IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 9, September 2020

DOI 10.17148/IJIREEICE.2020.8904

Satellite Image Processing to Detect Buildings using Deep Learning

Claudia P Dsouza[1], Dr. K. P. Shashikala[2]

M. Tech Student, Dept. of Electronics and Communication Engineering, DSCE, Bangalore, India[1]

Associate Professor, Dept. of Electronics and Communication Engineering, DSCE, Bangalore, India[2]

Abstract: Detecting Buildings from high resolution satellite images is incredibly helpful in map making, land use analysis, urban planning etc. If a human expert wants to manually extract this valuable data, it is difficult. So there is one potential solution to extract this data by using automatic techniques. In this project we are going to use deep learning approach to detect buildings from the satellite images. We have downloaded a large amount of dataset because deep learning model requires a huge dataset for training the model. Then pre-processing of these image like cropping, resizing and removal of noise by using Gaussian filter is being done. Split the dataset into training and validation set. After gathering all the dataset, we should train the model. In training process, the dataset undergoes through convolutional neural networks for different layers. Once training is completed the model has been built and we test the model by applying different input images and finally we get only the detected buildings as output. In this project we have also done the greenery detection by using HSV (Hue, Saturation, Value) colour format. We are using python for coding purpose.

Keywords: Satellite Images, Building Detection, Convolutional Neural Networks and Deep learning.

I. INTRODUCTION

The Satellite images provide valuable resource of information, which when used completely can give actionable insights for us to analyse. Analysis on satellite images can be performed to recognize buildings, monitor oil, detect vegetation land, etc. In our project, we detect buildings so that rate-of-urbanization can be drawn. Building detection from satellite images is being a basic but a difficult issue because it wants correct recovery of building footprints from high resolution images. This project builds a model using a deep learning algorithm that detects buildings from satellite images.

A. *Conventional Methods*

A various conventional method has been used to detect buildings. Few of these methods depends on shape, edges and features characteristics of satellite images.

B. *Drawbacks of Conventional Methods*

a. There are some methods which require to construct a visual dictionary and then make use of the trained visual words to detect an urban region in a new satellite image.
b. Some algorithms are computationally expensive.
c. There are some methods which uses the distribution of feature points to identify an urban area without doing any training process, but due to inaccurate feature points that leads to detection of some false objects in the image.
d. Some methods require to cooperate with various other images to accomplish the detection. The results of these are not sufficiently accurate.

C. *Objective of the project*

The main objectives of this project work is:

- A Deep Learning model is developed by using Convolutional Neural Networks technique to detect the buildings in an area using satellite data, to analyse and investigate rate-of-urbanization.
- Use of some simulation tools of python modules, like Spyder or Jupiter Notebook to observe the simulation results.
- Importing libraries such as Keras, Tensorflow and OpenCV which are mainly used for image processing.

II. LITERATURE SURVEY

This section gives the complete details of researches and their work regarding the existing techniques that have been used to automatically detect the Building from satellite images, remote sensing images and aerial images. Some of them are explained below.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 9, September 2020

DOI 10.17148/IJIREEICE.2020.8904

Liu Wei et al., [1] they have proposed a probability model and efficient plan to extract buildings object from high-resolution panchromatic image in dense urban area. They have segmented the image into regions to treat all extracted regions contours as candidates and utilizes probability model to select the true buildings in the satellite image. The features computation has been done to characterise the buildings.

Zongying Song et al., [2] they have proposed a region-based approach for building detection in high resolution satellite image. Model has been built with shape features and texture from training building set. They have over-segmented the input image into numerous small regions, they have also identified regions which have comparable pattern with building model these are the build like regions. By making the group of BLRs a Candidate building regions is developed which has comparable shape with the building model. Lines which have solid relationship with every CBR are extracted. They have also generated the 2D rooftop hypotheses from CBR and lines. The hypotheses are verified by using the shadow and geometric rules.

Parvaneh Saeedi et al., [3] they have development an automated roof detection system from single monocular electro-optic satellite imagery. The framework utilizes a new methodology in which each input image segmented at several levels. It generates a set of quadrilateral rooftop hypotheses from border line definitions of such segments joined with line segments recognized on the original image. For each hypothesis a probability score is figured that shows the proof of true buildings as per the image gradient field and line segment definitions. It detects small gabled residential rooftops.

Beril Sirmacek et al., [4] they have proposed the use of scale invariant feature transform (SIFT) and graph theoretical tools. SIFT keypoints are incredible in detecting objects but alone they are not sufficient in detecting urban area and buildings, so graph theory has been used. In formation of graph each keypoint is represented as vertex of the graph. The unary and binary connections between these vertices lead to the edges of the graph. Based on this they have extracted the urban areas using multiple subgraph matching method.

Melih Cetin et al., [5] they have proposed a method based on textural features and adaboost for detecting buildings in satellite images. It is an algorithm that uses both classification and deciding the helpful feature subset due to its feature selector nature. It also includes morphological operators which are applied for post processing.

Beril Sirmaçek et al., [6] they have proposed building detection method using local feature vectors and a probabilistic framework. The local feature vectors extraction method serves as observation of the probability density function to be assessed. The building location in the image are taken as joint random variable and estimate their probability density function. By using the modes of estimated density and probabilistic properties the building location in the image are detected.

Caglar Senaras et al., [7] they have proposed a method to overcome the problem of building detection in satellite images. This strategy utilizes two-layer hierarchical classification mechanism for ensemble learning. In this each part is divided by N different classifiers using different features at the primary layer. The class enrolment values of the segments are taken from various base layer classifiers are ensemble to shape a new fusion space which is linearly separable simplex. Further this simplex is divided by linear classifier at meta layer.

M. Vakalopoulou et al., [8] they have proposed an automated building detection framework from very high resolution remote sensing data based on deep convolutional neural networks. The method used is based on supervised classification procedure utilizing a huge training dataset. An MRF model is then responsible for getting the optimal labels with respect to detect the scene structures.

Naveen Chandra et al., [9] they have researched about the human cognitive capabilities by integrating cognitive task analysis for extraction of data from satellite images. Knowledge has been collected about the sequence of cognitive processes which human being uses during the interpretation and classification of images. The Rule based methodology for the portrayal of the knowledge which is acquired from the visual interpretation of image by the human beings. Defined rules are used to detect the buildings using mixture tuned matched filtering algorithm.

Li Sun et al., [10] they have proposed a two-stage CNN model to detect rural buildings in high-resolution imagery. By simulating the hierarchical processing system of human vision, the proposed model is built with two convolutional neural networks whose structures can automatically locate villages and recognize buildings.

Xiangyu Zhuo et al., [11] they have dealt with the problem of training the large amount of data which requires more time. So they have proposed to generate image annotations by transferring labels from aerial images to UAV images and refine the annotations using a densely connected CRF model with an embedded naive Bayes classifier. The generated annotations give right semantic labels and also protect class boundaries. To approve the utility of automatic annotations deploy them as training data for pixel-wise image segmentation.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 9, September 2020

DOI 10.17148/IJIREEICE.2020.8904

Geesara Prathap et al., [12] they have proposed a deep learning approach for building detection. The pre-processing of the dataset is done by 2-sigma percentile normalization. The data preparation involves ensemble modelling in which 3 models are created. The Binary Distance Transformation is utilized for better data labelling process and the U-NET is altered by putting on batch normalization wrappers.

Zeshan Lu et al., [13] they have presented a large-scale high-resolution building dataset named 5M Building after the quantity of tests in the dataset. The dataset is collected which comprises of more than 10 thousand images from GaoFen-2 with a spatial resolution of 0.8 meter. The dataset is presented with a baseline for evaluating 3 state of the art CNN based detectors.

Atmika Shetty et al., [14] they have proposed a model that predict the development or socio-economic status of areas by analysing their satellite images using CNN classification and image segmentation techniques. They have used two approaches supervised and unsupervised. They have done pre-processing and convolutional neural networks are used for classification, color HeatMap, contouring and segmentation it will show the output in the form of percentage of occurrence along with the socio-economic status.

III. METHODOLOGY

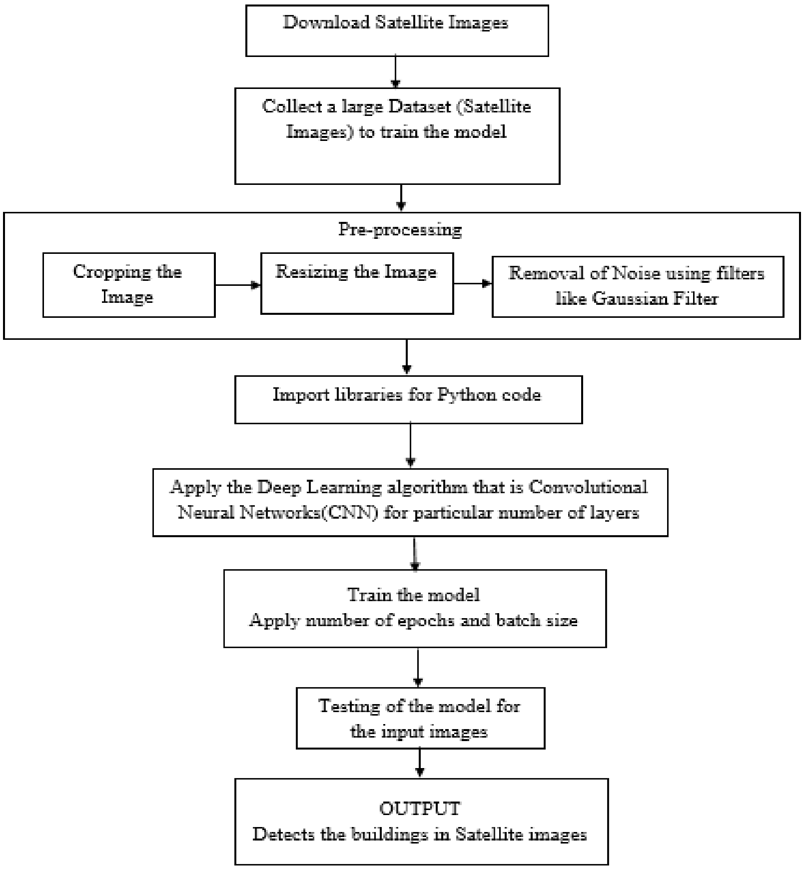

Fig. 1: Flow diagram of the proposed method.

Building Detection: The building detection can be done by the following process. It consists of dataset, importing libraries, training and testing the model. The fig. 1 shows the flow diagram of the proposed methodology.

A. *Dataset:*

1. *Download Dataset:* Download huge amount of dataset because for any deep learning model we require large amount of data to train the model. In this project we have downloaded the dataset from this website cs.toronto.edu/~vmnih/data/ it provides a lot of dataset for deep learning. Dataset should consist of raw images list and mask (labels or targets) list. The above website provides these kinds of dataset.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 9, September 2020

DOI 10.17148/IJIREEICE.2020.8904

2. *Pre-processing:*
   Pre-processing of the image which includes:
   i. Cropping: cropping of the image is done for removal of unwanted outer areas from a particular satellite image.
   ii. Resizing the image: Fig. 2 shows the resized image output. In this the original satellite image has been resized by using the following command in python code:
   resized_image=cv2.resize(img,(256,256))
   iii. Gaussian Filter: It is a linear filter. Used to reduce noise or blur the image. The Gaussian filter alone is sufficient to blur edges and reduce noise. The fig. 3 shows the output of Gaussian filter. The command used for Gaussian filtering is as follows:
   blur=cv2.GaussianBlur(gray,(5,5),0)

The 3 parameters considered for Gaussian filter are:
- Requires input image to be blurred.
- The kernel size must be an odd number: The kernel is the matrix that the algorithm uses to look over the image & in this situation the kernel is 5x5, where the center pixel is the pixel that will be changed as for the encompassing 24.
- Sigma: The width of the curve in X and Y directions. Separate values may be passed too. In the event that 0, it will be auto-determined from the kernel size.

Fig. 2 Resized image in colour

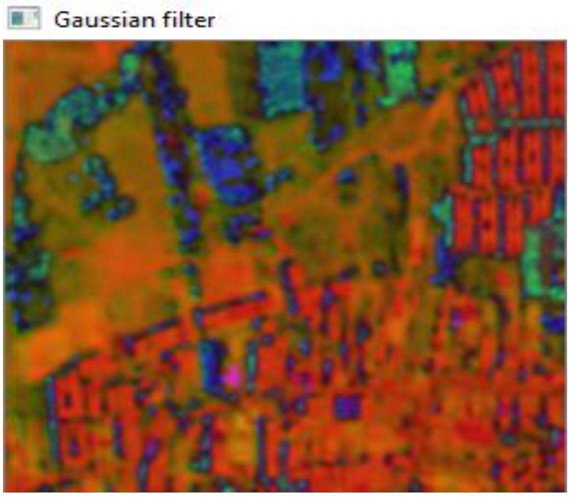

Fig. 3 Gaussian filter image after removal of noise

3. *Split the image:* The images are split into Training set (consists of images and masks) and Validation set (images and masks). Training means it will learn which is building and which is not building so it is given with both the images and masks. But in validation, images are considered as testing phase and it will detect buildings in these images and whatever answers it gets it tries to match with the masks.

4. *Data generation:* It is used to generate more number of images from the available images.

In this section the dataset is ready to train the model because it consists of huge amount of data which is pre-processed and also the dataset splits into 2 parts (training and validation) and data generation is done.

B. *Import Libraries*

In this project we are using python for coding purpose. Python requires libraries for image processing. Some of them are:
- OpenCV (Open Source Computer Vision): It is a software library and comprises of normal tasks like reading/loading images, saving images, resizing images displaying images, playing a video, Gaussian filtering, median filtering, histogram, binary thresholding and turning on a camera. The OpenCV library can be installed by using the command called conda install –c conda-forge OpenCV.
- Keras: It is an open source neural network library. Keras contains various executions of normally utilized neural network building blocks. It a deep learning library. It is called as application program interface(API). Keras is a supported that comes in between computer and the tensorflow.
- Tensorflow: It is used for Deep learning models. Tensorflow is an open source artificial intelligence library utilizing data flow graphs to build models. Tensorflow is fundamentally utilized for classification of the image, prediction, discovering, understanding and perception. The computer system does not understand tensorflow library so, keras is an intermediate that helps the system to understand tensorflow library.

IJIREEICE
ISSN (Online) 2321-2004
ISSN (Print) 2321-5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
Vol. 8, Issue 9, September 2020
DOI 10.17148/IJIREEICE.2020.8904

- OS library: It is operating system library. This library communicates our program with the actual OS of the system.
- Numpy: It is a numerical python library. Used for working with arrays.
- Matplotlib: It is a plotting library used for python programming language.

C. *Training*

The training of the model is done by using the deep learning which uses CNN algorithm. Deep learning is used because it can handle a very large amount of data to train the model and also uses GPU (graphics processing unit) to rapidly process the data. A Convolutional Neural Network is called a deep learning algorithm which takes the input image, allot significance to different objects in the image and have the option to separate one from the other.

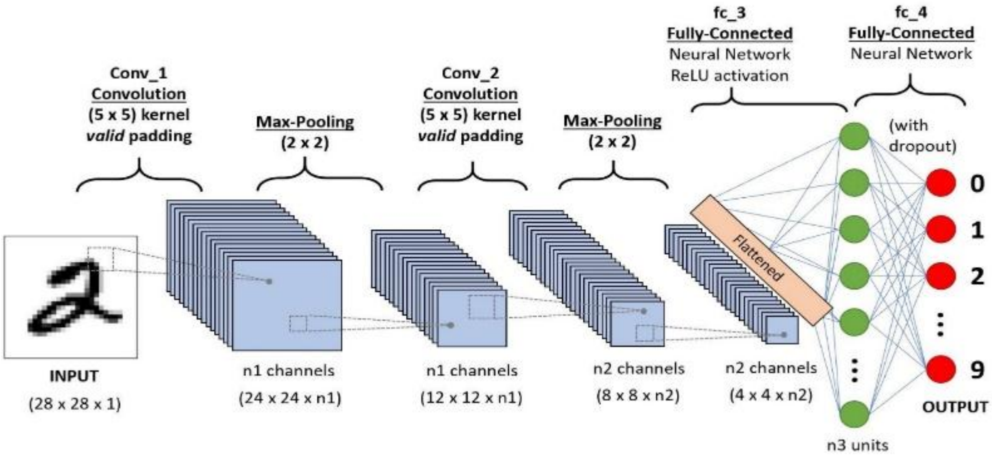

Fig. 4 Overview of Convolutional neural network

The fig. 4 shows the overview of CNN. The CNN learns to distinguish various features in a particular image using tens or several hidden layers. Every hidden layer expands the complexity of the learned image features. For ex: the first layer will learn to identify few edges and the last layer will learn to identify few difficult shapes.

In this training of the model is dependent on the following parameters like:

- Weights: how much point it will get for prediction. These points are assigned randomly to the images.
- Epoch: how much time it should go from training to validation step and predict the value.
- Batch size: no. of images that can be used at a time.
- No. of channels: means how the image is going to read in form of RGB, BGR or HSV format.
- Learning rate: It is a ratio of right and wrong prediction. If we give less learning rate, the model will try to achieve the learning rate to train the model.
- Loss: The loss is the amount of wrong prediction.

In our project the convolutional neural network has 9 layers. By using more number of layers the accuracy of the detected buildings increases. The fig. 5 shows the convolutional operation in our project in which first we need to define the input variables and then the convolution operation should be performed as follows

conv1= name of the output variable
Conv2D= name of the function with 2D images
(3,3) = size of the matrix
activation= This function is used to assign weights
elu= one type of activation
kernel_intializer= background kernel
he_normal=standard value
(padding='same')= addition of same values
BatchNormalization()= Initializing the batch normalization
Dropout= images that are not cooperating or images that are not useful are dropped out.

The output that comes from 1st layer is applied as input to the 2nd layer. Then the output of 2nd layer is applied as input to the 3rd layer and the process goes on continuing up to 9th layer.

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 9, September 2020

DOI 10.17148/IJIREEICE.2020.8904

```
conv1 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (inputs)
conv1 = BatchNormalization() (conv1)
conv1 = Dropout(0.1) (conv1)
conv1 = Conv2D(16, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (conv1)
conv1 = BatchNormalization() (conv1)
pooling1 = MaxPooling2D((2, 2)) (conv1)

conv2 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (pooling1)
conv2 = BatchNormalization() (conv2)
conv2 = Dropout(0.1) (conv2)
conv2 = Conv2D(32, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (conv2)
conv2 = BatchNormalization() (conv2)
pooling2 = MaxPooling2D((2, 2)) (conv2)

conv3 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (pooling2)
conv3 = BatchNormalization() (conv3)
conv3 = Dropout(0.2) (conv3)
conv3 = Conv2D(64, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (conv3)
conv3 = BatchNormalization() (conv3)
pooling3 = MaxPooling2D((2, 2)) (conv3)

conv4 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (pooling3)
conv4 = BatchNormalization() (conv4)
conv4 = Dropout(0.2) (conv4)
conv4 = Conv2D(128, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (conv4)
conv4 = BatchNormalization() (conv4)
pooling4 = MaxPooling2D(pool_size=(2, 2)) (conv4)

conv5 = Conv2D(256, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (pooling4)
conv5 = BatchNormalization() (conv5)
conv5 = Dropout(0.3) (conv5)
conv5 = Conv2D(256, (3, 3), activation='elu', kernel_initializer='he_normal', padding='same') (conv5)
conv5 = BatchNormalization() (conv5)
```

Fig. 5 Convolutional operation in our project

D. *Testing*

Once the training process is completed the model is built and we test this model for any number of input images. The fig 6 shows completion of training process in which we have used 100 epochs to increase the accuracy of the detected buildings. The fig 7 shows the output of the detected buildings which is the final output of our project. We have also showed buildings in the satellite images in bounding boxes. The fig. 8 shows the output of this.

```
Console 1/A
Epoch 84/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1160 - accuracy: 0.9404 - val_loss: 0.1802 - val_accuracy: 0.9212
Epoch 85/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1199 - accuracy: 0.9395 - val_loss: 0.1858 - val_accuracy: 0.9200
Epoch 86/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1152 - accuracy: 0.9406 - val_loss: 0.1874 - val_accuracy: 0.9198
Epoch 87/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1177 - accuracy: 0.9399 - val_loss: 0.1829 - val_accuracy: 0.9219
Epoch 88/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1150 - accuracy: 0.9408 - val_loss: 0.1813 - val_accuracy: 0.9217
Epoch 89/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1145 - accuracy: 0.9408 - val_loss: 0.1819 - val_accuracy: 0.9213
Epoch 90/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1144 - accuracy: 0.9409 - val_loss: 0.1873 - val_accuracy: 0.9184
Epoch 91/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1123 - accuracy: 0.9413 - val_loss: 0.1851 - val_accuracy: 0.9192
Epoch 92/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1131 - accuracy: 0.9413 - val_loss: 0.1770 - val_accuracy: 0.9234
Epoch 93/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1132 - accuracy: 0.9413 - val_loss: 0.1800 - val_accuracy: 0.9217
Epoch 94/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1161 - accuracy: 0.9405 - val_loss: 0.1803 - val_accuracy: 0.9222
Epoch 95/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1111 - accuracy: 0.9416 - val_loss: 0.1800 - val_accuracy: 0.9236
Epoch 96/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1117 - accuracy: 0.9414 - val_loss: 0.1854 - val_accuracy: 0.9179
Epoch 97/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1101 - accuracy: 0.9419 - val_loss: 0.1774 - val_accuracy: 0.9228
Epoch 98/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1093 - accuracy: 0.9422 - val_loss: 0.1888 - val_accuracy: 0.9175
Epoch 99/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1085 - accuracy: 0.9424 - val_loss: 0.1764 - val_accuracy: 0.9223
Epoch 100/100
6274/6274 [==============================] - 81s 13ms/sample - loss: 0.1093 - accuracy: 0.9421 - val_loss: 0.1804 - val_accuracy: 0.9218
Traceback (most recent call last):
```

Fig. 6 Training process

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 9, September 2020

DOI 10.17148/IJIREEICE.2020.8904

E. *Greenery Detection*

In this project we have also done greenery detection. By using HSV colour format we have detected only the green part in the satellite image. HSV (hue, saturation, value) represents model the way paints of various hues combine, with the saturation measurement resembling various colours of splendidly coloured paint and the value is the mixture of those paints with varying amount of black or white paint. If we want to get only the green composition in a satellite image we have some standard value as (36,25,25) ~ (86,255,255) it means hue varies from 36 to 86, saturation varies from 25 to 255 and value varies from 25 to 255. HSV has different values for different colours for this we can check the official HSV document. The fig.8 shows the greenery part detection.

IV. OUTPUTS

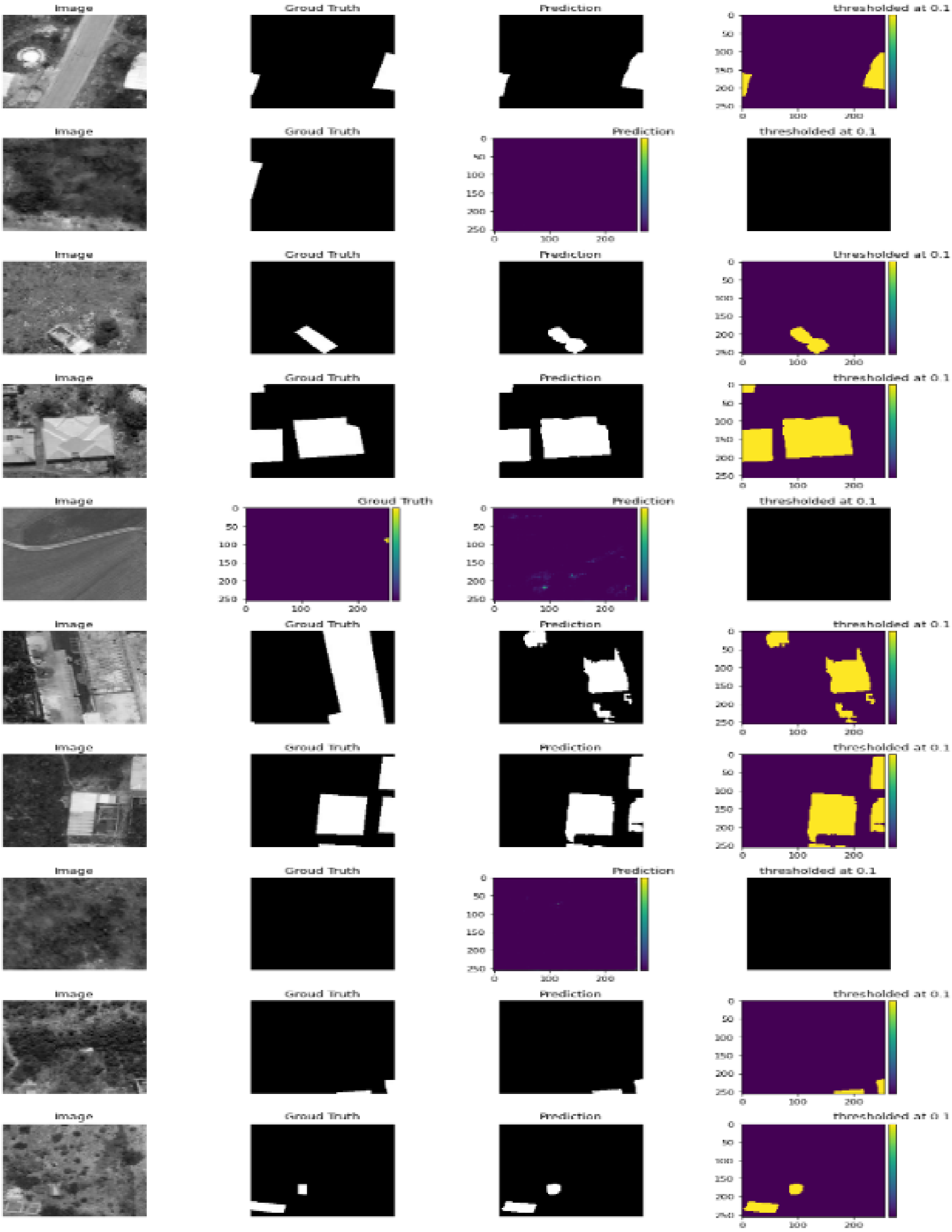

Fig. 7 Final output showing the detected buildings for input images=10 with epochs=100

IJIREEICE
ISSN (Online) 2321-2004
ISSN (Print) 2321-5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
Vol. 8, Issue 9, September 2020
DOI 10.17148/IJIREEICE.2020.8904

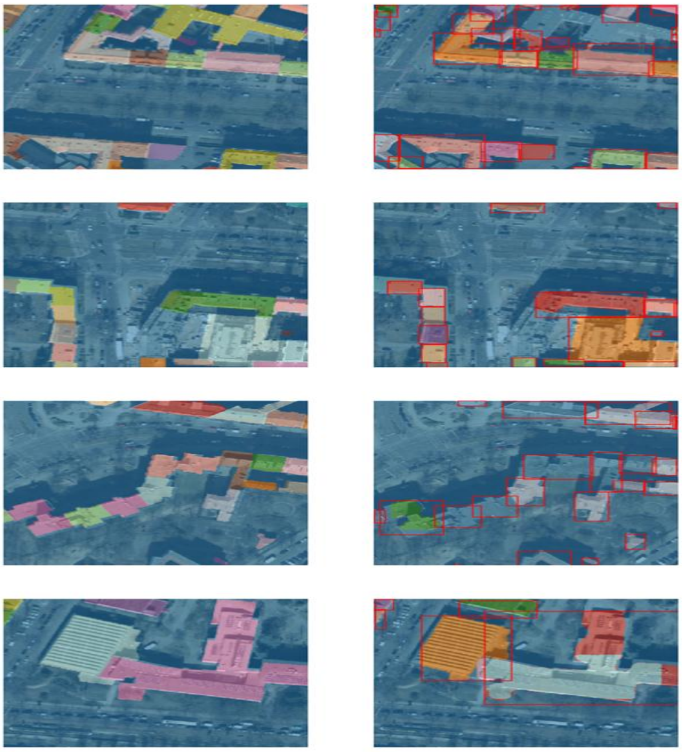

Fig.8 Buildings detected in bounding boxes for input images=4 with epochs=39

(9a) Input (9b) Output

Fig 9a & 9b shows the output of the greenery detection.

V. APPLICATIONS

The applications of this project are as follows:

- Provides valuable information in land exploration and national construction.
- Used for different urban applications such as city planning, infrastructure development, state cadastral inspection, provision of municipal services etc.
- High resolution satellite images are utilized to identify buildings for mapping applications.
- Used for Monitoring the rate-of-urbanization.
- Telecommunication companies need to make proper decision in situating the transmitter station to get a wide area coverage. The transmitter stations should be situated in such an area where there are no buildings, lakes, rivers etc.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 8, Issue 9, September 2020

DOI 10.17148/IJIREEICE.2020.8904

VI. CONCLUSION

In this project by using deep learning algorithm we have detected the buildings in the satellite images which are useful in many urban applications. We have gained an accuracy of 92.2% across the entire output image and also we have gained less loss. There are many methods which are trying to increase the accuracy by doing so it will be helpful for many urban applications.

REFERENCES

[1]. Liu Wei and V. Prinet, “Building Detection from High-resolution Satellite Image Using Probability Model”, IEEE International Geoscience and Remote Sensing Symposium, pp. 3888-3891, Aug. 2005.
[2]. Zongying Song, Chunhong Pan, Q Yang, “A region-based approach to building detection in densely build-up high resolution satellite image”, International Conference on Image Processing, pp. 3225-3228, Oct. 2006.
[3]. Parvaneh Saeedi and Harold Zwick, “Automatic Building Detection in Aerial and Satellite Images”, 10th International Conference on Control, Automation, Robotics and Vision, IEEE, pp. 623-629, Dec. 2008.
[4]. Beril Sirmacek and Cem Unsalan “Urban-Area and Building Detection Using SIFT Keypoints and Graph Theory”, IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no.4, pp.1156-1167, April 2009.
[5]. Melih Cetin, Ugur Halici and Orsan Aytekin, “Building detection in satellite images by textural features and Adaboost”, IAPR Workshop on Pattern Recognition in Remote Sensing, August 2010.
[6]. Beril Sirmaçek and Cem Unsalan, “A Probabilistic Framework to Detect Buildings in Aerial and Satellite Images”, IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 1, pp. 211-221, January 2011.
[7]. Caglar Senaras, Baris Yuksel, Mete Ozay and Fatos Yarman-Vural, “Automatic Building Detection with feature space fusion using ensemble learning”, IEEE International Geoscience and Remote Sensing Symposium, pp. 6713-6716, July 2012.
[8]. M. Vakalopoulou, K. Karantzalos, N.Komodakis, N. Paragios, “Building Detection in very high resolution multispectral data with Deep Learning features”, IEEE International Geoscience and Remote Sensing Symposium IGARSS , pp. 1873-1876, July 2015.
[9]. Naveen Chandra, Jayanta Kumar Ghosh and Ashu Sharma, “A Cognitive Based Approach for Building Detection from High Resolution Satellite Images” International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring), pp. 1-5, April 2016.
[10]. Li Sun, Yuqi Tang, and Liangpei Zhang, “Rural Building Detection in High-Resolution Imagery Based on a Two-Stage CNN Model”, IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 11, pp. 1998-2002, Nov. 2017.
[11]. Xiangyu Zhuo, Friedrich Fraundorfer, Franz Kurz and Peter Reinartz, “Building Detection and Segmentation using a CNN with automatically generated training data”, IEEE International Geoscience and Remote Sensing Symposium, pp. 3461-3464, July 2018.
[12]. Geesara Prathap and Ilya Afanasyev, “Deep Learning Approach for Building Detection in Satellite Multispectral Imagery”, International Conference on Intelligent Systems (IS), IEEE, Sep.2018.
[13]. Zeshan Lu, Tao Xu, Kun Liu, Zhen Liu, Feipeng Zhou and Qingjie Liu, “5M-Building: A Large-Scale High-Resolution Building Dataset with CNN based Detection Analysis”, IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1385-1389, Nov. 2019.
[14]. Atmika Shetty, Arya Thorat, Rumjhum Singru, Mrunali Shigawan and Vidya Gaikwad, “Predict Socio-Economic Status of an Area from Satellite Image Using Deep Learning”, International Conference on Electronics and Sustainable Communication Systems (ICESC) pp. 177-182, July 2020.