Assiut University Journal of Multidisciplinary Scientific Research (AUNJMSR)
Faculty of Science, Assiut University, Assiut, Egypt.
Printed ISSN 2812-5029
Online ISSN 2812-5037
Vol. 51(1): 21–39 (2022)
https://aunj.journals.ekb.eg/

Deep learning approach for catchment detection in Asyut –Egypt

Hanaa A. Sayed[1,2 *], Ahmed M. Sefelnasr[3, 4], and Essam KH. Abd-Elmageed[2 *]
[1] Dept. of Computer Science, College of Computer Science and Engineering Taibah University, KSA
[2] Dept. of Mathematics, Faculty of Science, Assiut University, 71516 Assiut, Egypt,
[3] National Water and Energy Center, United Arab Emirates University, P.O. BOX 15551 Al Ain, UAE
[4] Dept. of Geology, Faculty of Science, Assiut University, 71516 Assiut, Egypt

Corresponding authors: Hanaa A. Sayed (E-mail: hasali@taibahu.edu.sa);
Essam KH. Abd-Elmageed (E-mail: essamkhalifa61@gmail.com)

ARTICLE INFO

Article History:
Received: 8/11/2021
Accepted: 14/12/2021
Online: 14/2/2022

Keywords:
Elevation Model (DEM), deep learning, DenseNet, uncertainties, catchments, Geographic Information System(GIS).

ABSTRACT

Water resources assessment is an essential element in the sustainable development and management of water resources. It provides a basis for many applications, such as maintenance of projects associated with irrigation and drainage. Catchment detection and identification is one of the water resources assessment fields, especially in dry areas. Few studies have attempted to detect catchments based on DEM, such as the level-set method based on graph theory. In this work, a deep learning algorithm (DenseNet) was used to detect and locate catchments. Identifying Sink Features in the DEM is the first step. Then, using the level-set process to delineated topographic depressions in DEMs. Finally, Catchments are detected using DenseNet. As the DEM accuracy increase by removing uncertainty from DEM the catchment detection performance increase. Asyut Governorate, Egypt, is used as a study area.

INTRODUCTION

A Digital Elevation Model (DEM) is a digital raster image representation of an earth's surface created from elevation data, where regular grid represents elevation values in the earth's surface. There are many types of elevation models. The shape of the ground surface is represented by the Digital Terrain Model (DTM). The shape of the surface is represented by a digital surface model (DSM), such as buildings, vegetation, hills, etc. [1]. Topography is an essential factor in most types of hydrology analysis. It can be derived from various DEMs through many types of techniques, for example, digitizing contours from available topographic maps, topographic leveling, many types of GPS measurements, and Light Detection and Ranging (LIDAR). Numerous maps may be produced from DEMs using GIS operations. More data sources can be used for the

generation of DEMs. Selection is dependent on data availability for a field area, cost, and application. Derived DEMs can be used in a heuristic process or method. [2] small scale (e.g., background hill-shading images, physiographical classification, internal terrain, drainage density for statistical analysis at regional scales (e.g., elevation zones, slope, slope direction) and physical modeling at the local scale (local drain direction, slope gradient, flow path). The resolution of DEM is significant when using slope gradient maps in hazard assessment. In general, slope gradient maps should not be used for small-scale studies [3]. DEMs are used in hydrological modeling of slopes, and slope maps are used in physical modeling. DEMs' importance becomes increasingly because of the fundamental nature of the data and knowledge of the data they represent [4]. The DEM serves as the basis for modeling and analyzing spatial topographical information. Accuracy and acquisition techniques are developed and become available for researchers. Many DEMs technologies were used, including landform evolution, the modeling of soil erosion, and other geo-simulations. DEMs provide critical data for water resource management, such as detecting catchment areas and assessing flood risk [5].

All DEMs contain particular uncertainty or DEM errors (the difference of a given elevation relative to the ground truth or reality). DEM errors are frequent blunders or random errors. Systematic errors produced from making or generation DEM. Systematic errors can be removed or minimized. Vertical errors associated with the data collection process are typically identified and removed before data release [6]. The sources of errors are summarized as (a) incomplete density of observations may Couse errors in data sampling. (b) an error like numerical errors in the computer, interpolation errors, or classification and processing errors and (c) measurement errors like position inaccuracy (in both x and y directions) [7]. These observer bias errors lead to uncertainty in DEM, which reduces the accuracy of application results. Providing elevation measurements of an earth surface (ground truth) manually compared with DEM values to detect and remove uncertainty and improve the DEM accuracy [8].

A catchment is a low point in a specific area that makes creek, lake, and rivers by water flows. The size and depth of these catchments area depend on the volume, direction, and speed of water flows and shape of the catchment's land. Catchments vary significantly in size, and the amount of water they can hold is determined mainly by the size of the catchment and the amount of rain that falls on it. Rivers, bush land, fields, lakes, houses, streets, trees, pets, and people are all part of the natural and human processes that make up catchments [9].

Deep Neural Networks provide a better level of information reflection, potentially resulting in improved predictive control, generalization, and transferability [9-11]. CNN is a neural network that uses multiple building blocks to learn spatial hierarchies of features in an automated and adaptive manner, such as convolution layers, pooling layers, and fully connected layers. CNN's with a more complex layer structure are more capable of learning [10]. The majority of current high-resolution DEM reconstruction methods were created for natural terrains, divided into three types. i.e., DEM interpolation [11], DEM enhancement [12] and learning-based DEM reconstruction [13]. Another CNN application was used to estimate a topographic reconstruction to estimate a DEM using a

single complex image [14].application was used to estimate a topographic reconstruction to estimate a DEM using a single complex image [14].

RELATED WORK

Many methods used DEM to determine catchments areas in many types of earth surfaces like landscape, mountains, hills, and urban. The algorithms used to determine catchments are priority flood and level set.

The priority-flood algorithm was used to find and fill sinks in the DEM. The depression-less DEM was then subtracted from the initial DEM [15]. Grid cells with a difference in elevation of 0 are non-depression cells removed from further analysis. The binary image was generated from the elevation difference grid. Cells that contain value 1 denote depression cells. (Elevation change > 0) and value of 0 denotes non-depression cells. The depression region's attributes were computed using the region-group algorithm. The root mean square error (RMSE) of nonzero depressions was used as a threshold to illuminate small and shallow depressions. Finally, the refined binary frame was used to cut the original DEM using a mask. [16]. to detect the nested hierarchy of depressions, the population of depressions was passed to the level algorithm. The filling algorithm was used to form synthetic river pathways to drain surface water to the river system. The main limitation of this algorithm is that it only fills depressions dependent on flow accumulation and must be calculated before used input data.

The level set used to detect catchments. By tracing the dynamic topological changes, the level-set method was used to construct topological graphs and derive geometric properties of the nested depressions. They used conventional depression treatment, particularly when evidence on the nested hierarchical structure of depression is available. [18]. However, it has some disadvantages. Assume there is the first collection of points defining the changing boundary. That points are inserted into the border as it narrows or expands. The distance between the boundary points is small enough to allow smooth evolution. This process may become a difficult task during the implementation of the algorithm, especially when there's a topological change because of the nested hierarchy's characterization. The small depressions were not detectable due to pixel resolution limitations.

In the Delineation of Potential Catchments on the lidar-derived DEM, they used the localized contour approach to find wetland depressions. The wetland depression relates to the depression's full scope. Wetlands and catchments are linked and should be regarded as interconnected hydrological units, with the average height of wetland depressions being 2.6×103 m2. For each of the 33×241 wetland depressions, the associated wetland catchments were identified [19]. Wetland catchments were roughly 10 times larger than wetland depressions, with a median size of 26 × 103 m2. For each wetland depression, the proportion of depression area to catchment area was calculated. The proportions were found to range from 0.04 to 83.72%, with a median of 14.31%. The average wetland to catchment area was estimated. Because of the much larger sample size, the results were calculated from over 30000 wetland depressions and catchments, providing a statistically accurate result for the study region. This method has the limitation that it focuses on more

significant wetland depression greater than 500 m2 but ignores eliminated areas smaller than 500 m2.

MATERIALS AND METHODS

The automated method of inferring nested depressions consists of several steps: first, DEM filtering, in which image preprocessing tools are used to enhance the DEM data. Only depression pixels from a subset of the smoothed DEM are then obtained using the depression filling algorithm (i.e., priority flood). Second, the priority flood algorithm is used to classify depressions. And finally, characterizing catchment area hierarchical structure and using the DEM subset in the computation of geometric properties.

1. HYDROLOGICAL CORRECTION

Hydrologic modeling with Geographic Information Systems (GIS) technology starts with good elevation data. United States Geological Survey (USGS) provides DEM 30-meter. DEMs do not accurately reflect the surfaces of low-lying, extremely flat areas containing multiple ponds and wetlands. Data from hypsography was used to create this component.

In ArcGIS, the conditioning step of hydrologic modeling necessitates the development of three general utility datasets. A DEM with sinks filled, a grid showing the flow path for each cell, and a flow accumulation dataset in which each cell receives a value equal to the total number of cells that drain into it are generated in that order. Depressions or sinks in a DEM were used for flow routing and complied with before creating a flow path or accumulation grids. Many sinks are not actual sinks and have been removed as a result of DEM creation errors. Some sinks require regular maintenance. A prerequisite is the development of a threshold grid or sinks mask [20]. Identifying wetland features in a DEM, which in turn will be used to construct a realistic sink mask. This mask will subsequently be used with a set of ArcInfo GRID commands to fill a DEM selectively. The resultant DEM is employed in generating flow direction, flow accumulation, and drainages line grids to create the correct hydrological DEM.

2. IDENTIFYING SINKS IN DEM

Initially, ArcInfo was used to clip the initial DEM which shown in figure 1, filled DEM, flow path grid, and flow accumulation grids, and then a mask tool was used to extract. The original DEM grid is subtracted from the filled depression-less version to compute the difference grid as equation (1). Then, the cell values in the sink mask that are greater than 0 were coded to 1 as in equation (2), the Region Group algorithm was applied to the sink mask to create unique sink regions. The region group algorithm is used to connect cells, as shown in figure 2.

$$difference\ Grid = filled\ DEM - Original\ DEM \quad (1)$$

$$\text{sink mask} = \begin{cases} 1, \text{difference} > 0 \\ 0, \text{other wise} \end{cases} \quad (2)$$

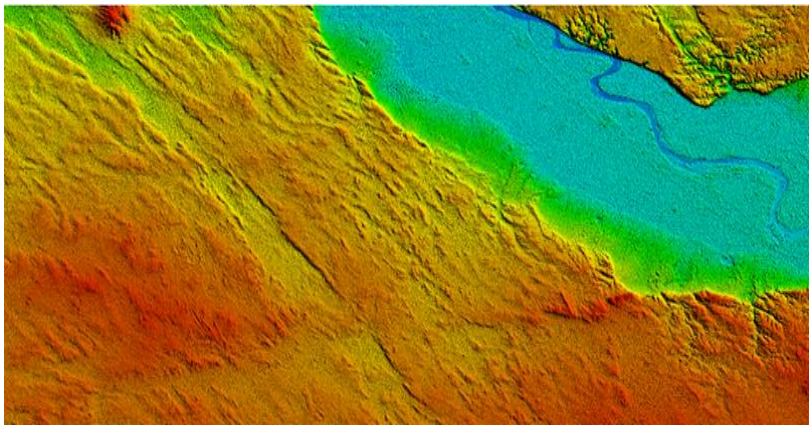


Figure 1: initial DEM that used under spatial operations

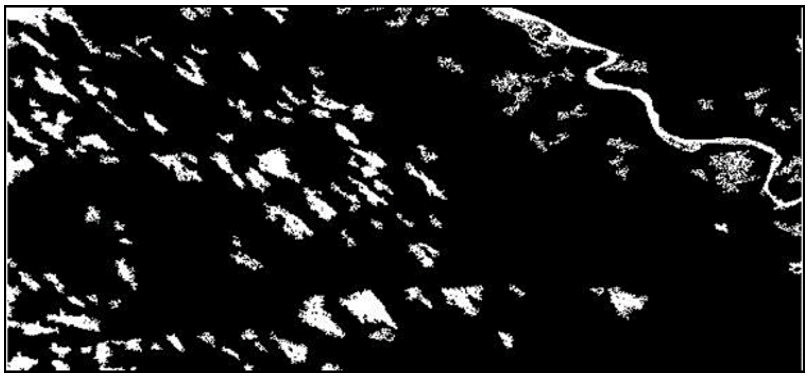


Figure 2: DEM sinks mask

3. DELINEATING TOPOGRAPHIC DEPRESSIONS

The sink mask and DEM are passed to the level set algorithm to detect the catchments regions. The level-set approach will be used to delineate topographic depressions in DEMs; we define the level-set function φ to have the following properties in equation (3).

$$\emptyset(z,h) = 0 \; for \; E(Z) > h$$
$$\emptyset(z,h) = 1 \; for \; E(z) \leq h \qquad (3)$$

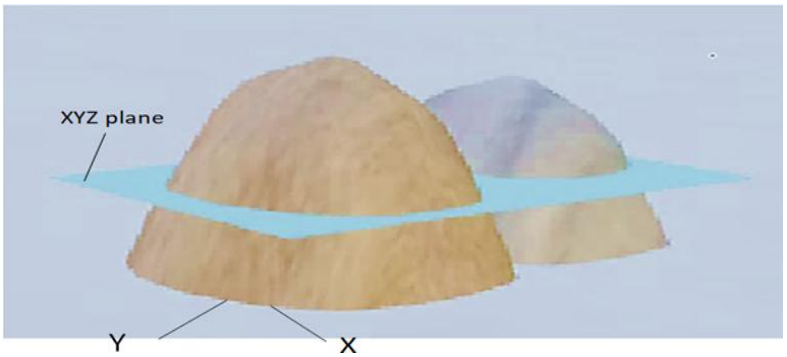


Figure 3. The level-set approach for numerical surface analysis

The point (x, y) on the surface is represented by z, The surface elevation of the point is E(z), and the elevation of the xy plane intersecting the surface is h [18], as shown in figure 3. Then the level-set function converts a surface into a binary image when these two properties are used. By intersecting the surface with the xy plane, points above the xy plane are numbered as 0, where surface points below the xy plane are labeled as 1. Based on the xy plane's elevation, which can vary between the depression's lowest and highest points, the resulting binary image identifies the inundated area as shown in figure 5 which declares the level set catchments area in three levels. These are output levels of the actual depression area.

4. CATCHMENT DETECTION

DenseNet (Densely Connected Convolutional Networks) is one of the most robust neural networks models for visual object recognition. An output of the previous layer acts as an input of the second layer by using composite function operation. This composite operation consists of the convolution layer, pooling layer, batch normalization, and non-linear activation layer. Both subsequent layers use the attribute maps from the previous layers as inputs, as shown in figure 4. [23], These connections mean that the network has L (L+1)/2 direct connections. L is the number of layers in the architecture. DenseNets are divided into DenseBlocks, where the dimensions of the feature maps remains constant within a block, but the number of filters changes between them. These layers between them are called Transition Layers and take care of the downsampling applying a batch normalization, a 1x1 convolution and a 2x2 pooling layers. After each layer block, apply regularizes penalties by 5 on layer parameters or layer activity during optimization to prevent network overfitting.

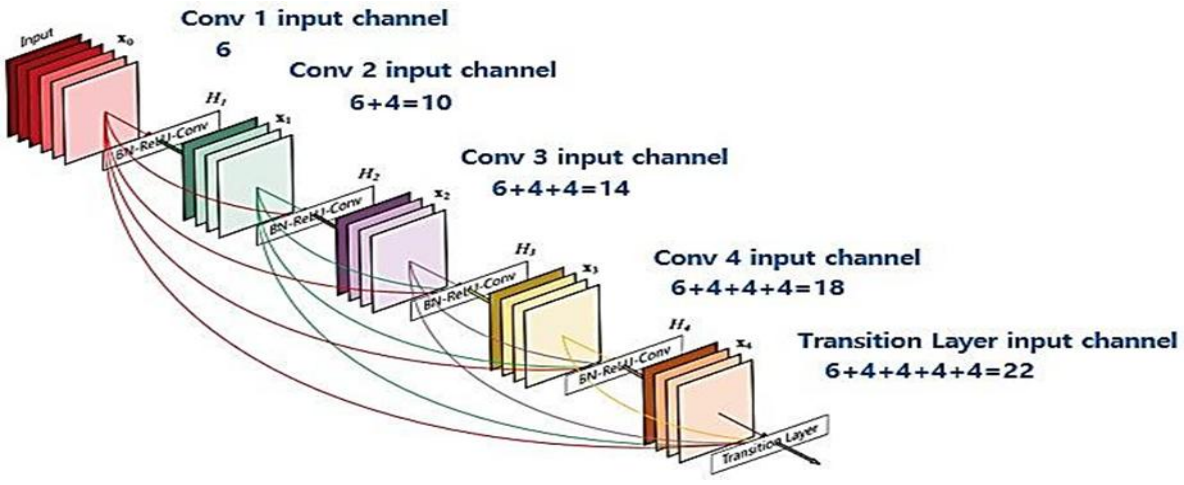


Figure 4: A 5-layer dense block with a growth rate of k = 4,

Rectified Linear Unit (ReLU) and Batch Normalization applied after each Dense Block. The number of filters changes between the DenseBlocks, increasing the dimensions of the channel. The growth rate (k) helps in generalizing the L^{th} layer. It controls the amount of information to be added to each layer. We used the DenseNet deep learning model to train spatial data [22]. The Dense Block is a critical component of the DenseNet because it enhances information flow between layers. It is composed of BN, ReLU, and 3 × 3 Conv. The specific formula is shown as follows, expressed in equation (4):

$$X_{L=}H_L([X_0,X_1,\dots,X_{L-1}]) \qquad (4)$$

Where$[X_{0,}X_1,\dots,X_{L-1}\ ]$denotes the concatenation of the feature-maps generated in layers 0,1,….., L-1and $H_L()$ is defined as a composite function three consecutive operations on the input of the L^{th}layer is defined as a composite function of three consecutive operations on the input of the l^{th} layer.Began with concentric patches for training generated by normalizing data. Used Dimensions of the outer patches 64x64 and the inner patches 32x32. Using Adam optimization, we trained our architecture for 45 epochs with a batch size of 64. The F1 score, which is a measurement of a test's accuracy, is used to assess our classification accuracy. It is measured using the test's precision and recall. Precision is defined as the number of correctly identified positive results divided by the total number of positive results, including those not identified correctly. The recall is calculated by dividing the number of correctly identified positive findings by the total number of samples that should have been positive.

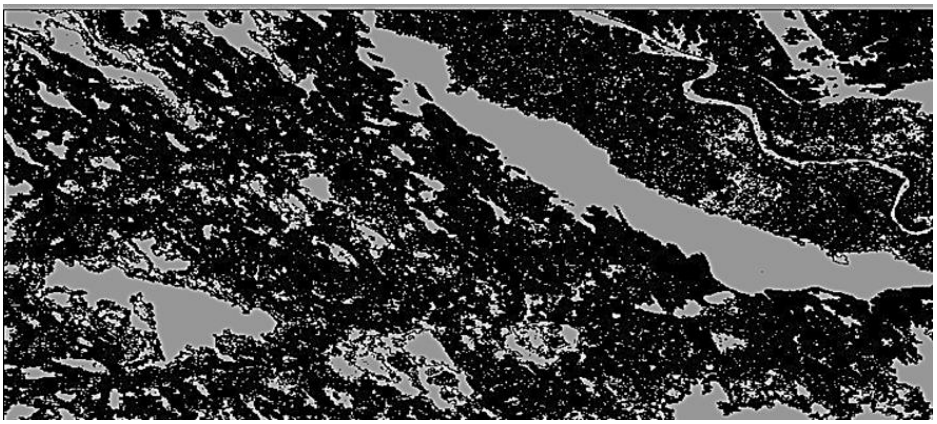


Figure 5: Polygon features output from the level set algorithm

DenseNet uses a combination of nonlinear transformations with high complexity from higher layers and transformations with low complexity from the shallow layer in its deep concatenations. Thus, it tends to get a smooth decision function with better generalization performance. DenseNet will also deepen the network, thus reducing overfitting. DenseNet uses concatenation to join features from different layers, so features must be the same size. Down-sampling, on the other hand, is critical in a CNN because it increases the receptive

field. A direct result output that is the same size as the input data is preferred, with each pixel on the original input image reflecting the group of the pixel at the exact location. We utilized feature polygons shapefile to validate our method and used corrected DEM downloaded from USGS [24] with size 2568 x1428 pixels. For training and testing, we manually labeled 70% of the pixels as ground truth. Each set of data needed normalization and patching from the normalized data. The polygon feature was converted to pixels masks and counted to classify into two pixels that followed normal elevation area and other followed catchments area. All calculations were completed using machine that have microprocessor Intel core i 7 , Windows operating system , NVidia GeForce GTX Titan X 12GB GPU with 12 G.B of memory and 16 GB of RAM. The sinks mask and the DEM used as input data to the level set algorithm. Figure 5 shows a polygon Shapefile in DEM that output from the level set in three levels: high, medium, and low, representing the elevations from 551 to 589, from 286 to 550, and 50 to 285, respectively. Labeled shapefile and DEM elevation data (training DEM) are input to the DenseNet model for training and validation.

5. STUDY AREA AND EXPERIMENTS

The study area is Asyut Governorate is located between the latitudes of 26° 50′ and 27° 40′ north, and the longitudes of 30° 40′ and 31° 32′ 13.5″ east, and it is bordered on both sides by the Nile River as shown in figure 6 . Asyut Governorate is located in Egypt's arid belt, characterized by long, hot summers, cold winters, low rainfall, and high evaporation rates. According to Asyut Meteorological Station [27].SRTM elevations data sets were downloaded from the United States Geological Surveys1 (USGS) Earth Explorer. DEM is provided 1° x 1° tiles at 1 arc-second (30m) resolution and is on the WGS84 datum. Also is referenced to mean sea level realized by the EGM 96 geoid model. Hence the heights are orthometric with vertical units in meters for calibration purposes.

In this work, two experiments are implemented in the study area. The first experiment used a shapefile containing a catchment area feature and corrected DEM as input data to DenseNet. While in the second experiment, the DEM uncertainties were detected to improve the DEM accuracy. The shapefile and corrected DEM after detected uncertainties parts are input to DenseNet. In the first experiment(catchment detection in raw DEM CDR-DEM), the raw DEM in Asyut study Area and sinks mask are used simultaneously as input data to the level set algorithm. The parameters use in this algorithm are minimum depression size = 50 m2 (the minimum catchment area detected) and depression depth = 1 m (the minimum depth of catchment recognized). The depression of each pixel calculated from which has the same depression grouped in the same region.

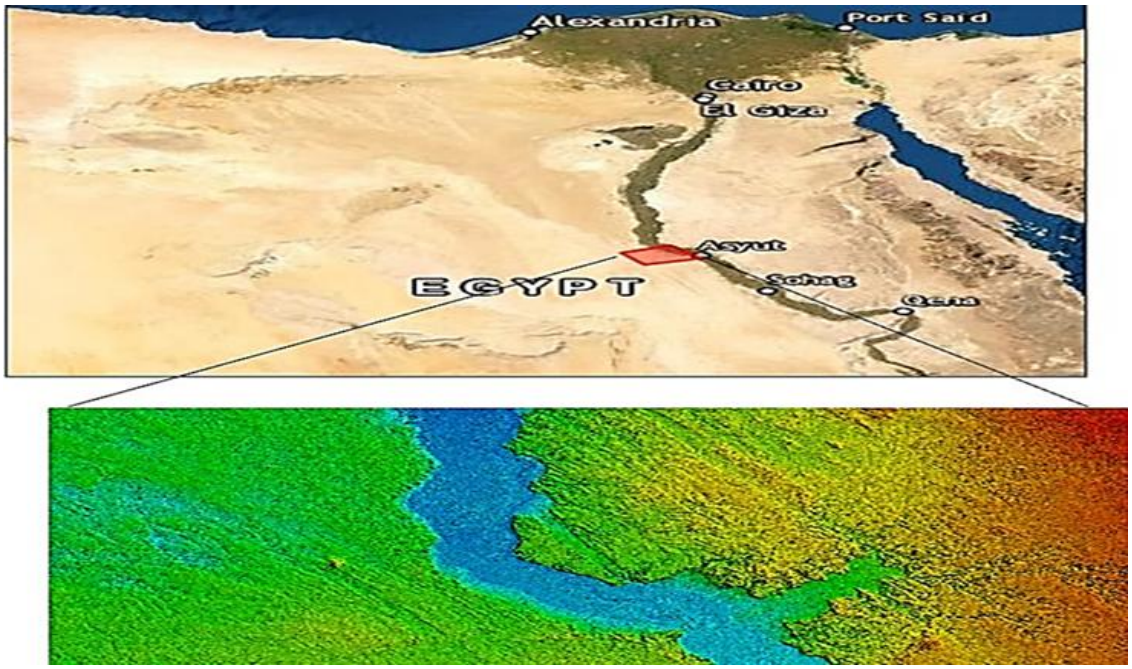


Figure 6. Location and Digital elevation model of the study area in Asyut

. The output of the level set algorithm is the feature of catchment formed in a polygon shapefile. This shapefile and DEM are used as input to DenseNet. The DenseNet detects catchments area (actual sinks or depression) in the DEM image. The catchments are detected in DEMs with an accuracy of 90.6%. Figure 7 and Figure 8 Shows the test and classified image for two experiments. In the second experiment (Catchment detection in Error Free DEM CDEF-DEM), due to the DEM low resolution (each pixel represents 30m x 30m of the surface), the DEM uncertainties were detected and removed. First, the DEM uncertainties were removed [25]. Accuracy descriptive statistics of elevation values were used to compare points between the elevation of raw data in raster image (H_{SRTM}) and elevation of GPS or ground truth points H_{GPS}. For the study area, the vertical differences between HSRTM and H_{GPS} were computed as $\Delta H_{(SRTM-GPS)}$ is shown in Table (1). The minimum elevation for the GPS is 266m while the maximum is 317m, and Root Mean Square Error is calculated using (5). Comparing SRTM DEM and reference GPS points shows that most SRTM point's elevation is lower than GPS elevation [25]. The points' elevation differences minimum and maximum values are -98m and 156m, respectively, as shown in Table 1:

$$RMSE_Z = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Z_{di} - Z_{ri})^2} \qquad (5)$$

Where Z_{di} represents the ith elevation value determined on the DEM surface, Z_{ri} represents the original elevation, and n represents the number of elevation points tested.

Table 1.2D elevation differences of first study area

Elevation statistics	$\Delta H_{(SRTM-GPS)}$
Min	-98
Max	156
Mean	-8.9748
RMSE	22.29

First, the 565 GPS survey points were overlaid on the DEMs in ArcGIS10.3 and used extracted values of DEMs at points located with the GPS data [26]. The SRTM and GPS data added in ArcGIS using add x, y data tool to compute the difference values in a 2-dimension raster image by performing Inverse Distance Weighted (IDW) interpolation from geostatistical analyst tools. 2-Dimension raster view was used to create uncertainties features by global mapper GIS software and save them in shapefile [26]. The shapefile was used as ground truth data that passed to the DenseNet model as validation data. 2-dimension raster image that contains elevation differences as training data. The system labels the uncertainties region in the study area. The below algorithm shows the steps used to detect the catchment region.

CDER-DEM Algorithm

1. By GIS software create polygon uncertainties feature and export it as shapefile.
2. Create difference image
   - 2.1 $diff\_points = (H_{GPS} - H_{DEM})$
   - 2.2 $Raster_{img} = IDWinterpolation(diff\_points)$
3. Raster image and uncertainty shapefiles are used for training the DenseNet to label the uncertainty regions.
4. The labeled uncertainty region image and sinks image are used by level set algorithm to compute shape file.
5. The shape file and DEM data were input to DenseNet model to label the catchment areas.

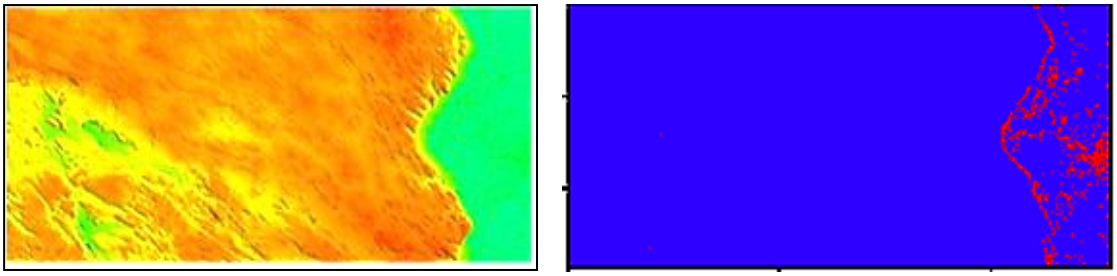
RESULTS

For both experiments, the DenseNet learning rate is 0.0001 For training the model for area bounded by longitudes =30° 26' 02.1588" E and 31° 08' 49.1588" E and latitudes =27° 25' 14.2078" N and 27° 01' 27.2078" N. the minimum and maximum elevation are 29 m and 285 m respectively, for testing the area between longitude =30° 05' 31.2528" E and 30° 48' 18.2528" E and between latitudes =27° 50' 37.1931" N and 27° 50' 37.1931" N. the minimum and maximum elevation are 22 m and 170 m. both regions captured by Projection DATUM WGS84. The study area contains varied surface elevations like medium mountains, hills, landscape, and water. Two kernels are used, 3 x 3 and 5 x 5, in the DenseNet model [25]. The network that uses 5 x 5 kernel filters achieved a better result, as shown in Table 2.

Table 2. Experiments accuracy results

CDR-DEM				
Kernel filter size	**precision**	**recall**	**f1-score**	**Accuracy of the model%**
3×3	0.90	0.97	0.93	89.189
5×5	0.91	0.98	0.94	90.585
CDER-DEM				
3×3	0.92	0.99	0.95	91.096
5×5	0.93	0.99	0.95	92.652

We used Python, a cross-platform programming language, to implement the level set algorithm. Several scientific libraries are used, such as SciPy, scikit-image, and matplotlib. And Gdal Geospatial Data library that Reading and writing DEM files makes it possible to integrate powerful algorithms for processing geospatial data (such as DEM) [18].



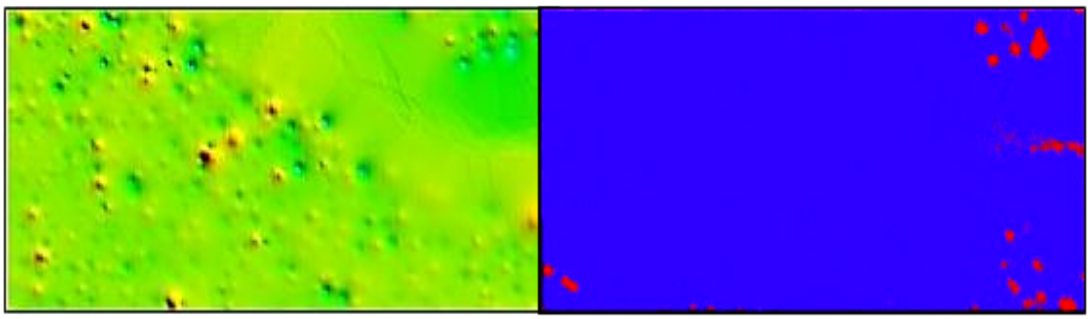
a-Test image b-Classified image

Figure 7. (a) tested image b) classified image of CDR-DEM method

The input image of elevation differences that passed to the DenseNet model and the output labeled image. Table 3 shows the output results of catchments areas detection in CDR-DEM and CDER-DEM. Catchment detection is on of many hydrology applications such as flash flood forecasting , groundwater localization, Calculates rainfall, surface runoff, etc. The accuracy of the DenseNet model depended on the spatial resolution of the digital elevation model and the accuracy of catchments shape files feature. The results show that catchments regions were located in slopes Areas. The overall classification accuracies are 90.6% for CDR-DEM and 92.7% for CDER-DEM, as shown in table 3:

Table 3.Statistics Experiments results

Model	CDR-DEM	CDER-DEM
Number of catchments	30	30
Maximum depth	153	150
Minimum depth	2.9	2
Mean depth	28.9	27
Loss	0.2501	0.2232
Accuracy	90.6%	92.7%



a- Differences elevation image b- Classified image

Figure 8. (a) Differences elevation image,(b) classified image of CDER-DEM

Many loss functions are used in deep learning like (Regression Loss Function, Mean Squared Error, Mean Squared Logarithmic Error Loss, Mean Absolute Error Loss. Binary Classification Loss Function, Binary Cross-Entropy Loss, and Hinge Loss). In this work cross-entropy was used [28] to improve the classification performance for each batch predicted values compared with the actual labels. The cross-entropy loss function computes the probability of misclassified pixels. In binary classification, the pixel probability is usually predicted by only one output. The binary cross-entropy loss function is calculated from (6).

$$Loss = -[y\log(p) + (1-y)\log(1-p)] \qquad (6)$$

Where log is the natural logarithm, y is a binary indicator (0 or 1), and p is prediction probability

Time complexity

The time-complexity of the CNN classification training process is influenced by three primary factors [29]: (1) the configuration of image feature extraction through CNN model. Including the convolution, the max-pooling and the drop-outs; (2) the depths of neural network layers (3) the number of iterations (epochs) to optimize the model; and (4) the model optimizer [30]. The time complexity for The core component of DenseNet model is:

$O(\sum_{l=1}^{d} n_{l-1} \times s_l^2 \times n_l \times m_l^2)$,where d is the number of convolution layer, n_l the number of filters, s_l is the size of filters, n_{l-1} is the number of input channels of the l layer, and m_l is the size of the output feature map.

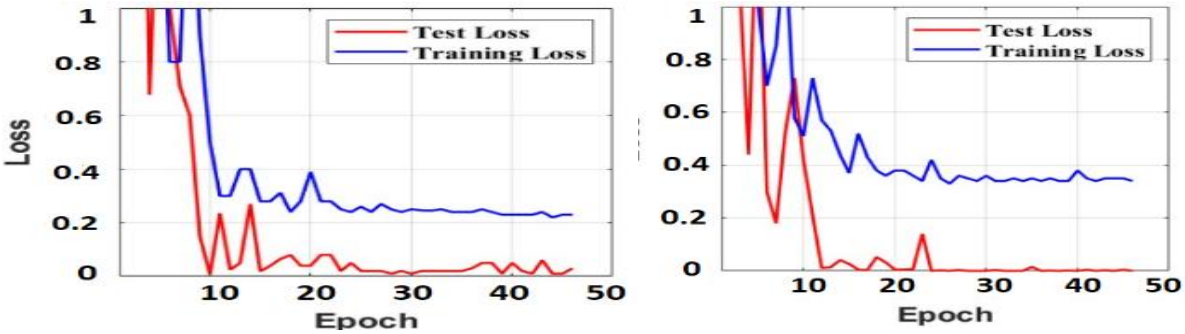
DISCUSSION

Deep Convolutional Neural Network is one of the most popular and effective approaches in the remote sensing and special data domain, especially for classification problems. The capability of DenseNet models to learn several levels of abstraction in a hierarchical fashion directly from images differentiate them from traditional machine learning approaches. GIS software like Arc map and Global mapper play an important role in this work which used to detect watershed, flow direction and used to sinks extraction from DEM. catchments area detection are very important field in water resource. In this work we used integration between power full an effective algorithm which used in identifying catchments regions depending on GIS software analysis provide information about actual sinks and actual water flow in digital elevation model. Accuracy of DEM is another important point must keep in mind in this work. The accuracy effect directly in DenseNet Performance. Shape files formed in various points, lines and polygons We use shape files in form in polygons that correspond features of catchments regions and Use corresponding digital elevation model to train our model. In first experiment(CDR-DEM) train DenseNet model with two kernel filter size and 45 epochs we observed that the accuracy increased from ~89.2 % to ~90.6% when used 5×5 filter and the loss decreased from 0.3834 to 0.3544 and when removed uncertainty from DEM the accuracy improved and obtained after 45 epoch ~91.1% to ~92.6% and loss from 0.2501 to 0.2232. figure 9 illustrate the relationship between loss and epochs for CDR-DEM in (a)and for CDR-DEM in (b), furthermore in figure 10 we observe the improvement in accuracy When representing the relationship between accuracy and epochs for CDR-DEM in (a)and CDR-DEM in (b) . It is clear from this that special data image affected by errors. And illustrated the effect the time of training by these errors as illustrated in table 4.

Table 4. Classification performance and Elapsed Time for DenseNet model

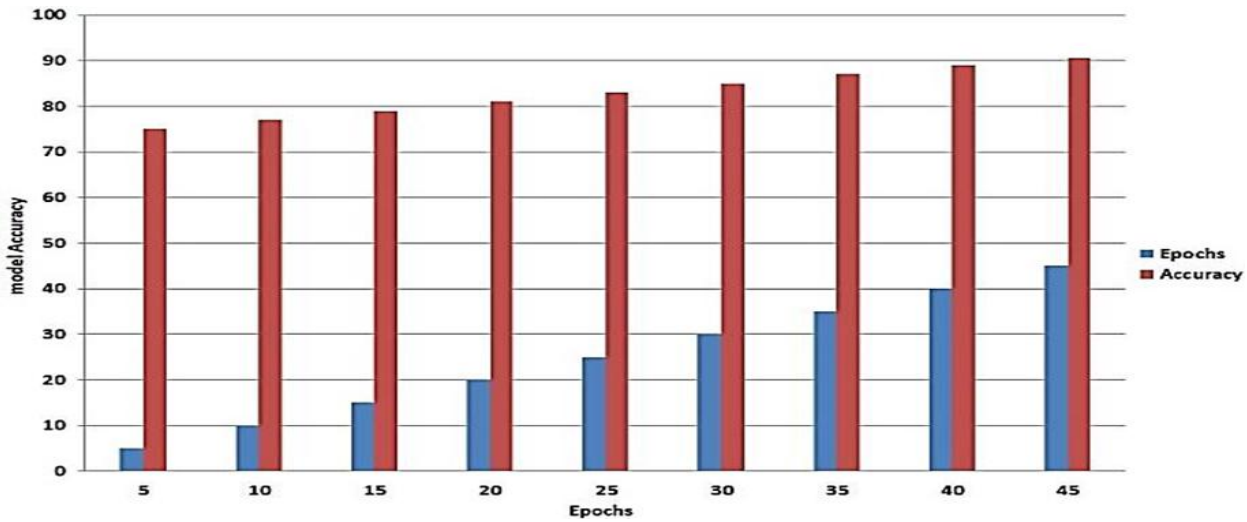
		CDR-DEM		
Kernel filter size	Epoch	Loss	Elapsed Time (~h:m)	Accuracy%
3×3	45	0.3834	1:25	89.189
5×5		0.3544	1:12	90.585
		CDER-DEM		
3×3	45	0.2501	1:20	91.096
5×5		0.2232	1:10	92.652

In paper[18] proposed level set algorithm using Python, The proposed level set method emulates water level decreasing from the spill point along the depression boundary to the lowest point at the bottom of a depression. By tracing the dynamic topological changes (i.e., depression splitting/merging) within a compound depression, the level-set method can construct topological graphs and derive geometric properties of the nested depressions. To test the applicability of the proposed level set algorithm for delineating nested depressions from DEMs at watershed and landscape scales, in Pipestem River watershed in the state of North Dakota in north America . The bare-earth LiDAR DEM can be downloaded by image tiles (2 9 2 km) with 1-m pixel resolution. Vertical accuracy of the LiDAR DEM is 15 cm. And actual sinks are computed depended on corrected and actual geographic data. The level set algorithm input needs to many spatial parameters for catchments area characteristic like catchments area and depths. But in our proposed method is an actual and automatically catchments detection method. That depends on features data and few many tiles of DEM in study area and efficient results in catchments location by training validation data in deep learning model which proven that is faster and accurate method.

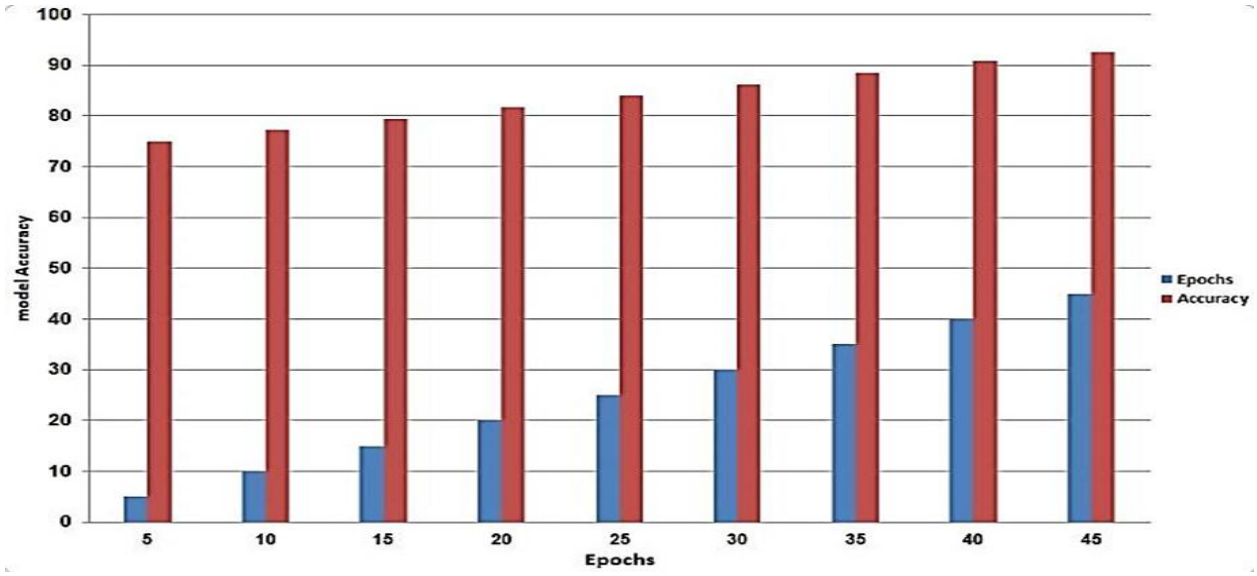


(a) CDR-DEM (b) CDER-DEM

Figure 9. Loss error curves obtained by two experiments for the training set (blue curves) and testing set (red curves): (a) CDR-DEM method, (b) CDER-DEM method



(a)



(b)

Figure 10. The relationship between Accuracy and epochs for two methods: (a) CDR-DEM method, (b) CDER-DEM method

CONCLUSION AND FUTURE WORK

In this work, two experiments were used to detect catchments regions in the study area. The first CDR-DEM used raw data DEM. which results catchments labels with an accuracy of 90.6%. In the second CDER-DEM, the uncertainty regions detected and eliminated from the DEM. The operations of Removing errors from DEMs provide more information and efficient results in training and produced more accuracy to 92.6%, as shown in Table 3.worked binary classification in deep learning using DenseNet model to classify DEM to two classes to detect catchments region in study area.

Using the integration between GIS software like ArcGIS, level set algorithm, and deep learning to predict the catchments regions in the digital elevation model achieved Promising results even with limited training data. Spatial Visualization of catchments is an essential part of several strategies to enable land cover analysts for decision-makers.

We tend to use the advantages of integrating between GIS software and powerful deep learning models to predict flash floods in future work.

REFERENCES

[1] Zhou, Qiming. "Digital elevation model and digital surface model." International Encyclopedia of Geography: People, the Earth, Environment and Technology (2017): 1-17.

[2] Nelson, A., H. I. Reuter, and P. Gessler. "DEM production methods and sources." *Developments in soil science* 33 (2009): 65-85.

[3] Jiang, Ling, Yang Hu, Xilin Xia, Qiuhua Liang, Andrea Soltoggio, and Syed Rezwan Kabir. "A multi-scale mapping approach based on a deep learning CNN model for reconstructing high-resolution urban DEMs." Water 12, no. 5 (2020): 1369.

[4] Thompson, James A., Jay C. Bell, and Charles A. Butler. "Digital elevation model resolution: effects on terrain attribute calculation and quantitative soil-landscape modeling." Geoderma 100, no. 1-2 (2001): 67-89

[5] Szypuła, Bartłomiej. "Quality assessment of DEM derived from topographic maps for geomorphometric purposes" Open Geosciences 11, no. 1 (2019): 843-865

[6] Amante, Christopher J. "Estimating coastal digital elevation model uncertainty." Journal of Coastal Research 34, no. 6 (2018): 1382-1397.

[7] Wechsler, Suzanne P., and Charles N. Kroll. "Quantifying DEM uncertainty and its effect on topographic parameters." Photogrammetric Engineering & Remote Sensing 72, no. 9 (2006): 1081-1090.

[8] Fisher, Peter F., and Nicholas J. Tate. "Causes and consequences of error in digital elevation models." Progress in physical Geography 30, no. 4 (2006): 467-489.

[9] Jones, Richard. "Algorithms for using a DEM for mapping catchment areas of stream sediment samples." *Computers & Geosciences* 28, no. 9 (2002): 1051-1060.

[10] Pashaei, Mohammad, Hamid Kamangir, Michael J. Starek, and Philippe Tissot. "Review and evaluation of deep learning architectures for efficient land cover mapping with UAS hyper-spatial imagery: A case study over a wetland." Remote Sensing 12, no. 6 (2020): 959.

[11] Arun, Pattathal Vijayakumar. "A comparative analysis of different DEM interpolation methods." The Egyptian Journal of Remote Sensing and Space Science 16, no. 2 (2013): 133-139.

[12] Leitao, Joao Paulo Correia. "Enhancement of digital elevation models and overland flow path delineation methods for advanced urban flood modelling." PhD diss., Imperial College London, 2009.

[13] Lafarge, Florent, Xavier Descombes, Josiane Zerubia, and Marc Pierrot-Deseilligny. "Structural approach for building re-construction from a single DSM." IEEE Transactions on pattern analysis and machine intelligence 32, no. 1 (2008): 135-147.

[14] Zhang, Liangpei, Lefei Zhang, and Bo Du. "Deep learning for remote sensing data: A technical tutorial on the state of the art." IEEE Geoscience and Remote Sensing Magazine 4, no. 2 (2016): 22-40.

[15] Lawaniya, Ruchi, and Manish Pandey. "A Fast Priority-Flood Algorithm with Pruning for Depression Filling in Hydrologic Analysis." International Journal of Computer Science and Information Security 14, no. 7 (2016): 694.

[16] Costante, Gabriele, Thomas A. Ciarfuglia, and Filippo Biondi. "Towards monocular digital elevation model (DEM) estimation by convolutional neural networks-application on synthetic aperture radar images." In EUSAR 2018; 12th European Conference on Synthetic Aperture Radar, pp. 1-6. VDE, 2018.

[17] Zhu, D., Q. Ren, Y. Xuan, Y. Chen, and I. D. Cluckie. "An effective depression filling algorithm for DEM-based 2-D surface flow modelling." Hydrology and Earth System Sciences 17, no. 2 (2013): 495-505.

[18] Wu, Qiusheng, Charles R. Lane, Lei Wang, Melanie K. Vanderhoof, Jay R. Christensen, and Hongxing Liu. "Efficient delineation of nested depression hierarchy in digital elevation models for hydrological analysis using level-set method." JAWRA Journal of the American Water Resources Association 55, no. 2 (2019): 354-368.

[19] Wu, Qiusheng, and Charles R. Lane. "Delineating wetland catchments and modeling hydrologic connectivity using lidar data and aerial imagery." Hydrology and earth system sciences 21, no. 7 (2017): 3579-3595.

[20] Winter, Thomas C., and James W. LaBaugh. "Hydrologic considerations in defining isolated wetlands." Wetlands 23, no. 3 (2003): 532-540.

[21] Torres, Rocio Nahime, Piero Fraternali, Federico Milani, and Darian Frajberg. "Automatic feature extraction to support Mountains Mapping in OSM." Editors (2019): 31.

[22] Liu, Xiaoyu, Diyu Yang, and Aly El Gamal. "Deep neural network architectures for modulation classification." In 2017 51st Asilomar Conference on Signals, Systems, and Computers, pp. 915-919. IEEE, 2017.

[23] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely connected convolutional networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708. 2017.

[24] Digital Elevation model images that downloaded in 2-3-2020 from https://earthexplorer.usgs.gov/

[25] Olusina, J. O., and C. J. Okolie. "Visualisation of uncertainty in 30m resolution Global Digital Elevation Models: SRTM v3. 0 and ASTER v2." Nigerian Journal of Technological Development 15, no. 3 (2018): 77-83.

[26] Hanaa A. Sayed, Ahmed M. Sefelnasr and Essam KH.Abd-Elmaged. “Deep Learning Approach for Detecting Digital Elevation Model (DEM) Uncertainty to Enhance Assessment of Water Resources”, in the Proceedings of the 29th International Conference on Computer Theory and Applications (ICCTA2019).

[27] Omran, A. A. "Integration of remote sensing, geophysics and GIS to evaluate groundwater potentiality–a case study in Sohag Region, Egypt." In The 3rd international conference on water

resources and arid environments and the 1st Arab Water Forum. 2008.
Xu, Yongyang, Zhong Xie, Yaxing Feng, and Zhanlong Chen. "Road extraction from high-resolution remote sensing imagery using deep learning." Remote Sensing 10, no. 9 (2018): 1461.

[28] Ho, Yaoshiang, and Samuel Wookey. "The real-world-weight cross-entropy loss function: Modeling the costs of mislabel-ing." IEEE Access 8 (2019): 4806-4813.

[29] LEE, RICH, and ING-YI CHEN. "The Time Complexity Analysis of Neural Network Model Configurations." In 2020 International Conference on Mathematics and Computers in Science and Engineering (MACISE), pp. 178-183. IEEE, 2020.

[30] Justus, Daniel, John Brennan, Stephen Bonner, and Andrew Stephen McGough. "Predicting the computational cost of deep learning models." In 2018 IEEE international conference on big data (Big Data), pp. 3873-3882. IEEE, 2018.