Taylor & Francis
Taylor & Francis Group

Geocarto International

ISSN: 1010-6049 (Print) 1752-0762 (Online) Journal homepage: https://www.tandfonline.com/loi/tgei20

Building footprint extraction in Yangon city from monocular optical satellite image using deep learning

Hein Thura Aung, Sao Hone Pha & Wataru Takeuchi

To cite this article: Hein Thura Aung, Sao Hone Pha & Wataru Takeuchi (2020): Building footprint extraction in Yangon city from monocular optical satellite image using deep learning, Geocarto International, DOI: 10.1080/10106049.2020.1740949

To link to this article: https://doi.org/10.1080/10106049.2020.1740949

Published online: 20 Mar 2020.

Submit your article to this journal

Article views: 97

View related articles

View Crossmark data

GEOCARTO INTERNATIONAL
https://doi.org/10.1080/10106049.2020.1740949

Building footprint extraction in Yangon city from monocular optical satellite image using deep learning

Hein Thura Aung[a], Sao Hone Pha[b] and Wataru Takeuchi[c]

[a]Department of Electronic Engineering, Yangon Technological University, Insein, Myanmar; [bb]Remote Sensing and GIS Research Center, Yangon Technological University, Yangon, Myanmar; [c]Institute of Industrial Science, The University of Tokyo, Tokyo, Japan

ABSTRACT

In this research, building footprints in Yangon City, Myanmar are extracted only from monocular optical satellite image by using conditional generative adversarial network (CGAN). Both training dataset and validating dataset are created from GeoEYE image of Dagon Township in Yangon City. Eight training models are created according to the change of values in three training parameters; learning rate, β_1 term of Adam, and number of filters in the first convolution layer of the generator and the discriminator. The images of the validating dataset are divided into four image groups; trees, buildings, mixed trees and buildings, and pagodas. The output images of eight trained models are transformed to the vector images and then evaluated by comparing with manually digitized polygons using completeness, correctness and F1 measure. According to the results, by using CGAN, building footprints can be extracted up to 71% of completeness, 81% of correctness and 69% of F1 score from only monocular optical satellite image.

ARTICLE HISTORY
Received 29 July 2019
Accepted 22 February 2020

KEYWORDS
GeoEYE monocular RGB image; learning rate; momentum; pix2pix; pixel-based evaluation

1. Introduction

According to the UN's report in 2014, 54% of world's population lived in the urban area (United Nations 2014). Yangon is the former capital of Myanmar, one of the developing countries in the world, whose urban area increased seven times from 1973 to 2015 in 42 years (Sritarapipat and Takeuchi 2018) and Myanmar population also increased 2.5 times from 1973 to 2014 (Ministry of Immigration and Population 2015). Although the urban area and urban population are growing, urban central functions are still in central business district with a population density of 365.5 persons/ha (Japan International Cooperation Agency (JICA) 2013). Urban building maps are one of the most important database for further urban related applications such as urban planning, urban management, natural resource management and so on for Yangon City. As of Japan International Cooperation Agency (JICA) (2018), there are 18 ongoing projects in Yangon region and its environment, all of which are in collaboration with Japan International Cooperation Agency (JICA) and other government institutions such as Yangon City Development

CONTACT Hein Thura Aung heinthuraung@ytu.edu.mm

Center (YCDC). These projects include mapping, improving transportation service and water supply management. Current urban building mapping project of Yangon city is mainly based on manual digitization of very high resolution (VHR) satellite image. Manual digitization gives better accuracy but takes a lot more time and human resources, and are more difficult for future updates. Thus, automatic digitization which is based on image processing techniques, machine learning and deep learning algorithms, is more preferable.

Several research works which are based on image processing techniques are as follows. In Yanfeng Wei (2004), urban building footprints extraction was done from high resolution panchromatic satellite image from QuickBird based on unsupervised clustering and Canny edge detection. In their work, shadow was one of the information sources for investigation of building presence. In Liu and Prinet (2005), QuickBird satellite image and probability model was used to detect building footprints. Segmentation for possible building candidates was the first step and using probability model to select true buildings was the second step. In Sirmacek and Unsalan (2008), building detection was done from aerial images. They also used segmentation by invariant colour features and edge detection for building extraction. In Akçay and Aksoy (2010), Ikonos image was used to detect building by using segmentation and clustering based on minimum spanning trees using shadow and sun azimuth angle information. In Gurshamnjot Singh et al. (2015), building extraction was done only from single satellite image based on shadow information. Building length was calculated from shadow information along with sun azimuth information after segmentation of the satellite image. Graphical based pruning method was used to eliminate overlapped extraction of the same building. In Ok et al. (2013), arbitrarily shaped buildings were detected by fuzzy landscape generation approach based on shadow regions to generate possible building regions, and GrabCut partitioning to extract building regions. The authors tested on QuickBird and GeoEYE-1 images. The work of Ghaffarian and Ghaffarian (2014) was the combination of Purposive FastICA(PFICA) and binary k-means clustering. Monocular high resolution Google Earth images were used for building detection. The original FastICA was improved by initializing with the Moore-Penrose pseudo-inverse matrix for three regions, shadow and dark vegetation, roads and bare soil and buildings. The image was masked using those three regions and the masked images were then implemented by PFICA algorithm and finally clustered by using binary k-means clustering. In Ghaffarian and Ghaffarian (2014), shadow detection based on novel double thresholding technique was implemented on high resolution Google Earth images. Then, those shadow areas along with the illumination angle were used to automatically create buffer zone based on shadow shapes and sizes. Supervised parallelpiped classification which contains a new proposed thresholding method based on standard deviation of the classes was used to detect building areas and the training samples were collected from shadow pixels.

In Lopez et al. (2015), image segmentation was used to get segments of potential building regions and then classification with a threshold was used to classify the segments into buildings. They used two software packages; eCognition and InterIMAGE. In Oztimur Karadag et al. (2015), they proposed a domain-specific building segmentation which was based on domain-specific information (DSI). In their work, the image dataset was first segmented by domain-specific information and the image segments were fused based on decision fusion. Their work needed multispectral satellite images. In Kabolizade et al. (2010), building detection was based on digital surface model (DSM) and modified snake model using colour aerial images and light detection and ranging (LiDAR) data. DSM was used to detect initial building contour. In Koc-San and Turker (2012), the existing building database was automatically updated from IKONOS images. Support Vector Machine (SVM) classification was used for determining building and non-building patches. The input to SVM classification was the pan-sharpened satellite image which was orthorectified by normalized digital surface model (nDSM). Normalized difference vegetation index (NDVI) was also used in determining potential building patches. During update process using existing geographical information system (GIS) database, new, destroyed and existing building patches were considered. Whilst updating new and existing building patches, a new building model based on shape parameters was used for the comparison of detected building patches and existing building database, and any changes are updated to the existing database. Those research works were based on image processing techniques, and using image processing techniques for building extraction can automate the extraction process. But it requires multiple different datasets such as optical datasets and LiDAR datasets, and multiple algorithms have to be combined together which requires expert knowledge.

Using deep learning algorithms can reduce the requirement of different datasets (optical image, LiDAR image, etc.) and algorithms than using image processing techniques and machine learning algorithms. In Vakalopoulou et al. (2015), they used deep convolutional neural network to extract building footprints from very high resolution multispectral data. They used ImageNet to extract deep features of buildings and used them to train SVM classifier. Then, the buildings were finally extracted through a Markov random field (MRF)-based model. In Zhang et al. (2016), they first used multi-scale saliency computation to extract the area which are covered with buildings. And then they used R-CNN to extract building footprints. They finally used an improved non-maximum suppression (NMS) method to improve the results. In Zuming Huang et al. (2016), they used deep deconvolution neural networks (DeconvNet)-based building extraction method. In their work, they used two datasets to train the neural network as well as saliency map fusion to improve the result. In Chen et al. (2017), building extraction was performed by using deep deconvolutional neural network. They used two datasets from ISPRS 2D Semantic Labelling Challenge which contained near-red, red and green bands with DSM. In Bittner et al. (2018), VHR remote sensing images combined with normalized DSMs were used for building footprint extraction. They used fully convolutional network (FCN) for building detection from RGB image, panchromatic (PAN) image and nDSM images. In those building footprint extraction tasks, the training image includes multiple bands such as red, green and blue (RGB) bands, and infrared (IR) band. IR band is very useful for vegetation detection in urban and suburban regions, as in those regions, buildings are surrounded or sometimes partially occluded by trees. However, VHR satellite images with IR band are more expensive than RGB images. Moreover, they used deep learning algorithms which are based on convolutional neural network (CNN) which requires a lot of training samples.

In Yang et al. (2018), four CNN architectures were tested for building extraction at large scale for United States. Branch-out CNN, fully convolutional network (FCN), conditional random field as recurrent neural network (CRFasRNN), and SegNet were used for semantic pixel-wise labelling for building footprints. The datasets contain 1 m spatial resolution images with red, green, blue and near infrared bands. The training dataset contains 4000 images that were selected across the country. In Yang et al. (2018), Dense Networks (DenseNets) were combined with an attention mechanism to improve the extraction performance. DenseNets contain dense blocks which are iterative concatenation of their previous feature maps. Thus, they improve feature reuse between each feature map, and reduce the number of parameters. The attention mechanism was used to extract the most useful information of the input image. They used very high resolution true orthophoto (TOP) images with a resolution of 5 cm from the ISPRS 2D semantic labelling contest (Potsdam). In Xu et al. (2018), the deep learning model ResNet was combined with hand-crafted feature engineering as pre-processing and a guided filter as post processing to improve the classification result. They also used ISPRS 2D semantic labelling contest datasets (Vaihingen and Potsdam) which contain RGB and near IR bands, and they also used NDVI to identify vegetation area in pre-processing. The guided filter was used to smoothen the extracted building edges. Those research works developed or modified existing state-of-the-art deep learning algorithms to get better performance. The algorithms are based on CNNs and they used thousands of training samples. The trained deep learning models were tested on the validating datasets which were apparently from the same area with the training images.

Automatic building extraction based on image processing techniques requires multiple datasets and multiple algorithms have to be combined together which can make the extraction process more complicated and longer. Machine learning algorithms require fewer steps but expert knowledge for feature engineering is the most important part. Deep learning algorithms give better accuracy but require a lot more training samples than machine learning algorithms. Automatic digitization requires very high resolution satellite image, and they are expensive. Thus, feasibility of training data is one of the important issues for applications of deep learning in building footprint extraction. Other research works used deep learning algorithms that are based on CNN. Training CNN-based algorithms demand a lot of labelled data. And our previous works (Hein Thura Aung et al. 2018a, 2018b) indicate that validation on the images of the study area after being trained with images of external open source training datasets from different locations cannot get as good extraction results as using the training images from the same study area. This is because the nature of buildings in Yangon City, roof colours, building shapes, etc. are different from those in external training datasets. Thus, it is very important to find the balance between minimum number of training images and optimum extraction results. Conditional generative adversarial networks (CGANs) are based on unsupervised learning, and they have two neural nets trying to fool each other; the generator and the discriminator (Mirza and Osindero 2014). The generator can generate fake samples and the discriminator distinguishes them whether they are fake or real. If the discriminator can catch fake samples, the generator tries to generate the samples that look more real. In this way, both the discriminator and the generator can learn features from the inputs. Those CGANs are therefore more suitable for remote sensing applications where availability of training samples is very limited. Thus, pix2pix which is based on CGAN is chosen for building footprint extraction.

Evaluation methods are also very important as they indicate how well an algorithm can do the extraction task. Evaluation can be based on either pixel-based evaluation or object-based evaluation. Pixel-based evaluation is more objective as it considers based on the status of every pixel of which class it belongs to (Zerrouki and Bouchaffra, 2014). However, object-based evaluation is more appropriate if spatial information, shape and texture, etc., is to be considered. In building footprint extraction tasks, both pixel-based and object-based evaluation are used. Some building footprint extraction tasks as in Kabolizade et al. (2010), Koc-San and Turker (2012) and Vakalopoulou et al. (2015) were based on image segmentation, and they considered spatial information. Such kinds of system used object-based evaluation. However, as deep learning algorithms consider both low-level features such as lines and dots, and high-level features such as shape, both pixel-based and object-based evaluation were used in those above research works.

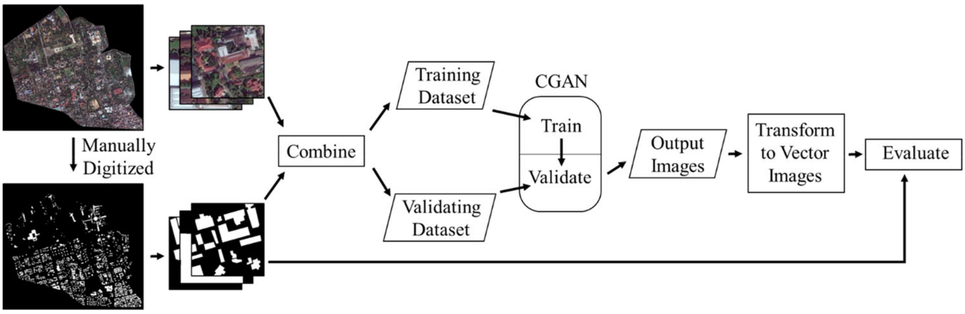


Figure 1. Building footprint extraction system.

As Yangon is the largest city in Myanmar and still under development, it is very important to have a reliable building map which can serve as a base map for further urban related applications. Current building footprint extraction is based on manual digitization which requires a lot of time and human resources. This research is part of Science and Technology Research Partnership for Sustainable Development (SATREPS Project) which is the collaboration between University of Tokyo, Japan and Yangon Technological University and other government organizations in Myanmar ('Development of a Comprehensive Disaster Resilience System and Collaboration Platform in Myanmar | SATREPS (Science and Technology Research Partnership for Sustainable Development)', n.d.). The project area is Yangon and Bago region where a comprehensive disaster resilience system will be established through disaster risk assessment and preparation. In other research works, deep learning algorithms were trained on thousands of labelled data, and multiple spectral bands, RGB and near, IR etc. However, this research focuses on automatic building footprint extraction in Yangon City with minimum requirement of spectral bands and dataset. The buildings in Yangon City have very different spatial properties and spectral reflectance from those images in freely available datasets. As a result, only monocular optical RGB satellite image of the study area from GeoEYE satellite is used for both training and validating, and training parameters are well tuned in order to get maximum extraction performance with minimum number of training images in the dataset. Pix2pix (Isola et al. 2017) is the deep learning algorithm that is used for building footprint extraction. The objectives of this research work are

- to create an automatic building extraction system for creation of Yangon map,
- to reduce the requirement of different algorithms and datasets for building extraction, and
- to investigate the effects of parameters in pix2pix to its performance.

2. Methodology

Figure 1 shows building footprint extraction system using Conditional Generative Adversarial Network (CGAN) with pix2pix. First, the satellite image of the study area is manually digitized in order to create a vector image of the area. Both raster and vector images are then sliced into multiple images and combined together in order to create training dataset and validating dataset. Then, the pix2pix is trained with the training dataset. After that, the trained pix2pix model will be validated by using the validating dataset. Before the evaluation process, the output images from the validation process are transformed into vector images, and then evaluated by using the manually digitized polygons.

2.1. Conditional generative adversarial network (CGAN) with pix2pix

In this building footprint extraction system, pix2pix is the deep learning algorithm that is going to be used. It is based on conditional generative adversarial network (CGAN) (Isola et al. 2017). The CGANs are based on generative adversarial network (GAN). The CGAN also consists of two neural networks; the generator and the discriminator which perform the same tasks with those in GAN. Based on the loss from the discriminator, the generator generates better samples and tries to make fool the discriminator.

In CGAN, both the generator and the discriminator are conditioned on additional input which can be a class label (Mirza and Osindero 2014). The CGANs are trained according to the objective function shown in Equation (1):

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim P_{data}(x)}\left[\log\left(D\left(x|y\right)\right)\right] + \mathbb{E}_{x\sim P_z(x)}\left[\log\left(1-D\left(G\left(z|y\right)\right)\right)\right] \tag{1}$$

$\mathbb{E}$ represents the expectation. $P_{data}(x)$ is the distribution of sample x and $P_z(x)$ is the distribution of z. $D(x|y)$ is the probability of x sampled from the real data. $G(z|y)$ represents the sample generated from z. 'y' represents the additional input that is used to condition the generator and the discriminator. The first term of Equation (1) is the expectation of the probability that the sample is real and the second term refers to the expectation of the probability that the sample comes from the generator. The discriminator has to distinguish the sample correctly so it has to maximize the function whereas the generator has to fool the discriminator so it has to minimize the first term of Equation (1).

2.2. Study area

The study area is the Dagon Township in Yangon City, Myanmar located between $16^{\circ}48'17.5752''N$ and $16^{\circ}46^{'}9160''N$, and $96^{\circ}07'5504''E$ and $96^{\circ}09'8423''E$. Monocular GeoEYE optical RGB image with 0.5 m spatial resolution is used for both training and validating. The original satellite image has the dimension of 5869 × 5248 and is sliced into 329 images of 256 × 256 pixels, all of which cover around 5.4 km^2. The sliced images are randomly chosen for the training dataset that includes 264 sliced images which covers about 4.3 km^2 (80% of total area) and the validating dataset that includes 65 images which covers around 1.1 km^2 (20% of total area) by using the split function in pix2pix.

In Hein Thura Aung et al. (2018a), the images from Google Map were used for training and the images of the study area were used for validating. In that work, the training images have 1 m spatial resolution, and the image location is New York City (affinelayer 2017), however the validating images have 0.5 m spatial resolution which was taken over Yangon City. According to the results of that work, as the spatial properties and spectral reflectance of training images are different from those of the validating images, the extraction results are poor. In Hein Thura Aung et al. (2018b), two training datasets were used; one of which contain 1850 images of New York from Google Map, and the other contains 265 images from the study area. In that research work, two validated results were compared, and the pix2pix trained with images from the study area had better performance in terms of F1 score although it was trained with only much fewer images. So, in this work, both training and validating datasets are created with the images of the study area.

Depending on the visualization, the images in the validating dataset are considered in four different types for evaluation;

Trees: images mostly covered with trees (23 images)

Buildings: images mostly covered with buildings (4 images)

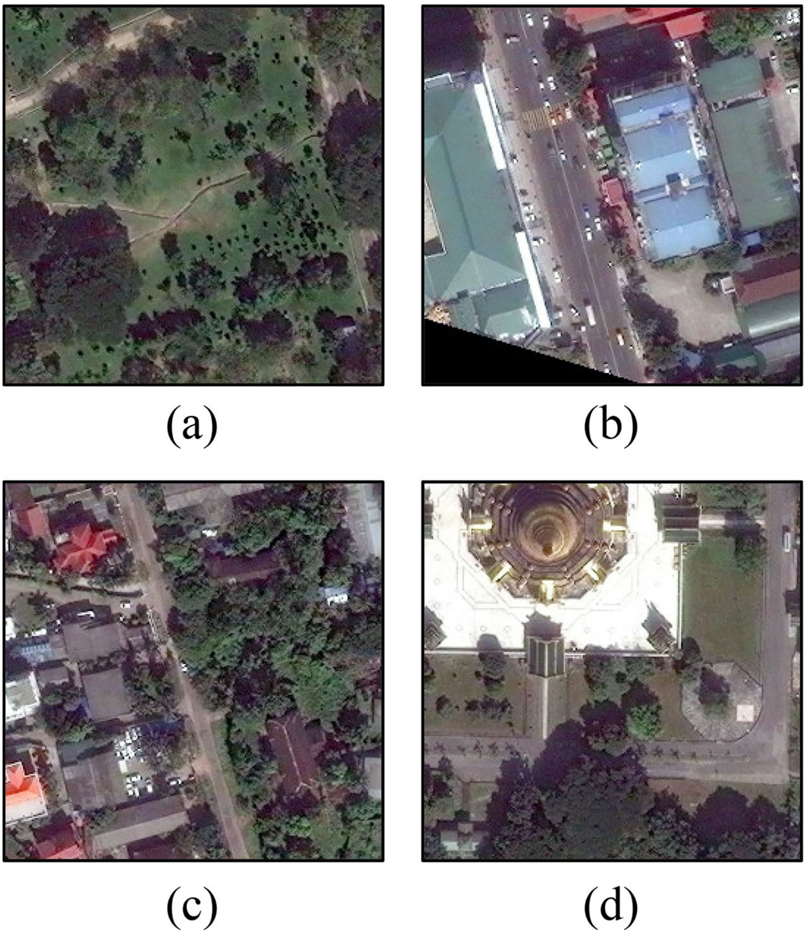


Figure 2. Example images of validating dataset: (a) Trees, (b) Buildings, (c) Mixed, (d) Pagodas.

Mixed: images evenly covered with trees and buildings (32 images)

Pagodas: images with pagodas (6 images)

Pagodas are usually golden structures with open space floors covered with white ceramic tiles. They have different spectral properties from other types of buildings and are found all over Yangon City and the country. Only the golden structures are regarded as building footprint areas and white ceramic floors are regarded as non-building footprint areas. In this research work, they are considered as one type of buildings. Amongst those four types, 'Buildings' and 'Mixed' types are considered as the most important ones because this research work aims to automate or speed up building footprint delineation process, and these two types are most commonly found in urban or suburban areas of Yangon city. Figure 2 shows sample images in validating dataset.

Table 1. Eight pix2pix models with different training parameters.

Model name	Epochs	Learning rate	β_1 (Adam)	NGF	NDF
A1	150	0.0002	0.5	64	64
A2	150	0.0001	0.5	64	64
A3	150	0.00002	0.5	64	64
B1	150	0.0002	0.9	64	64
B2	150	0.0002	0.1	64	64
C1	100	0.00002	0.9	128	128
C2	100	0.00002	0.5	128	128
C3	100	0.00002	0.1	128	128

2.3. Choosing training parameters

Training pix2pix is done using the workstation with NVIDIA Geforce GTX 1080 GPU, Ubuntu 18.04 and Python 3.6 and the tensorflow version of pix2pix is used. In Hein Thura Aung et al. (2018a), pix2pix was trained with different number of epochs and different training images from Google Map satellite images. Then, it was evaluated with the image of the study area. In Hein Thura Aung et al. (2018b), pix2pix was trained with two different training datasets of two different locations; one with the image from New York City and the other from the study area. The performance of two different trained pix2pix was then compared. In this work, based on three training parameters of pix2pix; learning rate (lr), beta 1(β_1) momentum of Adam optimization algorithm in pix2pix and number of filters in first convolution layer of the generator (NGF) and discriminator (NDF) of pix2pix, eight different pix2pix models are proposed as shown in Table 1.

Learning rate is the size of the step that pix2pix takes to its parameters (Ian Goodfellow et al. 2016) after each iteration. The usual way of finding appropriate learning rate is to start with a higher value and then reduce until the most appropriate value is obtained (Ian Goodfellow et al. 2016). The most appropriate learning rate gives minimum training loss which in turn can give better representation results. Pix2pix uses Adam algorithm which is adaptive learning rate optimization algorithm, and the default initial value of learning rate for pix2pix is 0.0002. However, that value is the recommended value by original researchers (Isola et al. 2017) for all image translation tasks. As building footprint extraction is the only focus in this work, the learning rate is changed to the most appropriate value that can give minimum training loss for our application. If the learning rate is too high, the training never gets to the minimum loss and oscillates. If the learning rate is too low, the training time will take longer to get to the minimum training loss (Ian Goodfellow et al. 2016).

Beta 1 (β_1) is the momentum value used in Adam optimization algorithm in pix2pix. Optimization is one of the most important parts in deep learning algorithms where the parameter values in layers of deep neural network are updated according to the training loss and gradient value in each epoch. During minimization of training loss, there can be local minimum points before getting to the global minimum point. If the training is stuck at local minimum points, it cannot go into any direction in order to reduce the loss. Momentum is used to overcome this problem. During optimization, optimization algorithms with momentum update the parameter values using not only the current gradient value but also the previous gradient values by multiplying them with a scale factor and adding to the current gradient value (Glassner n.d.). Thus, momentums smoothen and accelerate training, and it is analogous to the ball rolling down the training loss curve. If the momentum is too low, it cannot pass the local minimum point, but if it is too high, it will also pass the global minimum point (Glassner n.d.).

Deep learning networks are composed of convolutional layers, and filters in each layer are used for convolution of the input image during representation or training. The filters are applied to the image by moving over the image, and specific features of the input image are represented to the convolved map or feature map. Thus, filters are also called feature detectors (Glassner n.d.). As layers are connected in deep learning networks, outputs of one filter set become inputs of another filter set, and complex features of the input image are learnt in this way. The number of parameters in the network depends on the number of filters being used. The more filters being used, the more parameters the network will have, and the longer the training time will be. The values of the filters or feature detectors are automatically tuned through the learning process until the representation matches with the target image with minimum loss. As pix2pix contains two networks; the generator and the discriminator, the number of filters in the first convolution layer of both networks can be set.

In Table 1, for A1, A2 and A3, the values of β_1, NGF and NDF are default setting values of pix2pix and the number of epochs is set as 150. In Hein Thura Aung et al. (2018a), the number of epochs was adjusted, and how different number of epochs affect the performance was investigated. According to those results, high number of epochs can give more sharpness at building boundaries and more correct extraction results. However, higher number of epochs will take longer training time, and if the training gets to the global minimum point before finishing all number of epochs, subsequent training is useless. And, if the number of epochs is low, the training will not get to the global minimum point. Previous research works Hein Thura Aung et al. (2018a) also indicate if the number of epochs is too low, the building boundaries are not sharp enough, and if the epochs is too high, the training time takes longer. So, in this research work, the appropriate epoch value is checked by first initializing with an arbitrary number for example 100, and monitoring the exponential training loss curve by using Tensorboard. If the training loss curve still shows downward trend when the given epoch value is finished, that means minimum training loss has not reached and number of epochs should be increased. For those three models, only learning rate value is varied.

There are infinitely possible values for learning rate starting from the recommended value for pix2pix, 0.0002. However, in this research work, only three values are chosen based on visual monitoring of the curves of training loss by using Tensorboard in real time. The learning rate values are the default value, 0.0001 which is slightly smaller value, and 0.00002 which is one decimal digit smaller than the default value. If the learning rate is too small, the curve of training loss goes down very slowly, and if the learning rate is too large, it is very difficult to get to the minimum training loss. Choosing learning rate value which is too low also takes more time to get to the minimum training loss (Ian Goodfellow et al. 2016). Beta 1 (β_1) is one of the two exponential decay rates for Adam stochastic optimization algorithm in pix2pix and ranged from 0 to 1 (Kingma and Ba 2014). The recommended value of β_1 for pix2pix by original research paper of pix2pix is 0.5 (P. Isola et al. 2017), and 0.9 is the default value for tensorflow ('AdamOptimizer', n.d.). But in this work, 0.1 is also used to train pix2pix in order to investigate the effect of decreasing β_1 value and training with the lowest β_1 value for the study area.

The default values of the number of filters in the first convolution layer of the generator (NGF) and the discriminator (NDF) by the original research paper are both 64 (Isola et al. 2017). The maximum dimension of the input image to the pix2pix has to be 256 × 256 (Isola et al. 2017). In this work, for C1, C2 and C3 as shown in Table 1, the values of NGF and NDF are increased to 128, so that more information of the input images is captured in the first convolution layer. The input image to the pix2pix is 256 × 256 which

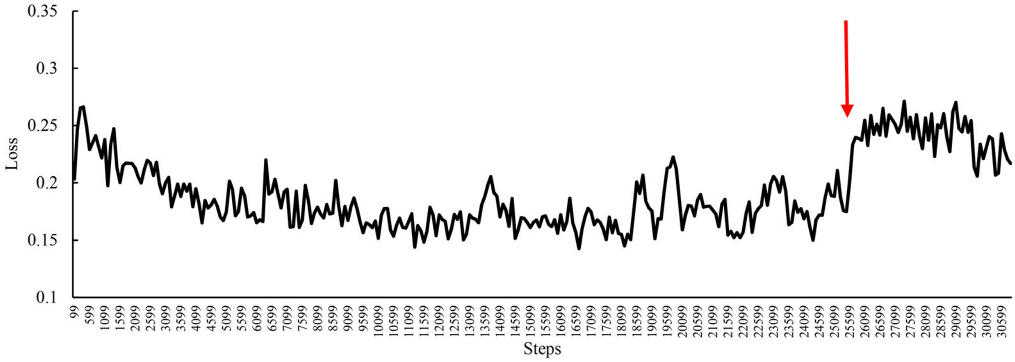


Figure 3. Loss of pix2pix trained with epoch 150, learning rate 0.0002 and NGF, NDF 128.

covers 128 × 128 squared metre. The default value 64 × 64 covers 32 × 32 squared metre in which case, some large buildings are not fully covered in the image. This could create inconsistencies between the images when they are combined together to form the whole map but the training time becomes faster. However, if the values of NGF/NDF are too high, the training time will be increased too much, but large buildings will be fully covered in one image. For choosing appropriate values of other parameters for increased NGF/NDF, pix2pix is first trained with learning rate 0.0002, NGF and NDF both 128, epochs 150 and default values for other parameters. As shown in Figure 3, the loss suddenly increased at around step 26,000, which did not happen during training other models with NGF/NDF 64 and epochs 150. According to the training graph from Tensorboard, the learning rate and epoch are decreased to 0.00002 and 100. Similarly, all other parameter tuning processes are done by visual interpretation of training loss graph using Tensorboard.

2.4. Accuracy assessment

Figure 4 shows two possible building areas after validating process; rectangular polygon as the ground truth footprint and oval polygon as the extracted footprint. As shown in Figure 4, from the overlap of two building footprints, three different area types are possible; true positive (TP) area where the extracted area overlays with the ground truth area, false negative (FN) area where part of the ground truth area is missed to be extracted, and false positive (FP) area where part of the extracted area does not exist in the ground truth area.

The output images of validation with pix2pix are in raster format with 256 × 256 dimension. However, manually digitized shape file is in vector format, and in order to make comparison for validation, the output images are transformed into vector format. After that, they are compared with manually digitized shape files, and three possible area types are determined. Figure 5 shows how the comparison is made, and the area types are determined. After the area types are determined, the number of pixels in each area type is calculated based on the area in one pixel. As the spatial resolution is 0.5 m, there is 0.25 m^2 in one pixel.

Inspired by the work in Gavankar and Ghosh (2018) and Rutzinger et al. (2009), the pixel-based performance of all pix2pix models are evaluated with three factors; completeness, correctness and F1 measure which are defined in Equation (2). The values of

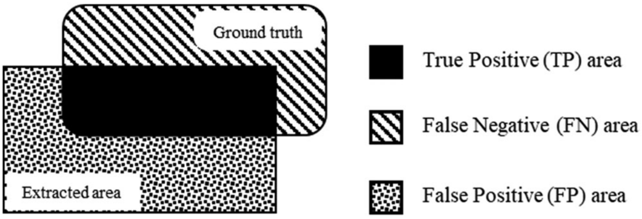


Figure 4. Three possible area types (Shan and Lee 2005).

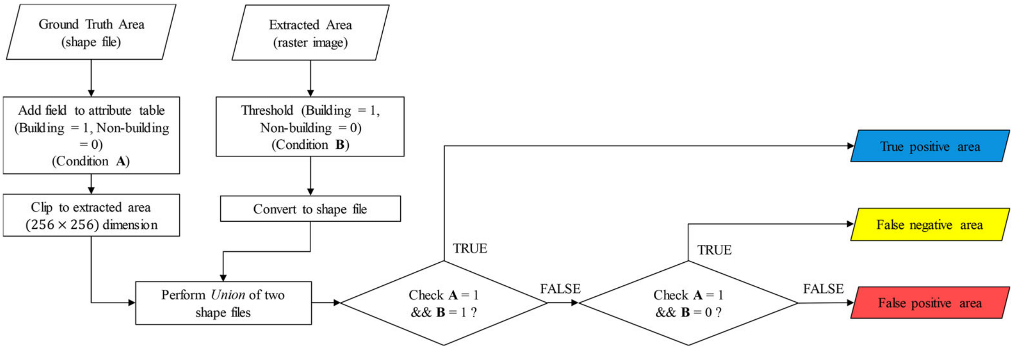

Ground Truth Area (shape file)
Extracted Area (raster image)
Add field to attribute table (Building = 1, Non-building = 0) (Condition A)
Threshold (Building = 1, Non-building = 0) (Condition B)
Clip to extracted area (256 × 256) dimension
Convert to shape file
Perform Union of two shape files
Check A = 1 && B = 1 ?
TRUE
FALSE
Check A = 1 && B = 0 ?
TRUE
FALSE
True positive area
False negative area
False positive area

Figure 5. Flow chart of evaluation process.

completeness, correctness and F1 are between 0 and 1. The larger the number, the better the performance of the model.

$$Completeness(Comp) = \frac{TP}{TP + FN}, Correctness(Corr) = \frac{TP}{TP + FP}, F1 = \frac{2TP}{2TP + FN + FP}. \tag{2}$$

The completeness is the ratio between true positive pixels and the sum of true positive and false negative pixels, and the correctness is the ratio between true positive pixels and the sum of true positive and false positive pixels. However, F1 score considers both false negative and false positive pixels. Completeness, correctness and F1 score are used in order to evaluate how far a pix2pix model can completely and correctly extracts building footprints. In this evaluation process, no overlap threshold is considered.

3. Results and analysis

In this section, the output images of all pix2pix models that are trained with eight different configurations are compared with manually digitized polygons and the compared results are analyzed.

3.1. Training results

Table 2 shows model name, training configuration, training loss and training time for all eight models. As shown in Table 2, all models converge to the loss value around 0.15 although they have different training parameter values. Pix2pix models with higher NGF and NDF, C1, C2 and C3 models, have lower training loss around 0.12–0.13. As a result of increasing NGF/NDF, more information is represented between the input image and

Table 2. Model name, training configuration, training loss and training time for all eight pix2pix models.

Model	Training configuration (default unless stated)	Training loss	Training time (min)
A1	NGF/NDF 64, lr 0.0002, β_1 0.5, epochs 150	$\approx$ 0.15	76
A2	NGF/NDF 64, lr 0.0001, β_1 0.5, epochs 150	$\approx$ 0.15	64
A3	NGF/NDF 64, lr 0.00002, β_1 0.5, epochs 150	$\approx$ 0.15	63
B1	NGF/NDF 64, lr 0.0002, β_1 0.9, epochs 150	$\approx$ 0.15	71
B2	NGF/NDF 64, lr 0.0002, β_1 0.1, epochs 150	$\approx$ 0.15	69
C1	NGF/NDF 128, lr 0.00002, β_1 0.9, epochs 100	$\approx$ 0.12	192
C2	NGF/NDF 128, lr 0.00002, β_1 0.5, epochs 100	$\approx$ 0.12	193
C3	NGF/NDF 128, lr 0.00002, β_1 0.1, epochs 100	$\approx$ 0.12	194

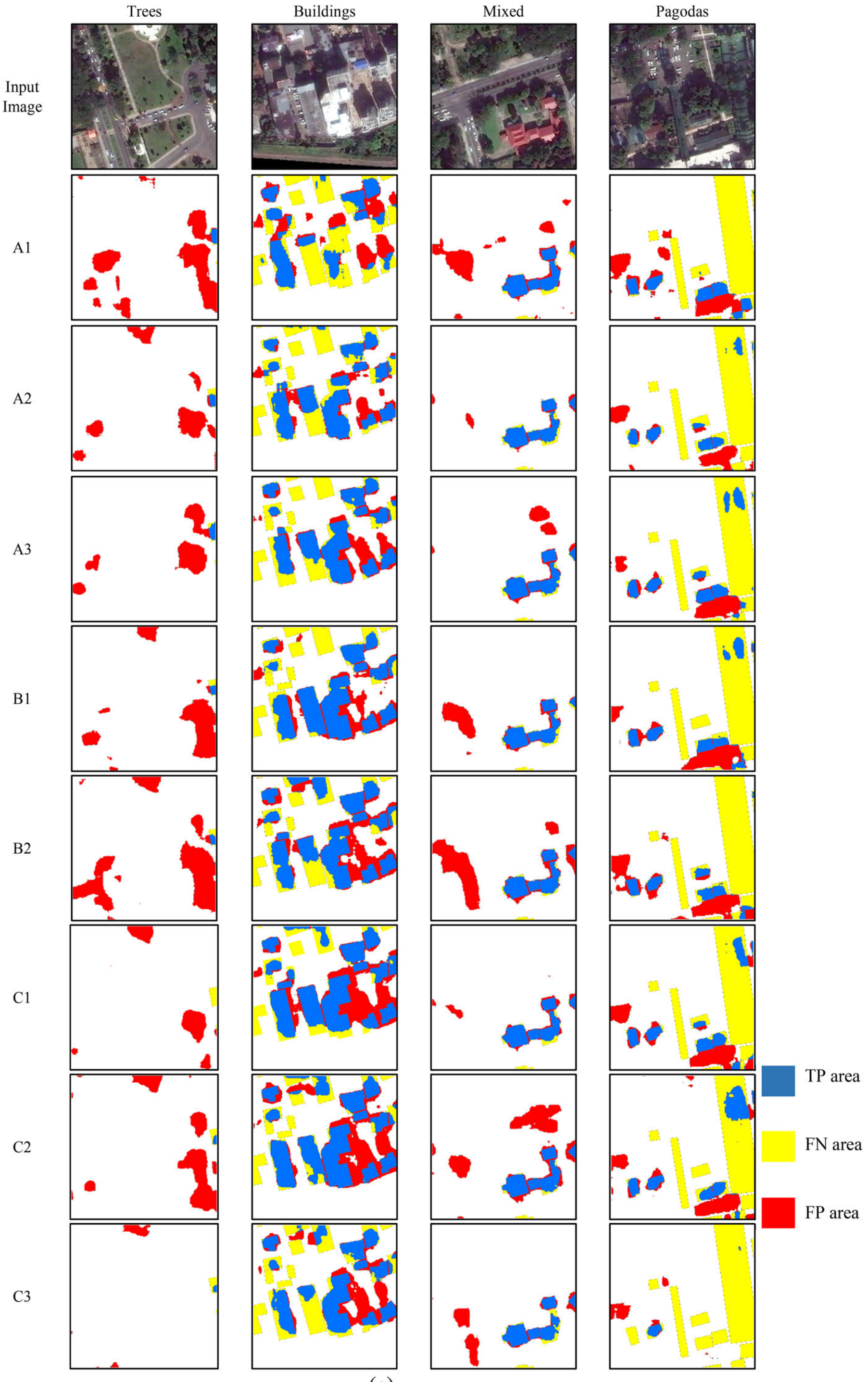
the target image, and the training loss is reduced. Moreover, as shown in Table 2, changing learning rates and β_1 values does not have much effect on the training time. However, increasing NGF/NDF twice also increases the training time about three times even though the maximum epoch number is reduced.

3.2. Validating results

In the evaluation process, the validated output images are transformed into vector images and compared with manually digitized polygon images. As described in section 2.2, the images in validating dataset are considered in four different groups depending on how much trees and buildings are distributed in the images. Figure 6(a,b) shows selected input images from each type and their corresponding output images of all eight pix2pix models. Blue area represents true positive areas, yellow area false negative areas and red area false positive areas.

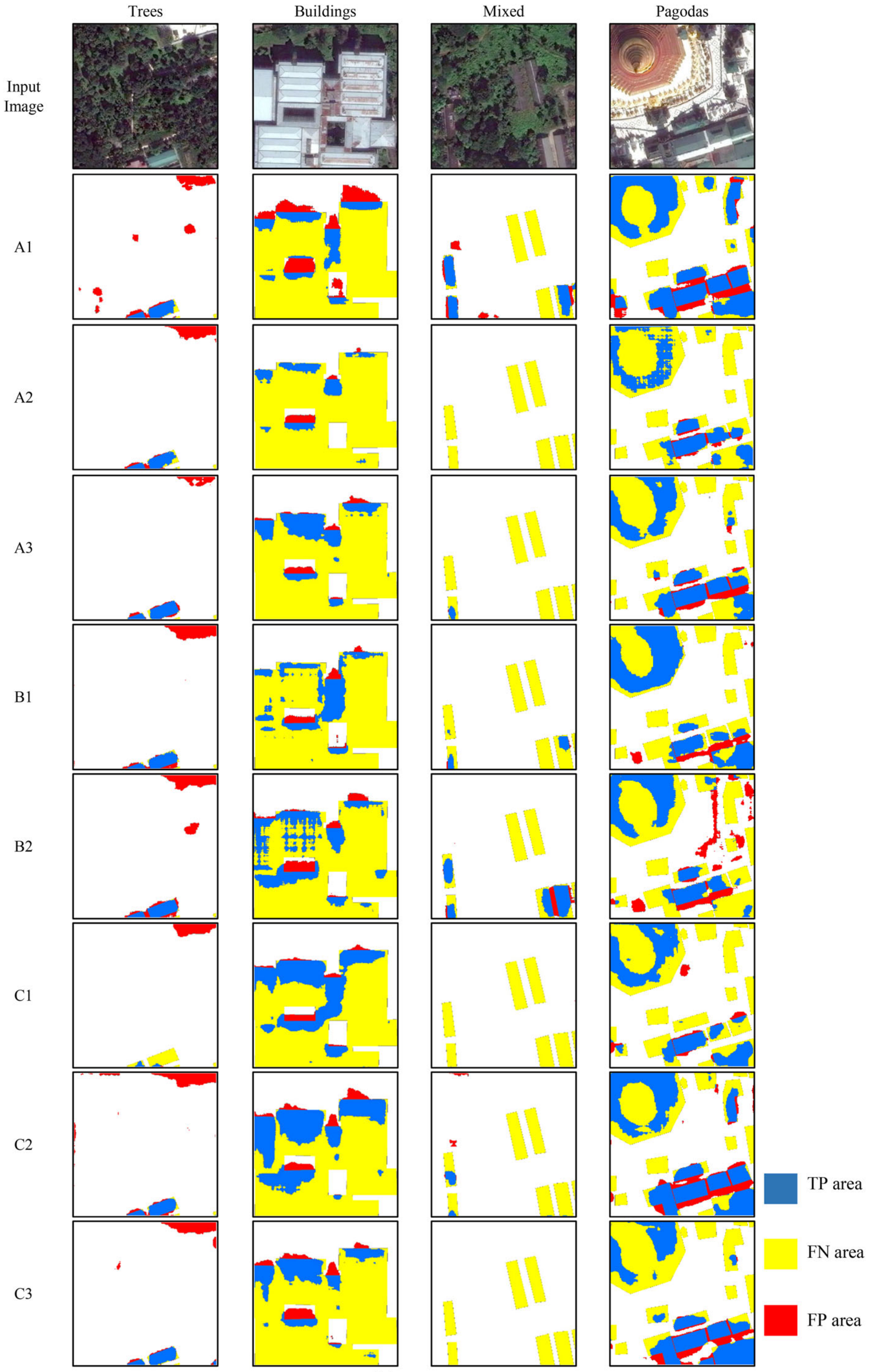
As shown in Figure 6(a), the input image for ‘Trees’ contains trees and asphalt areas. Only a small building footprint area exists on the right of the image. All pix2pix models except for C3 incorrectly extract asphalt regions as building footprints. Moreover, at the top of the image, there is a small impervious concrete-like area and all pix2pix models except for A3 incorrectly extract that region as building footprints. However, all pix2pix models except for C1 are able to extract part of the small building footprint area on the right of the image. The input image for “Buildings” in Figure 6(a) shows complex urban patterns which mainly involves shadows, buildings with whitish, bluish, rusty and concrete-colour rooftops, vehicles and some trees. All eight pix2pix models incorrectly extract shadow regions as building footprint areas. C2 detects almost all or part of the buildings with multiple colour rooftops although it misses buildings with small footprint areas. As shown in the image for ‘Trees’, one of the buildings in the image of ‘Buildings’ has similar rooftop colour and all pix2pix models correctly detect that building.

As shown in Figure 6(a), the image of ‘Mixed’ contains a large building with reddish rooftop. Part of the building is also occluded by trees and there are some shadow on the roof. However, all eight pix2pix models extract almost the whole part of the buildings although they have some false positive extraction on asphalt regions. There are two parts of asphalt region but the models misclassify on the region which also contains many white vehicles. This is similar to the incorrect classification in the image of ‘Trees’. There is also a very small building footprint at the bottom but all models miss that area. The image for ‘Pagodas’ also contains complex objects such as buildings with dark greenish and rusty rooftops, shadows and trees, white vehicles and white impervious floor which is part of the Great Shwedagon Pagoda. In the output images of all eight pix2pix models, only very small or no part of the buildings with dark greenish rooftops are extracted although they are able to extract the buildings with rusty rooftops on the left. The models



(a)

Figure 6. Selected sample input images and their corresponding output images of all eight pix2pix models.



(b)

Figure 6. Continued.

Table 3. Average Completeness, Correctness, and F1 Score for four image types validated by eight pix2pix models (highest values in bold and lowest values in italic).

Image Types	Evaluation	A1	A2	A3	B1	B2	C1	C2	C3
Trees	Comp	**0.5485**	0.4162	0.4700	0.4221	0.5121	*0.2922*	0.5114	0.4220
	Corr	*0.0961*	**0.3268**	0.1736	0.2491	0.1126	0.2424	0.1360	0.1524
	F1	*0.0955*	**0.2174**	0.1661	0.1916	0.1254	0.1632	0.1135	0.1633
Buildings	Comp	0.4061	*0.3758*	0.4622	0.5758	0.5390	0.5815	**0.6390**	0.4733
	Corr	*0.6380*	0.8143	**0.8463**	0.8288	0.8106	0.8021	0.8189	0.7837
	F1	*0.4825*	0.4814	0.5741	0.6422	0.6211	0.6445	**0.6943**	0.5675
Mixed	Comp	0.6298	*0.5012*	0.5499	0.5856	0.6817	0.5610	**0.7154**	0.5706
	Corr	*0.6528*	**0.8143**	0.7868	0.7613	0.6993	0.7652	0.7162	0.7885
	F1	0.6263	*0.5950*	0.6219	0.6379	0.6730	0.6224	**0.6943**	0.6270
Pagodas	Comp	0.4187	0.3234	0.3678	0.4033	0.3677	0.3731	**0.4265**	*0.2733*
	Corr	**0.5773**	0.5616	0.5566	0.5074	0.4698	0.5365	0.5004	*0.4667*
	F1	**0.4757**	0.3964	0.4322	0.4288	0.3999	0.4113	0.4482	*0.3222*

Table 4. Average completeness, correctness and F1 values of all images by eight pix2pix models.

Image types	Evaluation	A1	A2	A3	B1	B2	C1	C2	C3
All images	Comp	0.5008	0.4042	0.4625	0.4967	0.5251	0.4520	**0.5731**	0.4348
	Corr	0.4911	**0.6293**	0.5908	0.5866	0.5231	0.5865	0.5429	0.5478
	F1	0.4200	0.4226	0.4486	0.4751	0.4549	0.4603	**0.4876**	0.4200

The highest values are presented in bold.

also incorrectly extract the whitish floor of the pagoda as building footprints which is similar to the image of 'Trees'. C3 detects no or very small part of all buildings in the image except for one small building.

As shown in Figure 6(b), the image for 'Trees' mainly contains trees and vegetation, and two small buildings with greenish and rusty rooftops. Similarly in the image of 'Trees' in Figure 6(a), the whitish area at the upper right corner of the image is misclassified as building footprint regions by all eight pix2pix models. Unlike the image of 'Pagodas' in Figure 6(a), the building with greenish rooftop is extracted by all pix2pix models except for C1. This might be due to the lighter greenish colour of the rooftop. As there is no asphalt region, there is low false positive area except for those in the upper right corner of the image. For the image of 'Buildings' in Figure 6(b), the whole image is almost covered by a large building with a whitish rooftop. There are also two holes inside the rooftop region. By looking at the output images of the image of 'Buildings' in Figure 6(b), although the spectral properties of the rooftop region look similar, all eight models can only extract the upper part of the whole rooftop whilst they misclassify the hole and some part of shadow region as building footprints especially for A1. As the GeoEYE image of the study area is monocular, side of the wall of the hole is seen from the top view and most of the models except for A1 can distinguish between the shadow of the hole and side of the wall.

As shown in Figure 6(b), the image of 'Mixed' is with six buildings with rusty rooftops and trees. The spectral properties of the rooftops seem similar except for them at the lower right corner of the image. By looking at the output images, only A1 and B2 can extract the buildings at the bottom right corner but the other pix2pix models miss to extract. A1 and B2 are also able to extract the buildings which are partially occluded by trees at the left of the image but they have more false positive areas. The false positive area of A1 at the upper left part of the output image is actually building footprint region which is not included in the ground truth as it is too small. As shown in Figure 6(b), the image of 'Pagodas' includes the Shwedagon Pagoda which is a golden structure with white ceramic floors and surrounded by the buildings with greenish rooftops. Unlike the image

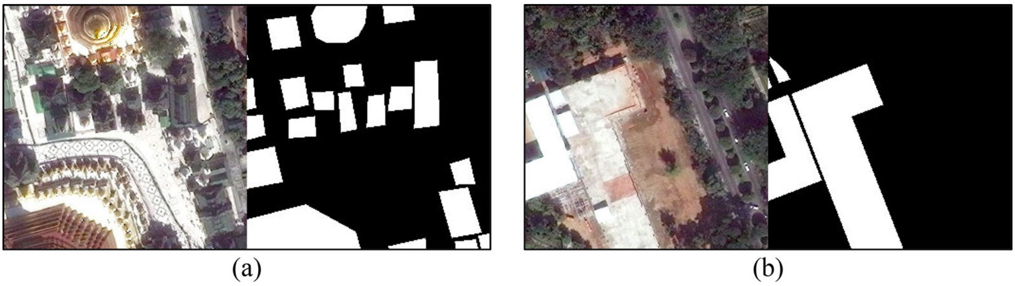


Figure 7. Two example images in the training dataset which have inconsistent representation of building footprint areas.

of 'Pagodas' in Figure 6(a), the buildings with greenish rooftops are properly extracted especially by A1 and C2. This might be because of the whitish ceramic floor that is surrounding the buildings with greenish rooftops which makes them distinct objects from its environment. The central part and outer part of the golden structure are missed to be extracted by the models although they have similar spectral properties.

Table 3 shows the average completeness, correctness and F1 score of all images in each four different types for all eight pix2pix models. All three evaluating parameters, completeness, correctness and F1 score, are compared for all four different types of images; Trees, Buildings, Mixed and Pagodas. The highest completeness value of all four different types is at 'Mixed' images from C2 at the value of 0.7154 and the highest correctness value of all four different types is at 'Buildings' from A3 with 0.8463. The highest F1 value of all different types ties at 0.6943 between 'Buildings' and 'Mixed' images both from C2.

The lowest F1 value of all four types is at 'Trees' images with a value of 0.2174. The lowest correctness value of all different image types is 0.0961 is also in this type from A1. All images in the 'Trees' images are mostly covered with trees. As trees are major part of the images, there are only few building footprints. All models except for C1 are able to detect more than 40% of the ground truth for 'Trees' images but they have very high false positive areas. This high false positive areas make very low correctness values for the models. The lowest completeness value of all four different image types is in 'Pagodas' images with 0.2733 from C3. As shown in Table 3, C2 has the highest average completeness values in 'Building' images and 'Mixed' images. C2 also has high correctness values in both different image types which make it have the highest average F1 values in 'Building' images and 'Mixed' images. A2 has the highest average correctness and F1 values for 'Trees' images, highest average correctness for 'Mixed' images, and high correctness values in other types, but it shows the lowest performance for completeness for 'Building' and 'Mixed' images.

As shown in Table 3, for 'Trees' images, A1 has the highest completeness value with 0.5485 and A2 has the highest correctness value with 0.3268 and highest F1 value with 0.2174. All eight pix2pix models show very poor performance for the images in 'Trees'. For 'Buildings' images, all eight models except for A2 have average completeness of more than 0.4 and C2 has the highest value up to 0.63. Correctness values are higher than completeness values with up to 0.84 by A3 and the lowest average correctness value is 0.63 by A1. For 'Mixed' images, the average completeness is the highest amongst the other image types with up to 0.71 by C2 and the average correctness values are higher than the completeness values with the highest value up to 0.81 by A2. As in Table 3, for 'Pagodas' images, C2 has the highest average completeness value with up to 0.42 and A1 has the highest average correctness and F1 values with 0.57 and 0.47 respectively.

Table 4 shows the average values of completeness, correctness and F1 values of all 65 images in the validating dataset from all eight pix2pix models. As shown in Table 4, C2 has the best average completeness and F1 values with 0.57 and 0.48, respectively, and high average correctness values with 0.5429 and is the best pix2pix setup of all eight models. The highest average correctness value is 0.6293 by A2 and the lowest average correctness is 0.4911 by A1. B1 has average F1 value of 0.4751 and becomes the second best pix2pix model.

4. Discussion

In this work, three training parameters of pix2pix, learning rate, β_1 and number of filters in the first convolution layers of the generator and the discriminator (NGF/NDF) are varied in order to set up different training configurations. According to the results, both setting β_1 value at default and increasing NDF/NGF together which is C2 model can increase both completeness and correctness of pix2pix with default setting. Pix2pix with both increased and decreased β_1 values and higher NGF/NDF which are C1 and C3, have lower completeness but higher correctness than pix2pix with default value. However, the effect of changing β_1 values is different for B1, A1, B2 models and C1, C2, C3 models. For example, as shown in Table 4, decreasing β_1 from default value to 0.1 in A1, B2 models increases both completeness and correctness, but the effect is the opposite for C2, C3 models. Moreover, increasing β_1 from default value to 0.9 has similar effect for both B1, A1 and C1, C2 models i.e. decreased completeness and increased correctness (Table 4). According to the results in Table 4, changing training parameter values has effect on both completeness and correctness which make them increased or decreased. It can also be said that the models with decreased completeness values have increased correctness values; for example, A2, A3, B1, C1 and C3. The changed values of completeness and correctness in C2 and B2 are almost balanced.

According to the evaluation results in Figure 6(a,b), it can be seen that all pix2pix models have poor extraction rate for buildings with greenish rooftops. But, that depends on the objects around the buildings i.e. if the surrounding objects have different spectral properties, the detection rate is higher as shown in the image of ‘Pagodas’, Figure 6(b). The trained pix2pix models also have low extraction rate for buildings with rusty rooftops and buildings with small footprint areas. Especially for the buildings with rusty rooftops which are surrounded by trees and vegetation, the detection rate is very low as shown in the image of ‘Mixed’, Figure 6(b). The size of input image to pix2pix is set 256 × 256 in order to cover large buildings. More features can be learned in one image if larger input image size is used. But if smaller input image is used, the extraction for small buildings such as buildings in ‘Mixed’ group with rusty rooftops surrounded by trees can be better. However, small input image gives small output images and they can produce inconsistencies as discussed in section 2.3. Moreover, as shown in the images of ‘Buildings’ in Figure 6(a,b), some buildings with whitish rooftops are missed and some are extracted. This is due to inconsistent colour representation of building and non-building in the training datasets, i.e. the white ceramic floors of pagoda region are regarded as non-building but some buildings have rooftops with similar spectral properties of pagoda floors. This can be seen in Figure 7(a,b).

According to the results in Table 3, although the best pix2pix model C2 has 0.1135 and 0.4482 of F1 values for ‘Trees’ and ‘Pagodas’ image groups, it has 0.6943 of F1 values for both ‘Buildings’ and ‘Mixed’ image groups which are commonly found in urban and suburban areas. The reasons why the best pix2pix model C2 has only 0.4876 of F1 value

(Table 4) is because of low F1 values of 'Trees' and 'Pagodas' image groups. For the images in "Trees" group, there are only very few buildings in one image and a small amount of missed extraction or incorrect extraction gives a big impact on the performance (Table 3). And due to inconsistencies in the training images as shown in Figure 7, the pix2pix models show low completeness, correctness and F1 values for images in 'Pagodas'. As shown in Figure 7(a), pagodas are open structure without roofs, and their floors are made of whitish ceramic tiles. Their spatial and spectral properties are different from other buildings in the study area. Such poor extraction results on those images with pagodas have effect on overall performance. It can also be seen in Table 3 that for the model with high correctness, it has low completeness for example, for the case of A2 in 'Mixed' and A3 in 'Buildings'. In this research, F1 value is chosen to decide for the best performance but to make the choice between the models with better completeness and better correctness depends on the intended application of the map.

During training process, it takes around 420–480 min (7–8 h) by a single user with GIS software to create 1806 polygons of building boundaries in the study area which only covers one township of Yangon city. There are 45 townships in the whole Yangon city and it will take a lot of time if manual digitization is used. However, in this research, it only takes 193 min (3 h) to train pix2pix with 264 training images and it takes less than 1 min to validate 64 images in the validating dataset. So, pix2pix generates building footprint map of the study area in about 5 min whereas manual digitization takes around 420–480 min.

5. Conclusion

In this research work, several parameters of pix2pix are adjusted in order to analyze the performance of pix2pix from the changes and to get the best pix2pix configuration. Learning rate, β_1 term of Adam and number of filters in the first convolutional layer of the generator and the discriminator are chosen to test. According to the validating and evaluating results, pix2pix setup with increased NGF/NDF and learning rate and default β_1 has the best performance. In this work, the best performance is decided by using F1 score as it is the ratio of true positive areas to both false positive and false negative areas which means it considers both completeness and correctness altogether. Although three training parameters are changed in order to get the best performance, it is difficult to conclude the effect of changing each parameter. For example, it cannot be said decreasing β_1 increases the performance because changing β_1 from 0.9 to 0.5 decreases F1 score but it is opposite when β_1 is changed from 0.5 to 0.1. Although the performance of pix2pix is increased by increasing the number of filters twice, the training time also increases up to three times.

If the spectral properties of validating images are more similar to those of training images, the extraction results are better. As building rooftops of Yangon City have diverse colours such as concrete-like, reddish, greenish, bluish, whitish, rusty etc., and buildings are not constructed in systematic layout, training pix2pix with images from the study area gives better extraction results. For other urban areas which have buildings with similar spectral properties of Yangon City, our training parameter configurations can give similar results. For urban areas which have different spectral properties of Yangon City and in case they need to train with their own dataset, they can learn from our experience with adjusting training parameters and their effects, and do a jump start. Based on our experience, only the training process takes long time depending on the number of images in the training dataset and training parameter configurations, but once the training settings for best performance of pix2pix is configured and training process is done, the implementation is much faster than training process.

This research is still ongoing as part of SATREPS Project and only RGB bands of GeoEYE image are used to train pix2pix and validate the performance of trained pix2pix. The limitation of availability of the satellite image with multiple bands is one of the most crucial reasons for using only RGB bands. However, this is important for developing countries such as Myanmar which cannot afford much for VHR satellite image with RGB and infrared (IR) bands. Using the same satellite image for both training and validation, we also have plans to apply other deep learning algorithms such as fully convolutional network (FCN) which is for semantic segmentation and Glow which is another generative deep learning model, and compare with pix2pix. At this moment, we are also discussing with other government organization for collaboration such as Yangon City Development Committee (YCDC) which maintains Yangon City Database. And then, we can implement our method for the whole area of Yangon City and compare the results and time taken of our method with those of manual digitization.

In conclusion, this research work tries to reduce the cost of necessary datasets, and automate and speed up the digitization process in creation of urban building map by using conditional generative adversarial network (CGAN) and according to the results, pix2pix can show promising performance in building footprint extraction from only GeoEYE monocular optical satellite RGB image.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Author contributions

Conceptualization, Hein Thura Aung, Sao Hone Pha and Wataru Takeuchi.; Methodology, Hein Thura Aung.; Software, Hein Thura Aung.; Formal Analysis, Hein Thura Aung.; Investigation, Hein Thura Aung.; Resources, Wataru Takeuchi.; Writing – original draft preparation, Hein Thura Aung.; Writing – review and editing, Sao Hone Pha and Wataru Takeuchi.; Visualization, Hein Thura Aung.; Supervision, Sao Hone Pha and Wataru Takeuchi.

ORCID

Hein Thura Aung http://orcid.org/0000-0001-6265-9271

References

AdamOptimizer. n.d. TensorFlow [WWW Document]. [accessed 2019 Jan 25]. https://www.tensorflow.org/api_docs/python/tf/train/AdamOptimizer.

affinelayer 2017. pix2pix-tensorflow [WWW Document]. GitHub. [accessed 2019 Nov 30]. https://github.com/affinelayer/pix2pix-tensorflow.

Akçay HG, Aksoy S. 2010. Building detection using directional spatial constraint. Presented at the 2010 IEEE International Geoscience and Remote Sensing Symposium; July 25–30; Honolulu, HI, USA. IEEE. p. 1932–1935.

Aung HT, Pha SH, Takeuchi W. 2018a. Automatic building footprints extraction of Yangon City from GeoEYE monocular optical satellite image by using deep learning.Presented at the 39th Asian Conference on Remote Sensing, Kuala Lumpur, Malaysia. p. 1987–1996.

Aung HT, Pha SH, Takeuchi W. 2018b. Performance evaluation of building footprint delineation in Yangon City using deep learning with different training datasets. Presented at the 9th International Conference on Science and Engineering, Yangon Technological University, Yangon.

Bittner K, Adam F, Cui S, Korner M, Reinartz P. 2018. Building footprint extraction from VHR remote sensing images combined with normalized DSMs using fused fully convolutional networks. IEEE J Sel Top Appl Earth Observ Remote Sens. 11(8):2615–2629.

Chen K, Fu K, Gao X, Yan M, Sun X, Zhang H. 2017. Building extraction from remote sensing images with deep learning in a supervised manner. Presented at the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS); Fort Worth, TX. IEEE. p. 1672–1675.

Development of a Comprehensive Disaster Resilience System and Collaboration Platform in Myanmar | SATREPS (Science and Technology Research Partnership for Sustainable Development) [WWW Document]. n.d. [accessed 2019 July 14]. https://www.jst.go.jp/global/english/kadai/h2607_myanmar.html.

Gavankar NL, Ghosh SK. 2018. Automatic building footprint extraction from high-resolution satellite image using mathematical morphology. Eur J Remote Sens. 51(1):182–193.

Ghaffarian S, Ghaffarian S. 2014. Automatic building detection based on supervised classification using high resolution Google earth images. Int Arch Photogramm Remote Sens Spatial Inf Sci. XL-3: 101–106.

Ghaffarian S, Ghaffarian S. 2014. Automatic building detection based on Purposive FastICA (PFICA) algorithm using monocular high resolution Google Earth images. ISPRS J Photogramm Remote Sens. 97:152–159.

Glassner A. n.d. Deep learning. Vol. 1, From basics to practice. Amazon.com Services LLC.

Glassner A. n.d. Deep learning. Vol. 2, From basics to practice. Amazon.com Services LLC.

Goodfellow I, Bengio Y, Courville A. 2016. Deep learning. Cambridge (MA): MIT Press.

Huang Z, Cheng G, Wang H, Li H, Shi L, Pan C. 2016. Building extraction from multi-source remote sensing images via deep deconvolution neural networks. Presented at the IGARSS 2016 – 2016 IEEE International Geoscience and Remote Sensing Symposium; Beijing, China. IEEE. p. 1835–1838.

Isola P, Zhu JY, Zhou T, Efros AA. 2017. Image-to-image translation with conditional adversarial networks. Presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA. pp. 5967–5976.

Japan International Cooperation Agency (JICA). 2013. A strategic urban development plan of greater Yangon (No. Final Report 1).

Japan International Cooperation Agency (JICA). 2018. Maps of JICA Major Projects. Yangon, Myanmar: JICA.

Kabolizade M, Ebadi H, Ahmadi S. 2010. An improved snake model for automatic extraction of buildings from urban aerial images and LiDAR data. Comput Environ Urban Syst. 34(5):435–441.

Kingma DP, Ba J. 2014. Adam: a method for stochastic optimization. In: ArXiv:1412.6980 [Cs]. Presented at the International Conference on Learning Representations (ICLR 2015), San Dieo, USA.

Koc-San D, Turker M. 2012. A model-based approach for automatic building database updating from high-resolution space imagery. Int J Remote Sens . 33(13):4193–4218.

Liu W, Prinet V. 2005. Building detection from high-resolution satellite image using probability model. Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS'05; July 29; Seoul, South Korea. IEEE. p. 3888–3891.

Lopez ER, Santos HF, Ana RRCS, Samalburo SJD. 2015. Evaluation of building footprint generation from LIDAR data and orthoimages using object-based image analysis technique. Presented at the 36th Asian Conference on Remote Sensing, Manila, Philippines.

Ministry of Immigration and Population M. 2015. A changing population: Yangon region figures at a glance. Myanmar: Department of Population.

Mirza M, Osindero S. 2014. Conditional Generative Adversarial Nets. arXiv:1411.1784 [cs, stat].

Ok AO, Senaras C, Yuksel B. 2013. Automated detection of arbitrarily shaped buildings in complex environments from monocular VHR optical satellite imagery. IEEE Trans Geosci Remote Sens. 51(3): 1701–1717.

Oztimur Karadag O, Senaras C, Yarman Vural FT. 2015. Segmentation fusion for building detection using domain-specific information. IEEE J Sel Top Appl Earth Observ Remote Sens. 8(7):3305–3315.

Rutzinger M, Rottensteiner F, Pfeifer N. 2009. A comparison of evaluation techniques for building extraction from airborne laser scanning. IEEE J Sel Top Appl Earth Observ Remote Sens. 2(1):11–20.

Shan J, Lee SD. 2005. Quality of building extraction from IKONOS imagery. J Surv Eng. 131(1):27–32.

Singh G, Jouppi M, Zhang Z, Zakhor A. 2015. Shadow based building extraction from single satellite image. Presented at the SPIE/IS&T Electronic Imaging. San Francisco (CA): SPIE. p. 15.

Sirmacek B, Unsalan C. 2008. Building detection from aerial images using invariant color features and shadow information. Presented at the 2008 23rd International Symposium on Computer and Information Sciences; Istanbul. IEEE. p. 1–5.

Sritarapipat T, Takeuchi W, 2018. Land cover change simulations in Yangon under several scenarios of flood and earthquake vulnerabilities with master plan. J Disaster Res. 13(1):50–61.

United Nations. 2014. World's population increasingly urban with more than half living in urban areas [WWW Document]. Welcome to the United Nations. It's your world. http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html.

Vakalopoulou M, Karantzalos K, Komodakis N, Paragios N. 2015. Building detection in very high resolution multispectral data with deep learning features. Presented at the IGARSS 2015 – 2015 IEEE International Geoscience and Remote Sensing Symposium; Milan, Italy. IEEE. p. 1873–1876.

Xu Y, Wu L, Xie Z, Chen Z. 2018. Building extraction in very high resolution remote sensing imagery using deep learning and guided filters. Remote Sens. 10(1):144.

Yanfeng Wei ZZ. 2004. Urban building extraction from high-resolution satellite panchromatic image using clustering and edge detection. IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium; Anchorage, AK, USA. IEEE. p. 2008–2010.

Yang H, Wu P, Yao X, Wu Y, Wang B, Xu Y. 2018. Building extraction in very high resolution imagery by dense-attention networks. Remote Sens. 10(11):1768.

Yang HL, Yuan J, Lunga D, Laverdiere M, Rose A, Bhaduri B. 2018. Building extraction at scale using convolutional neural network: mapping of the United States. IEEE J Sel Top Appl Earth Observ Remote Sens. 11(8):2600–2614.

Zerrouki N, Bouchaffra D. 2014. Pixel-based or Object-based: which approach is more appropriate for remote sensing image classification? Presented at the 2014 IEEE International Conference on Systems, Man and Cybernetics – SMC; San Diego, CA, USA. IEEE. p. 864–869.

Zhang Q, Wang Y, Liu Q, Liu X, Wang W. 2016. CNN based suburban building detection using monocular high resolution Google Earth images.Presented at the IGARSS 2016 – 2016 IEEE International Geoscience and Remote Sensing Symposium;, Beijing, China. IEEE. p. 661–664.