ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume X-4/W3-2022
7th International Conference on Smart Data and Smart Cities (SDSC), 19–21 October 2022, Sydney, Australia

BUILDING FOOTPRINT SEGMENTATION USING TRANSFER LEARNING: A CASE STUDY OF THE CITY OF MELBOURNE

B. Neupane*, J. Aryal, A. Rajabifard

Department of Infrastructure Engineering, Faculty of Engineering and IT, The University of Melbourne,
Melbourne VIC 3010, Australia - bneupane@student.unimelb.edu.au, (jagannath.aryal, abbas.r)@unimelb.edu.au

Commission IV, WG IV/9

KEY WORDS: Deep Learning, Building Footprint Segmentation, Urban Feature Extraction, Fully Convolutional Network, U-Net, Trans-fer Learning

ABSTRACT:

Earth observation data including very high-resolution (VHR) imagery from satellites and unmanned aerial vehicles (UAVs) are the primary sources for highly accurate building footprint segmentation and extraction. However, with the increase in spatial resolution, smaller objects are prominently visible in the images, and using intelligent approaches like deep learning (DL) suffers from several problems. In this paper, we outline four prominent problems while using DL-based methods (P1, P2, P3, and P4)): (P1) lack of contextual features, (P2) requirement of a large training dataset, (P3) domain-shift problem, and (P4) computational expense. In tackling P1, we modify a commonly used DL architecture called U-Net to increase the contextual feature information. Likewise, for P2 and P3, we use transfer learning to fine-tune the DL model on a smaller dataset utilising the knowledge previously gained from a larger dataset. For P4, we study the trade-off between the network's performance and computational expense with reduced training parameters and optimum learning rates. Our experiments on a case study from the City of Melbourne show that the modified U-Net is highly robust than the original U-Net and SegNet, and the dataset we develop is significantly more robust than an existing benchmark dataset. Furthermore, the overall method of fine-tuning the modified U-Net reduces the number of training parameters by 300 times and training time by 2.5 times while preserving the precision of segmentation.

1. INTRODUCTION

Precise and accurate building footprints are useful for planning the smart cities in which geo-spatial industries and local agencies are investing and putting their effort. Very high-resolution (VHR) earth observation (EO) imagery is the primary source for the highly accurate building footprint extraction. The increase in spatial resolution of these images has increased the visibility of smaller objects and provided a forward direction with a positive impact in the domain of building footprint extraction. Meanwhile, a shift of paradigm from pixel-based image classification (PBIC) to object-based image analysis (OBIA) and most recently towards pixel-level semantic segmentation using deep learning (DL) methods in feature extraction is supporting this forward direction. However, there are several problems associated with DL-based semantic segmentation due to VHR datasets (Neupane et al., 2021). In this paper, we focus on four prominent problems in DL methods:

1. the lack of contextual features in the layers of deep learning (P1),
2. the requirement of a large labelled training dataset (P2),
3. the domain-shift problem that comes from the difference in train and test data (P3), and
4. computational expense due to the requirement of a large number of training parameters (P4).

We design a DL-based semantic segmentation method to tackle these problems in a case study from the City of Melbourne for building footprint extraction.

*Corresponding author.

Commonly used DL architectures like convolutional neural networks (CNNs) are effective for object detection, scene-wise classification, and feature extraction. However, in the case of semantic segmentation of building features, the use of pooling layers in CNN diminishes contextual information, resulting in inadequate PBIC in feature maps and predictions. As our solution to P1, we train and test a DL architecture of a fully convolutional network (FCN) family. A specific focus is on U-Net (Ronneberger et al., 2015). Unlike CNN, the symmetrical encoder-decoder architecture of U-Net with skip-connections can leverage both lower and higher-level contextual information for finer predictions. Multiple layers of convolutions, activation functions like rectified linear unit (ReLU), and max pooling combine the outputs of lower layers and higher layers to generate the final output. Batch normalization (BN) and ReLU are heavily applied to accelerate training and avoid vanishing gradient problems. U-Net learns patterns from all these layers when trained using optimizers such as Stochastic Gradient Descent (SGD) and backpropagation (BP) of error. We modify the original U-Net architecture to increase the contextual feature information.

To tackle P2, we perform a two-step training of the modified U-Net. The model is first trained on a large training dataset called WHU Building dataset (Ji et al., 2018). Then we transfer the learnings from this model to a new model trained on another dataset from the City of Melbourne. This allowed producing precise segmentation results on a relatively smaller dataset. Similarly for P3, we use the power of transfer learning (Torrey and Shavlik, 2010)-based fine-tuning to minimise the effects of domain shift between the two training datasets. Likewise for P4, we reduce the number of training parameters in the second step of training the models and produce three fine-tuning strategies. It is important to reduce the number of training parameters to train on a smaller dataset because a large number of training parameters demand a larger training dataset and vice-versa. We further

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W3-2022-173-2022 | © Author(s) 2022. CC BY 4.0 License.

experiment with the lowering of training parameters and increasing the learning rate in order to find optimal ways to reduce the computational expense.

Transfer learning is used to reduce the domain-shift impact caused by imaging sensors, resolution, and class representation, and to increase the performance of a DL model with fewer training samples and less computational power. First introduced in 1976, transfer learning (Bozinovski and Fulgosi, 1976) in neural networks is defined as "deep transfer learning" much later in 2018 as a task of a non-linear function that reflected a deep neural network (Tan et al., 2018). The reason behind such a time gap could be the recent spike in the usability of DL due to DL frameworks such as *Tensorflow*, *Keras*, and *Pytorch*, and the advancement in memory cards and graphics processing units (GPUs) in a computer. Transfer learning allows the transfer of knowledge gained from solving one problem to be used for similar problems, making it widely used in studies that lack enough training images and labels (Pan and Yang, 2009, Weiss et al., 2016). A domain-specific transfer learning to transfer the knowledge obtained from a global convolutional network (GCN) trained on VHR images to a network trained on medium resolution images is proposed to segment urban features (Panboonyuen et al., 2019). It is also used to transfer the learning between models trained on different satellite images collected from various sensors (QuickBird, Sentinel-2, and TerraSAR-X) with varying image resolution (Wurm et al., 2019) in order to segment slum areas on EO images. Some have used transfer learning to test their model for building extraction in different ablations of their experiments (Ji et al., 2018, Ji et al., 2019). Most transfer learning-based approaches are performed to transfer the weights of a pre-trained model on generic datasets such as Visual Object Classes (VOC) (Everingham et al., 2010) and ImageNet (Deng et al., 2009) dataset, and very few from the above-mentioned studies have used it between existing building footprint dataset. In our experiments, we evaluate three transfer learning-based fine-tuning strategies to minimise the effects of domain-shift, optimize the computational expense and reduce the training parameters without pre-training on the generic dataset. Similar techniques to reduce the training parameters have also been experimented by experimenting with several CNN architectures on a medical image dataset (Taormina et al., 2020).

The rest of the paper is structured as: Section 2. describes the datasets and method; Section 3. presents the experiments and results; and Section 4. concludes the paper with future direction.

2. METHOD

2.1 Data Preparation

We perform our experiments on two datasets: (i) WHU Building dataset and (ii) a new dataset that we build: Melbourne Building dataset. The WHU Building dataset (abbr. TR1) includes images of Christchurch collected from QuickBird, Worldview, IKONOS, and ZY-3, with spatial resolution ranging from 0.3m to 2.5m. The vector labels were extensively corrected and the original images were down-sampled from 0.075m to 0.3m by (Ji et al., 2018). The improvement, therefore, provided better accuracy to their tested models. We crop their 512x512 sized tiles into four equal parts, resulting in 256x256 tiles to keep the image size uniform between the two datasets we experiment on. A total of 23088 and 9664 tiles were prepared as train and test image samples in TR1.

For the second dataset, we develop Melbourne Building dataset (abbr. TR2). The labels in TR2 are sourced from a shapefile of building roof samples collected from "2018 Building Footprints" data provided by the City of Melbourne and image tiles of 1.2m ground resolution are collected for each building footprint using Nearmap API. Mask layers are created from the building footprint shapefile keeping the size of the image and label pair the same. Out of 13 Census of Land Use and Employment (CLUE) areas in the City of Melbourne, one suburb (Carlton) with 16.5% roof samples is considered as a test area and the remaining as a train area as shown in Figure 1. With an overlap of 50% between the adjacent tiles, 4889 and 435 tiles were prepared as train and test image samples.

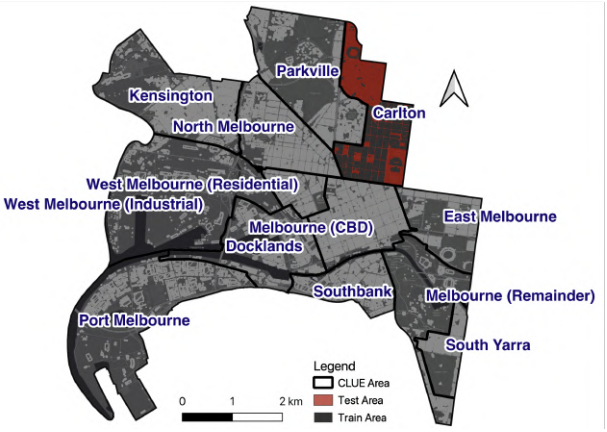

Figure 1. Census of Land Use and Employment (CLUE) areas of the City of Melbourne.

Melbourne Building dataset (TR2) is a complex small dataset as compared to TR1. Some samples of TR1 and TR2 are shown in Figure 2. The complexity comes from several factors including the smaller size of TR2 (TR2 is five t imes s maller t han TR1), lower spatial resolution (TR2 has four times lower resolution images), and the inclusion of high-rise buildings that are difficult to separate. The number of high-rise buildings in the City of Melbourne is 753, including 67 skyscrapers that are above 150m high. The WHU Building dataset that includes the building footprints of Christchurch, New Zealand contains only three high-rise buildings with the highest one of 86.5m, and mostly includes residential and short commercial buildings. The high-rise buildings bring other challenges from the shadows and angle of inclination of the camera sensors. Due to these factors of complexity in TR2, there exists a domain shift between the two datasets resulting in poor precision of the DL models trained on TR1 when validated on the TR2 dataset. Our methodology, therefore, tackles this problem as explained in the next sections.

2.2 Training and fine-tuning the modified U-Net

Our modification to the original U-Net focuses on the increment of features in the convolution blocks of the encoder. This modification allows extracting more feature information from our small TR2 dataset. In the encoder of the original U-Net architecture, the feature information is increased in the encoder by adding a pair of convolution and batch normalization layers on the bottleneck and two consecutive blocks before it. As the feature information is increased, the model can learn complex structures effectively. The up-sampling convolutions in the decoder later concatenate these high-resolution features and spatial information coming from the encoding convolutional blocks through skip branches. This change increases the number of extracted features while still preserving their spatial relationships as our solution to the problem (P1).

The training of the modified U-Net is done in two steps as shown in Figure 3. First, the modified U-Net is trained on the large TR1 dataset to produce a base model (M1) with a larger knowledge

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W3-2022-173-2022 | © Author(s) 2022. CC BY 4.0 License.

Figure 2. Sample image tiles from WHU Building dataset (TR1) and Melbourne Building dataset (TR2).

base. Secondly, M1 is further fine-tuned and trained on TR2 to produce a new model M2. The overall architecture, hyper-parameters, and augmentation steps are kept the same as before, but the weights are initialized from M1 using transfer learning-based fine-tuning strategies. We evaluate three fine-tuning strategies (TL1, TL2, and TL3) on M2 to optimize the number of training parameters. Also, the three strategies are tested with three learning rates of 1e-4, 1e-5, and 1e-6 to find the optimal rate for each fine-tuning strategy. This change of learning rate and fine-tuning strategies allow a surgical fine-tune of M1 to the training data TR2. Furthermore, the fine-tuning also tackles the problems (P2) and (P3) by leveraging the knowledge gained from an existing large dataset TR1 through model M1 and applying it to model M2 utilising the smaller TR2 dataset. This ensures the minimisation of the domain-shift effects from TR1 to TR2.

Explaining the three fine-tuning strategies, TL1 trains the convolutional layers from both the encoder and decoder side of the modified U-Net. TL2 trains only the layers from the decoder side. TL3 trains only the layers from the final block of the decoder. The number of training parameters significantly decreases from TL1 to TL3, keeping the training time shorter and models much smaller in size. The evaluation of these three strategies, therefore, helps in identifying the optimal computational expense with lower training parameters needed to segment the building features from the VHR satellite images, addressing the problem (P4). The evaluation of the modified U-Net against the original U-Net and the evaluation of three fine-tuning strategies with three different learning rates is presented in the next section.

3. EXPERIMENT AND RESULTS

In our experimental design, we evaluate the modified U-Net against the original U-Net and SegNet using the two datasets TR1 and TR2. Further, we test the modified U-Net in the cross-domain of the two datasets to see the extent of the domain shift problem and present the efficiency of the fine-tuning strategies in tackling the problem. We further present a trade-off between the model performance and computational expense by varying the learning rates during the surgical fine-tuning of the model.

3.1 Evaluation Metrics

We use several metrics to evaluate the robustness of our method on the validation dataset: (i) pixel accuracy, (ii) average accuracy, (iii) F1 score (aka. Dice Coefficient), (iv) Intersection over Union (IoU aka. Jaccard Coefficient), and (v) Matthews correlation coefficient (MCC aka. Phi coefficient). Pixel accuracy (Eqn. 1) calculates how often the predictions match binary labels. Average accuracy (Eqn. 2) is the average of Sensitivity (Eqn. 3) and specificity (Eqn. 4). Sensitivity and specificity respectively measure the proportion of actual positives and actual negatives that are correctly identified. F1 score (Eqn. 5) and IoU (Eqn. 6) is calculated from the 'area of overlap' between prediction and binary labels and 'area of union' that consists of all of the predictions and binary labels minus the overlap. Lastly, MCC (Eqn. 7) measures the difference between the prediction and binary labels considering the ratio between positive and negative elements. MCC is considered superior among the other accuracy metrics as it addresses the problem of imbalanced class (Chicco and Jurman, 2020). The metrics are represented by the following equations:

$$Pixel\ accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Average\ accuracy = \frac{Sensitivity + Specificity}{2} \tag{2}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$Specificity = 1 - \frac{TN}{TN + FP} \tag{4}$$

$$F1score = \frac{2 \times TP}{2 \times TP + FN + FP} \tag{5}$$

$$IoU = \frac{TP}{TP + FN + FP} \tag{6}$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7}$$

where, TP (true positive) is the case when prediction matches binary label (i.e. prediction = 1, label = 1); FP (false positive) is prediction = 1, label = 0; FN (false negative) is prediction = 0, label = 1; and TN (true negative) is prediction = 0, label = 0. Other than five metrics, we also evaluate the time required to train the models per step of training.

3.2 Training details

Any experimentation that uses DL-based methods is incomplete without providing the training details. Here, we provide the training details of the modified U-Net model on the TR1 and TR2 datasets. We wrap all the models of our experiments in the Keras framework. Several random augmentation techniques such as rotation, horizontal flip, width and height shift, shear, and zoom are applied to increase the training dataset during the training of the models that we experiment on. We use a mini-batch of 2. The number of steps is set as the ratio of the number of training images to the batch size. As the number of image samples is higher in TR1, the number of epochs to train the base models (M1) is kept lower such that the total number of steps can be kept approximately similar during all experiments. A learning

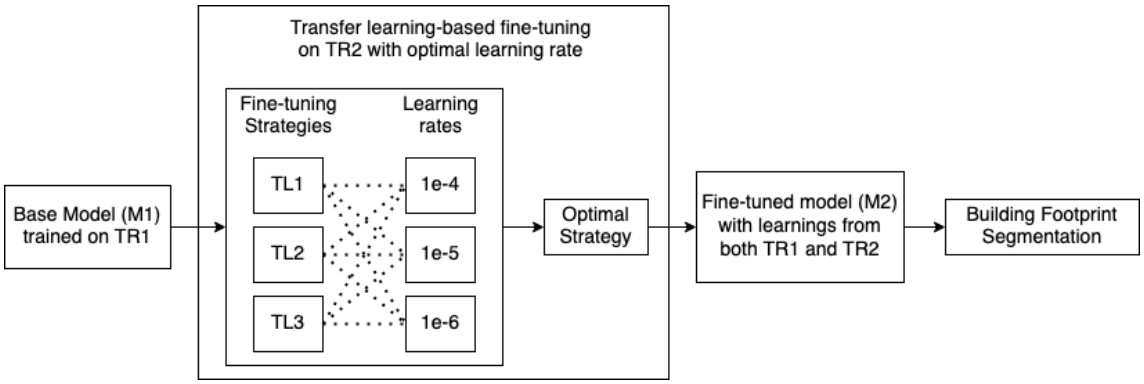

Figure 3. Overall workflow of training and fine-tuning.

rate of 1e-4 is used to train the base model M1, which is changed to the optimal value during the fine-tuning. A *dropout* of 50% is used to avoid over-fitting, and an *Adam optimizer* is used to optimize the model with an initializer of *He Normal* for the activation function of *Rectified Linear Units (ReLU)*. A *binary cross-entropy (CE)* loss function is used to monitor the models. U-Net and SegNet are used to compare the performance of the modified U-Net that is developed. The batch normalization, activation, initializer, optimizer, and learning rate are kept the same as in our modified U-Net. However, in the SegNet, pooling is done using the argmax function instead of the maxpooling. All experiments are carried out on a Macbook M1 Pro with 8-core CPU, 14-core GPU, 14-core NPU, and 16GB RAM.

3.3 Performance of the base model

Our base model M1 (i.e. the model trained before transfer learning) of the modified U-Net (our solution to P1) is compared against the original U-Net and SegNet using both TR1 (WHU dataset) and TR2 (Melbourne Building dataset). Starting with the number of training parameters, the modified U-Net has approximately 7 and 5.5 million fewer training parameters than U-Net and SegNet respectively. As TR1 contains almost five times more training data, the time taken per step to train the models on TR1 is higher than on TR2. Table 3.3 presents the same-domain validation results obtained from the first step of training the models on datasets TR1 and TR2. Same-domain implies the case when the training and validation data are from the same dataset. The results show that all three models performed better in the TR1 dataset when compared to the models trained on TR2. This is due to the complexity of the TR2 dataset as mentioned before. Despite the complexity, the modified U-Net produced the best pixel accuracy (0.94 vs. 0.93), average accuracy (0.73 vs. 0.72), F1 score (0.62 vs. 0.42), IoU (0.53 vs. 0.33), and MCC (0.43 vs. 0.42) in TR2 dataset when compared against U-Net. SegNet is slightly behind both versions of U-Net. Unlike the pixel accuracy, the other metrics are usually lower because these metrics are biased in mainly reporting how well the model identifies the positive case. The increased average accuracy, F1 score, IoU, and MCC from the modified U-Net in TR2 dataset shows that the increment of features while training an FCN on smaller and complex data such as TR2, can produce better segmentation. Figure 4 shows the sample results from the modified U-Net, U-Net, and SegNet on TR2. The next section presents the cross-domain validation results of the modified U-Net. Cross-domain implies the case when the training and validation data are from a different dataset.

3.4 Cross-domain validation and domain-shift

The previous section presented the results from the first step of training on datasets TR1 and TR2 when tested against the validation data of the corresponding dataset. However, due to the difference in complexity in the two datasets, the performance changes when cross-validated against the other's validation data. The second columns of Table 3.4 and 3.4 present the performance of the base model (modified U-Net) trained on TR1 but validated against TR2 and vice-versa respectively. Compared to the same-domain validation from Table 3.3, the cross-domain validation of the model M1 trained on TR1 and validated on TR2 shows significantly lower average accuracy (0.85 vs. 0.55), F1 score (0.83 vs. 0.28), IoU (0.66 vs. 0.22), and MCC (0.48 vs. 0.16). Similarly, M1 trained on TR2 and validated on TR1 also shows poor performance as compared to Table 3.3 and 3.4.

To train a robust model on a small yet complex dataset (solution to P2) and to minimise the effects of the domain shift (P3) between the two datasets, we take the base model M1 trained on TR1 and fine-tune it to TR2 so that fine-tuned model M2 can perform in both data domains. For this, we perform the second step of training using transfer learning to fine-tune the model M1 with the same number of steps but optimal learning rate. The optimal learning rate is chosen from the experiments shown in the next section. We test three surgical fine-tuning strategies (TL1, TL2, and TL3) with a reduced number of training parameters as explained before. The study of the trade-off between the network's performance and computational expense with reduced training parameters and optimum learning rate tackles P4.

The number of training parameters significantly decreases from TL1 to TL3 as shown in Table 3.4. This results in reduced time to train the three fine-tuned models. When compared to the original U-Net, the three strategies TL1, TL2, and TL3 are set up with 1.3, 3.6, and approximately 300 times lower training parameters. Using the strategy TL3, the model trains nearly 2.5 times faster than using TL1 (119ms vs. 291ms per step). The major difference however comes in terms of accuracy metrics. Compared to the base model M1 trained only on TR1, the model M2 previously trained on TR1 and fine-tuned on TR2 produced better pixel accuracy (0.94 vs. 0.91), average accuracy (0.75 vs. 0.55), F1 score (0.64 vs. 0.28), IoU (0.55 vs. 0.22), and MCC (0.45 vs. 0.16) as seen from the results. All three strategies out-performed M1, with TL3 as the optimal strategy both in terms of computational expense and evaluation metrics. The sample results from the cross-validation of M1 and M2 with the fine-tuning strategy of TL3 on TR2 are shown in Figure 5.

The comparison of M1 trained only on TR2 and M2 trained on TR1 and fine-tuned on TR2 with the three strategies is shown in Table 3.4, where the validation data is of TR1. The results are similar to those from Table 3.4. M2 produced better pixel accuracy (0.92 vs. 0.89), average accuracy (0.81 vs. 0.74), F1 score (0.67 vs. 0.52), IoU (0.56 vs. 0.40), and MCC (0.55 vs.

Evaluation Metrics	Mod. U-Net		U-Net		SegNet	
	TR1	TR2	TR1	TR2	TR1	TR2
Train Params	23.949M		31.037M		29.443M	
Non-train Params	5504		3968		15874	
Steps	57,720	48,980	57,720	48,980	57,720	48,980
Time/step (ms)	321	291	365	345	393	338
Pixel Acc.	0.973	0.938	0.973	0.928	0.969	0.854
Average Acc	0.847	0.730	0.826	0.724	0.831	0.715
F1 score	0.834	0.622	0.844	0.437	0.666	0.356
IoU	0.761	0.532	0.770	0.342	0.584	0.262
MCC	0.688	0.425	0.681	0.419	0.673	0.346

Table 1. Same-domain validation results of the modified U-Net on TR1 and TR2 datasets compared to U-Net and SegNet.

Evaluation Metrics	M1	M2		
		TL1	TL2	TL3
Train Params	-	23.943M	9.229M	0.109M
Non-train Params	-	0.011M	14.726M	23.846M
Steps	-	48980	48980	48,980
Time/step (ms)	-	291	194	119
Pixel Acc.	0.907	**0.944**	0.933	0.943
Avg. Acc.	0.550	0.743	0.735	**0.745**
F1 score	0.275	0.643	0.619	**0.643**
IoU	0.217	**0.552**	0.521	0.551
MCC	0.156	0.447	0.416	**0.447**

Table 2. Cross-domain validation results on Melbourne Building dataset (TR2). M1 refers to the base model trained on the WHU Building dataset (TR1) and validated on TR2. M2 refers to the model previously trained on TR1 and fine-tuned on TR2 with three fine-tuning strategies (TL1, TL2, and TL3).

0.42) when compared to M1. Once again, all three strategies outperformed M1, with TL3 as the most optimal strategy to fine-tune. The sample results from the cross-validation of M1 and M2 with the fine-tuning strategy of TL3 on TR1 are shown in Figure 5.

Evaluation Metrics	M1	M2		
		TL1	TL2	TL3
Pixel Acc.	0.887	0.922	0.891	**0.922**
Avg. Acc.	0.739	**0.811**	0.772	0.809
F1 score	0.517	0.673	0.530	**0.673**
IoU	0.397	**0.560**	0.407	0.559
MCC	0.424	0.548	0.456	**0.545**

Table 3. Cross-domain validation results on WHU Building dataset (TR1). M1 refers to the base model of modified U-Net that is trained on only the Melbourne Building dataset (TR2). M2 refers to the model previously trained on TR1 and fine-tuned on TR2 with three fine-tuning strategies (TL1, TL2, and TL3)

It has to be noted from the experiments in this section that the TR2 dataset is more robust compared to TR1 during cross-domain validation in terms of average accuracy, F1 score, IoU, and MCC. The domain-shift effects on the base model M1 trained on TR2 are smaller than the M1 trained on the WHU dataset. The M1 trained on TR1 produced a 55% lower F1 score (0.83 vs. 0.28), while M1 trained on TR2 produced a 10% lower F1 score (0.62 vs. 0.52). This shows that TR2 is 45% robust in terms of F1-score during the cross-validation of TR1 and TR2. Similarly, TR2 is 31%, 41%, and 32% more robust in terms of average accuracy, IoU, and MCC respectively.

3.5 The trade-off between accuracy and computational expense

As shown by the experiments earlier, the reduced number of training parameters in the three fine-tuning strategies produced significantly smaller and faster models ascending from TL1 to TL3. It is seen that the computational expense to train the DL model can be reduced without losing the performance by reducing the training parameters. Another fundamental parameter of training a deep learning network to lower the computational expense is the learning rate. Conceptually, a lower learning rate optimizes the network's weights allowing an operational fine-tuning of the network to the training data, while increasing the training and processing time. A higher learning rate allows the network to learn faster with smaller training time, but at the cost of precision of the adjustments. In this section, we experiment with the three fine-tuning strategies with three learning rates to fine-tune the base models and see the trade-off between the precision and computational expense required for fine-tuning. This also helps in finding the optimal learning rate that results in the highest accuracy metrics without taking many training steps to converge.

We experiment on three learning rates of 1e-4, 1e-5, and 1e-6 as shown in Table 3.5. The experiment shows that despite no significant difference being seen in time per step due to the change in learning rates, there is a mixed difference in evaluation metrics. In our experiments, reducing the learning rate by 10 times from 1e-4 to 1e-5 produces the best fine-tuning on TL1, while keeping it the same, results in the best fine-tuning in TL2 and TL3. However, lowering it by 100 times to 1e-6 produces the lowest evaluation metrics on strategies TL2 and TL3. TL1 performs the worse at the faster learning rate of 1e-4. From the experiment, it is seen that smaller training parameters can produce the best results with a faster learning rate, and higher training parameters need a slower learning rate.

4. CONCLUSION

Accurate building footprints are advantageous for planning smart and sustainable cities. Geo-spatial industries and local data curators have therefore put tremendous efforts into producing accurate and precise building footprint data. The industries rely on state-of-the-art DL-based methods to extract the buildings from fundamental EO data and to produce their data products. However, the data products lack robustness and reliability due to several problems associated with the DL-based methods. In particular, we focus on four prominent problems of DL methods: (P1) lack of contextual features, (P2) requirement of a large training dataset, (P3) domain-shift problem, and (P4) computational expense. We increase the feature information in the layers of convolutions in a symmetrical DL model of the FCN family called U-Net to prepare a modified U-Net to tackle P1. This increment of features is helpful for our Melbourne building dataset, which is smaller and complex compared to other existing building datasets. To address P2, we train the modified U-Net in two steps to leverage the learning from a larger dataset (WHU Building dataset) into a model trained on a smaller and more complex

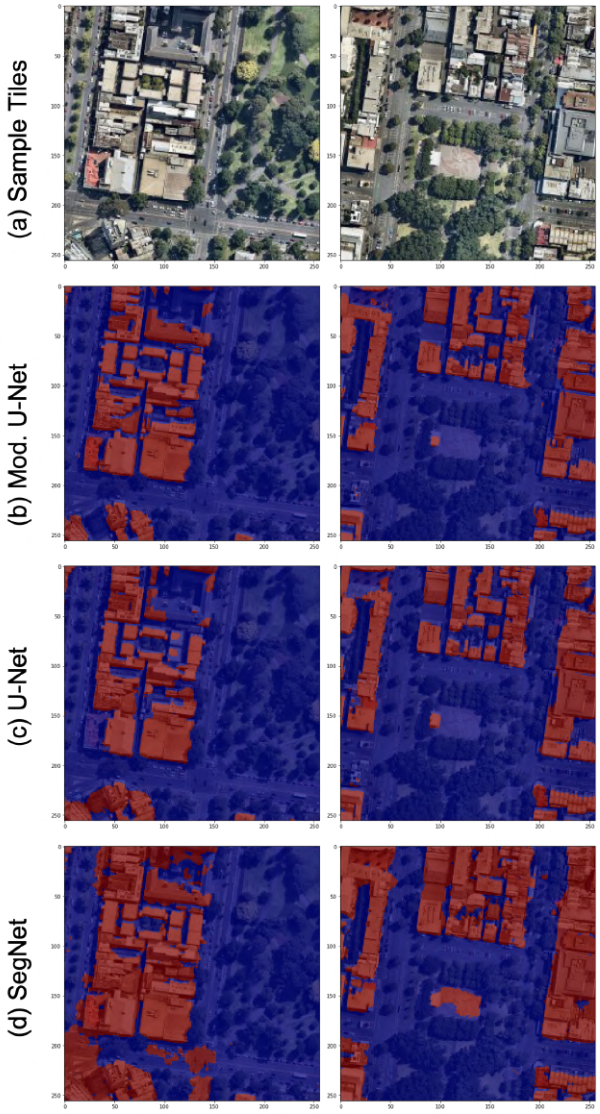

Figure 4. Sample results from the modified U-Net, U-Net, and SegNet on the Melbourne Building dataset (TR2).

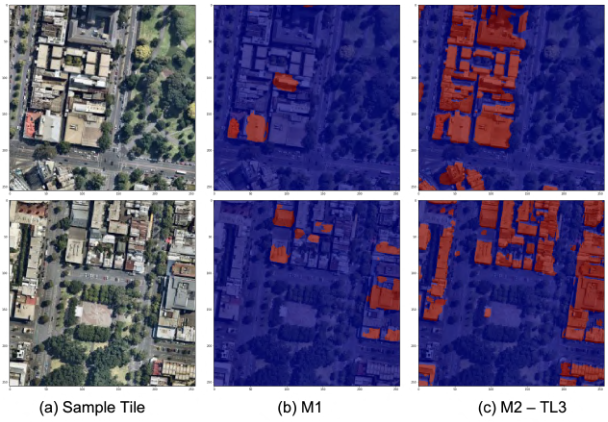

Figure 5. Sample results from cross-validation of the base model (M1) of modified U-Net and fine-tuned model (M2) with three fine-tuning strategies (TL1, TL2, and TL3) on the Melbourne Building dataset (TR2).

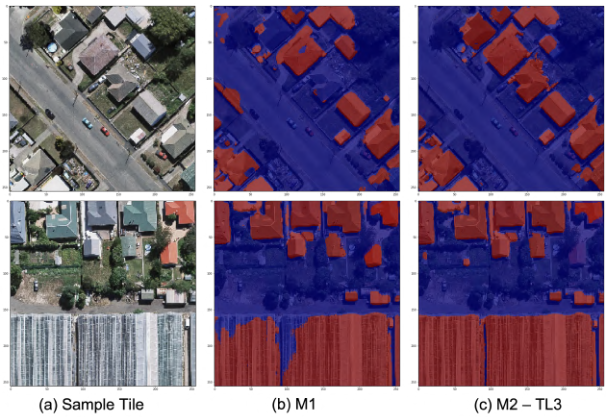

Figure 6. Sample results from cross-validation of the base model (M1) of modified U-Net and fine-tuned model (M2) with three fine-tuning strategies (TL1, TL2, and TL3) on the WHU Building dataset (TR1).

Melbourne Building dataset using transfer learning-based fine-tuning. This fine-tuning not only reduced the number of training samples but also produced a model that could achieve increased precision in cross-domain validation of the two datasets, minimising the domain-shift effects (P3). We test three fine-tuning strategies to see how the minimisation of training parameters in the DL model affects the precision of the output. Furthermore, we also experiment with the reduction of the learning rate during the fine-tuning of the models to see the trade-off between precision and computational expense (P4).

The increment of features in the encoder of the modified U-Net significantly improved F1 score (by 18%), and IoU (by 19%) and slightly improved MCC on the same-domain evaluation using the Melbourne Building dataset. The improved F1 score shows the model's robustness in terms of incorrectly classified pixels, the increased IoU effectively highlights the similarity between the ground truth and prediction, and better MCC shows the model's robustness in the majority of both positive and negative cases. When performing the cross-domain validation using the WHU dataset and the Melbourne Building dataset, the overall method of fine-tuning the modified U-Net with reduced training parameters with optimal fine-tuning strategy and learning rate shows a significant increase in all accuracy measures and a remarkable decrease in the domain-shift. The experiments from the three fine-tuning strategies reduced the number of training parameters by 300 times and training time by 2.5 times at the cost of model performance in our experiments. It is also seen that fine-tuning could significantly optimize the training parameters by 300% while also producing the most precise segmentation. Furthermore, choosing the optimal learning rate during the fine-tuning effectively improved the segmentation. To sum up, we successfully tackle the four prominent problems of DL methods with our overall method.

Our smaller yet more complex Melbourne Building dataset with aerial imagery of four times lower spatial resolution, five times smaller number of images, and a higher number of high-rise buildings shows significant robustness compared to the WHU Building dataset when it comes to cross-domain validation. This dataset can further be used by the geospatial industries as well as researchers in the domain of city planning and urban feature modelling. In future works, we recommend the use of robust post-processing techniques to produce more accurate building footprint datasets.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W3-2022-173-2022 | © Author(s) 2022. CC BY 4.0 License.

Learning Rate	Evaluation Metric	TL1	TL2	TL3
LR = 1e-4	Time/step (ms)	287	**194**	**119**
	Pixel Acc.	0.907	**0.933**	**0.943**
	Average Acc.	0.759	**0.735**	**0.745**
	F1 score	0.597	**0.619**	**0.642**
	IoU	0.502	**0.521**	**0.551**
	MCC	0.405	**0.416**	**0.447**
LR = 1e-5	Time/step (ms)	**291**	197	121
	Pixel Acc.	**0.944**	0.926	0.905
	Average Acc.	**0.743**	0.738	0.732
	F1 score	**0.643**	0.582	0.482
	IoU	**0.552**	0.485	0.385
	MCC	**0.447**	0.402	0.359
LR = 1e-6	Time/step (ms)	295	201	123
	Pixel Acc.	0.926	0.901	0.886
	Average Acc.	0.740	0.738	0.725
	F1 score	0.602	0.509	0.457
	IoU	0.506	0.411	0.359
	MCC	0.412	0.369	0.330

Table 4. Validation results obtained on TR1 and TR2 datasets using three fine-tuning strategies TL1, TL2, and TL3, each evaluated with three learning rates.

ACKNOWLEDGEMENTS

Bipul Neupane is supported by the University of Melbourne for his Ph.D. research and is awarded by Melbourne Research Scholarship.

REFERENCES

Bozinovski, S. and Fulgosi, A., 1976. The influence of pattern similarity and transfer learning upon training of a base perceptron b2. In: *Proceedings of Symposium Informatica*, Vol. 3, pp. 121–126.

Chicco, D. and Jurman, G., 2020. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics* 21(1), pp. 1–13.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE conference on computer vision and pattern recognition*, Ieee, pp. 248–255.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J. and Zisserman, A., 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88(2), pp. 303–338.

Ji, S., Wei, S. and Lu, M., 2018. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Transactions on Geoscience and Remote Sensing* 57(1), pp. 574–586.

Ji, S., Wei, S. and Lu, M., 2019. A scale robust convolutional neural network for automatic building extraction from aerial and satellite imagery. *International Journal of Remote Sensing* 40(9), pp. 3308–3322.

Neupane, B., Horanont, T. and Aryal, J., 2021. Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis. *Remote Sensing* 13(4), pp. 808.

Pan, S. J. and Yang, Q., 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10), pp. 1345–1359.

Panboonyuen, T., Jitkajornwanich, K., Lawawirojwong, S., Srestasathiern, P. and Vateekul, P., 2019. Semantic segmentation on remotely sensed images using an enhanced global convolutional network with channel attention and domain specific transfer learning. *Remote Sensing* 11(1), pp. 83.

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*, Springer, pp. 234–241.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C. and Liu, C., 2018. A survey on deep transfer learning. In: *International conference on artificial neural networks*, Springer, pp. 270–279.

Taormina, V., Cascio, D., Abbene, L. and Raso, G., 2020. Performance of fine-tuning convolutional neural networks for hep-2 image classification. *Applied Sciences* 10(19), pp. 6940.

Torrey, L. and Shavlik, J., 2010. Transfer learning. In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, IGI global, pp. 242–264.

Weiss, K., Khoshgoftaar, T. M. and Wang, D., 2016. A survey of transfer learning. *Journal of Big data* 3(1), pp. 9.

Wurm, M., Stark, T., Zhu, X. X., Weigand, M. and Taubenböck, H., 2019. Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing* 150, pp. 59–69.