extraction under forest canopy using LiDAR data and a modified invasive weed optimization algorithm. *Remote Sensing of Environment* 216:289-300
- Blaschke T (2010) Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing* 65:2-16
- Blaschke T, Tomljenović I (2012) LidarScapes and OBIA, Proceedings of the ASPRS. Annual Conference, Sacramento, CA, USA, pp. 19-23
- Chen C, Li Y, Li W, Dai H (2013) A multiresolution hierarchical classification algorithm for filtering airborne LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing* 82:1-9
- Chen H, Cheng M, Li J, Liu Y (2012) An iterative terrain recovery approach to automated DTM generation from airborne LiDAR point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 39:B4
- Chen Q, Wang H, Zhang H, Sun M, Liu X (2016a) A point cloud filtering approach to generating DTMs for steep mountainous areas and adjacent residential areas. *Remote Sensing* 8:71
- Chen Z, Gao B (2014) An object-based method for urban land cover classification using airborne lidar data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7:4243-4254
- Chen Z, Xu B, Gao B (2016b) An image-segmentation-based urban DTM generation method using airborne lidar data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9:496-506
- Elmqvist M (2002) Ground surface estimation from airborne laser scanner data using active shape models. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences* 34:114-118
- Gevaert C, Persello C, Nex F, Vosselman G (2018) A deep learning approach to DTM extraction from imagery using rule-based training

- labels. *ISPRS Journal of Photogrammetry and Remote Sensing* 142:106-123
- Guan H, Li J, Yu Y, Zhong L, Ji Z (2014) DEM generation from lidar data in wooded mountain areas by cross-section-plane analysis. *International Journal of Remote Sensing* 35:927-948
- Hong T, Huang X, Chen G, Yang Y, Chen L (2023) Exploring the spatiotemporal relationship between green infrastructure and urban heat island under multi-source remote sensing imagery: A case study of Fuzhou City. *CAAI Transactions on Intelligence Technology* 8(4):1337-1349
- Van Houtte C, Drouet S, Cotton F (2011) Analysis of the origins of  $\kappa$  (kappa) to compute hard rock to rock adjustment factors for GMPEs. *Bulletin of the Seismological Society of America* 101(6):2926-2941
- Hu X, Ye L, Pang S, Shan J (2015) Semi-global filtering of airborne LiDAR data for fast extraction of digital terrain models. *Remote Sensing* 7:10996-11015
- Hu X, Yuan Y (2016) Deep-learning-based classification for DTM extraction from ALS point cloud. *Remote Sensing* 8:730
- Huang X, Zhang L, Gong W (2011) Information fusion of aerial images and LIDAR data in urban areas: vector-stacking, re-classification and post-processing approaches. *International Journal of Remote Sensing* 32:69-84
- Hui Z, Hu Y, Yevenyo YZ, Yu X (2016) An improved morphological algorithm for filtering airborne LiDAR point cloud based on multi-level kriging interpolation. *Remote Sensing* 8:35
- Kobler A, Pfeifer N, Ogrinc P, Todorovski L, Oštir K, Džeroski S (2007) Repetitive interpolation: A robust algorithm for DTM generation from Aerial Laser Scanner Data in forested terrain. *Remote Sensing of Environment* 108:9-23
- Kraus K, Pfeifer N (2001) Advanced DTM generation from LIDAR data. *International Archives of Photogrammetry Remote Sensing And Spatial Information Sciences* 34:23-30
- Lee AC, Lucas RM (2007) A LiDAR-derived canopy density model for tree stem and crown mapping in Australian forests. *Remote Sensing of Environment* 111:493-518
- Li B, Shi Y, Qi Z, Chen Z (2018) A survey on semantic segmentation. In 2018 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 1233-1240). IEEE
- Li M, Zang S, Zhang B, Li S, Wu C (2014) A review of remote sensing image classification techniques: The role of spatio-contextual information. *European Journal of Remote Sensing* 47(1):389-411
- Li Y, Yong B, Wu H, An R, Xu H (2014) An improved top-hat filter with sloped brim for extracting ground points from airborne lidar point clouds. *Remote Sensing* 6:12885-12908
- Lu WL, Murphy KP, Little JJ, Sheffer A, Fu H (2009) A hybrid conditional random field for estimating the underlying ground surface from airborne lidar data. *IEEE Transactions on Geoscience and Remote Sensing* 47:2913-2922
- Maguya AS, Junttila V, Kauranne T (2013) Adaptive algorithm for large scale dtm interpolation from lidar data for forestry applications in steep forested terrain. *ISPRS Journal of Photogrammetry and Remote Sensing* 85:74-83
- Meng X, Wang L, Silván-Cárdenas JL, Currit N (2009) A multi-directional ground filtering algorithm for airborne LIDAR. *ISPRS Journal of Photogrammetry and Remote Sensing* 64:117-124
- Mokayed H, Quan TZ, Alkhaled L, Sivakumar V (2023) Real-time human detection and counting system using deep learning computer vision techniques. *In Artificial Intelligence and Applications* 1(4): 221-229
- Mongus D, Lukač N, Žalik B (2014) Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces. *ISPRS Journal of Photogrammetry and Remote Sensing* 93:145-156
- Mongus D, Žalik B (2012) Parameter-free ground filtering of LiDAR data for automatic DTM generation. *ISPRS Journal of Photogrammetry and Remote Sensing* 67:1-12
- Mongus D, Žalik B (2013) Computationally efficient method for the generation of a digital terrain model from airborne LiDAR data using connected operators. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7:340-351
- Mukherji A, Mondal A, Banerjee R, Mallik S (2022) Recent landscape of deep learning intervention and consecutive clustering on biomedical diagnosis. In *Artificial Intelligence and Applications*
- Pfeifer N, Stadler P, Briese C (2001) Derivation of digital terrain models in the SCOP++ environment, Proceedings of OEEPE Workshop on Airborne Laserscanning and Interferometric SAR for Detailed Digital Terrain Models, Stockholm, Sweden
- Pingel TJ, Clarke KC, McBride WA (2013) An improved simple morphological filter for the terrain classification of airborne LIDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing* 77:21-30
- Roggero M (2001) Dense DTM from laser scanner data, Proc. OEEPE Workshop Airborne Laser Scanning Interferometric SAR. Stockholm.
- Samadzadegan F, Bigdeli B, Ramzi P (2010) A multiple classifier system for classification of LIDAR remote sensing data using multi-class SVM, International Workshop on Multiple Classifier Systems. Springer, 254-263
- Sampath A, Shan J (2004) Urban modeling based on segmentation and regularization of airborne LIDAR point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 35(B3):937-942
- Shan J, Aparajithan S (2005) Urban DEM generation from raw LiDAR data. *Photogrammetric Engineering & Remote Sensing* 71:217-226
- Shao YC, Chen LC (2008) Automated searching of ground points from airborne lidar data using a climbing and sliding method. *Photogrammetric Engineering & Remote Sensing* 74:625-635
- Shi Y (2004) Particle swarm optimization. *IEEE Connections* 2(1):8-13
- Simon K, Vicent M, Addah K, Bamutura D, Atwiine B, Nanjebe D, Mukama AO (2023) Comparison of deep learning techniques in detection of sickle cell disease. *In Artificial Intelligence and Applications* 1(4):252-259
- Sithole G (2002) Filtering strategy: Working towards reliability. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences* 34(3/A):330-335
- Sohn G, Dowman I (2002) Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion, International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences. Natural Resources Canada, 336-344
- Vosselman G (2000) Slope based filtering of laser altimetry data. *International Archives of Photogrammetry and Remote Sensing* 33:935-942
- Wang CK, Tseng YH (2014) Dual-directional profile filter for digital terrain model generation from airborne laser scanning data. *Journal of Applied Remote Sensing* 8:083619
- Wang YQ, Li JY, Chen CH, Zhang J, Zhan ZH (2023) Scale adaptive fitness evaluation-based particle swarm optimisation for hyperparameter and architecture optimisation in neural networks and deep learning. *CAAI Transactions on Intelligence Technology* 8(3):849-862
- Zhang J, Lin X (2013) Filtering airborne LiDAR data by embedding smoothness-constrained segmentation in progressive TIN densification. *ISPRS Journal of Photogrammetry and Remote Sensing* 81:44-59
- Zhang W, Qi J, Wan P, Wang H, Xie D, Wang X, Yan G (2016) An easy-

- to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sensing* 8:501
- Zhao X, Guo Q, Su Y, Xue B (2016) Improved progressive TIN densification filtering algorithm for airborne LiDAR data in forested areas. *ISPRS Journal of Photogrammetry and Remote Sensing* 117:79-91
- Zhao Z, Sun B (2023) Hyperspectral anomaly detection via memory-augmented autoencoders. *CAAI CAAI Transactions on Intelligence Technology* 8(4):1274-1287, <https://doi.org/10.1049/cit2.12116>
- Zhu L, Suomalainen J, Liu J, Hyypä J, Kaartinen H, Haggren H (2018) A review: Remote sensing sensors. *Multi-purposeful Application of Geospatial Data*, 19-42