www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 12 Issue 08 August 2023, Page No. 25792-25800
ISSN: 2319-7242 DOI: 10.18535/ijecs/v12i08.4751

A Comprehensive Study of Maximum Building Area Estimation

Shantanu Sharma, Shaurya Shrivastava, Srikanth M, Vansh Hans, Dr. Rubeena Vohra (Asst. Prof.)

Department of Electronics and Communication Bharati Vidyapeeth's College Of Engineering
New Delhi, India

Abstract

In recent years, with the increase in urbanization, it has become important to keep track of infrastructure in a geographical area. Urban planners rely on accurate building area calculations to assess the spatial characteristics and dynamics of built environments, informing critical decision-making processes. Satellite technologies have been developing rapidly and applied to many remote sensing applications due to which high-resolution images of a geographical area are obtained with much ease. To estimate the area of buildings, deep neural networks are employed to analyze images from high-resolution satellite data. This involves extracting valuable semantic features and segmenting all buildings. Subsequently, four models are evaluated and compared to assess their performance in accurately segmenting buildings within the images. Upon training and evaluation, the models are compared based on the mIoU score, with the UNet model (utilizing a ResNet18 encoder) achieving the highest mIoU score of 0.83.

Keywords: Satellite imagery, area estimation, Neural Networks, deep learning, segmentation.

1. Introduction

Increase in urbanization has led to various problems nowadays. While urbanization can bring about economic development, improved infrastructure, and increased opportunities, it also gives rise to several pressing issues like overcrowding, housing shortages, inadequate infrastructure, environmental degradation, and social inequalities. Addressing the problems of urbanization requires careful urban planning, sustainable development strategies, and effective governance.

Calculating the building area plays a crucial role in addressing these issues and promoting sustainable urban development. Calculating building area relates directly to urbanization as it influences the spatial distribution and intensity of development within cities. When urban areas expand without proper planning, there is a tendency for inefficient land use and excessive building sprawl. This can lead to the consumption of valuable agricultural land, loss of natural habitats, and increased pressure on infrastructure systems. By accurately calculating the building area, urban planners and policymakers can gain insights into the existing urban fabric and make informed decisions to tackle the challenges of urbanization.

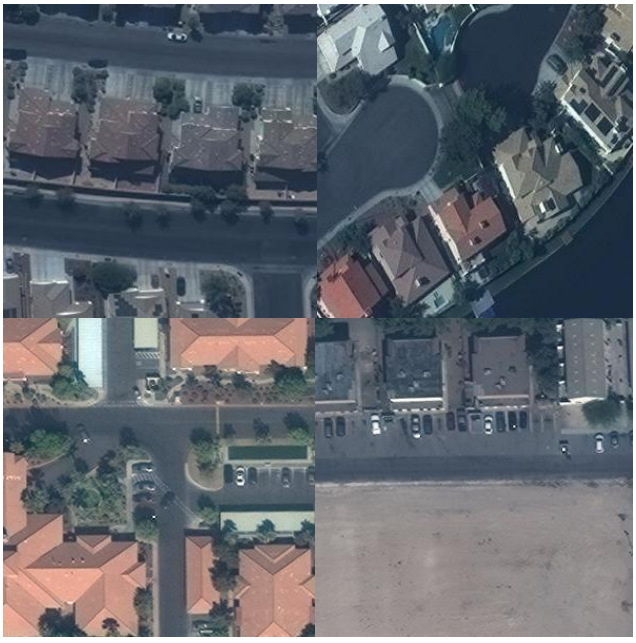


Figure 1: Some samples from the dataset.

Calculating building areas helps in solving the problem of urbanization in various ways including effective land use, infrastructure planning, sustainable development, social equality, etc.

In recent years, there has been substantial and accelerated progress in satellite technologies, leading to their widespread utilization across various remote sensing applications. Consequently, the utilization of satellite imagery for the detection and extraction of buildings has emerged as a prevailing trend. However, the ground resolution of satellite images is relatively low, and large errors are caused when calculating the area, which will affect the property value assessment. There are two main reasons why satellite images are used for property evaluation. First, equipped with a high-definition camera, high-resolution images can be easily procured even at the centimeter level. Second, in some disaster-prone areas, using satellite images to collect data can reduce the risk.

The technique used in this paper to calculate the area of the buildings is segmentation. Image segmentation is the process of partitioning a digital image into multiple image segments, also known as image regions or image objects. Recently, Convolutional Neural Networks (CNN) [1] are showing exceptional results in the field of segmentation. The goal of segmentation is to simplify or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in images. More precisely, image segmentation is the process of designating a label to every pixel in an image such that pixels with the same label share certain characteristics. There are two types of segmentation, semantic and instance. Semantic segmentation and instance segmentation are two advanced techniques used in computer vision and image processing to understand and analyze images at a more granular level. While both methods involve segmenting images, they differ in their objectives and outputs.

Semantic segmentation focuses on assigning semantic labels to each pixel in an image, grouping them into meaningful categories. The goal is to understand the overall scene and identify different objects or regions based on their semantic class. For example, in an image containing a road, cars, and pedestrians, semantic segmentation would assign each pixel to a class such as "road," "car," or "person." The output of semantic segmentation is a pixel-level mask that represents the class label for each pixel in the image. It provides a holistic understanding of the scene without differentiating individual instances.

Instance segmentation goes beyond semantic segmentation by not only assigning semantic labels but also distinguishing between individual instances of objects within a particular class. It aims to identify and differentiate each distinct object instance separately. For example, in an image with multiple cars, instance segmentation would label each car with a unique identifier or instance ID. The output of instance segmentation is a pixel-level mask that not only assigns a semantic label but also assigns a unique ID to each pixel corresponding to a specific instance. This enables precise object detection, tracking, and analysis at the individual object level. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each pixel within a given region exhibits similarity in terms of certain characteristics or computed properties, such as color, intensity, or texture.

In this paper, segmentation is utilized to distinguish buildings from the rest of the area.

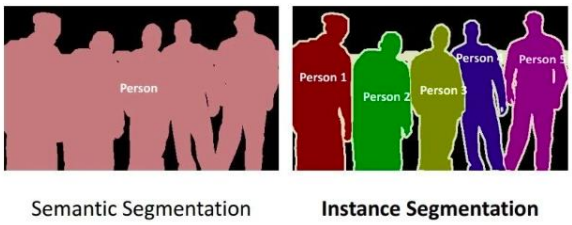


Figure 2: Semantic segmentation vs Instance segmentation.

The result of semantic segmentation in this case is to extract the contour of the building and the number of pixels within the mask of the building. To calculate the area, the unit area represented by each pixel is determined and the building area is estimated by adding area within the pixels classified as buildings by the model.

2. Related work

Building area estimation using deep learning models [2] applied to satellite imagery has gained significant attention in recent years. Researchers have explored various approaches and techniques to accurately estimate the area of buildings, contributing to the advancement of this field. In this section, key studies that are relevant to the research on the comparative analysis of PPM, UPerNet, HRNet, and U-Net models for building area estimation are reviewed.

Zhao et al. [3] proposed the PSPNet, a deep learning framework designed for semantic segmentation

tasks. The PSPNet introduced a pyramid pooling module that effectively captures multi-scale contextual information, enabling more accurate segmentation results. By leveraging the pyramid pooling module, the PSPNet demonstrated superior performance in scene understanding tasks, including building area estimation. Incorporating the PPM module into the ResNet architecture improves the accuracy of building area estimation by leveraging its multi-scale context aggregation capabilities.

Xiao et al. [4] proposed the UPerNet framework to achieve comprehensive scene understanding through precise semantic segmentation. Unified Perceptual Parsing (UPP) focuses on holistic scene understanding by simultaneously addressing pixel-level semantic segmentation and global scene parsing. The techniques and approaches employed in UPP are adaptable to building area estimation tasks. Considering the holistic context of the scene, UPP improves the accuracy and contextual understanding of building segmentation, enhancing the precision of building area estimation.

Sun et al. [5] proposed HRNetV2, a High-Resolution Network that maintains high-resolution feature maps throughout the network to capture fine-grained details. HRNetV2 has demonstrated state-of-the-art performance in various vision tasks, including semantic segmentation. Emphasizing the significance of capturing fine-grained details for accurate building area estimation, HRNetV2 enables precise delineation of building boundaries and improves the accuracy of area estimation.

Ronneberger et al. [6] proposed the U-Net architecture, originally designed for biomedical image segmentation. U-Net's encoder-decoder architecture with skip connections has been widely adopted in various segmentation tasks, including building area estimation from satellite imagery. The skip connections enable the network to effectively capture both low-level and high-level features, facilitating precise localization and segmentation of buildings.

These relevant works provide important foundations and methodologies for this research on comparative analysis. By examining the strengths and unique characteristics of PPM, UPerNet, HRNet, and U-Net models, valuable insights are learned about their performance in building area estimation. By comparing these models on metric mean IoU (mIoU), the most suitable model for accurate and efficient building area estimation from satellite imagery is determined.

3. Proposal

In this paper, four different segmentation models are trained on the same dataset and then are compared by evaluating each model on a separate validation dataset.

3.1 Models

Choice of models is crucial to any deep learning project or experiment. Given the limited computational resources, lightweight models are selected, which means that they have a smaller number of parameters so that models can be loaded easily onto GPU memory.

Another factor for choosing the models is to consider what type of data they were originally designed to fit. Three out of four selected models were used before for similar kinds of data (satellite imagery) and hence their architecture better suits the target dataset.

All the selected models employ an encoder-decoder based [7] approach. Encoder-decoder based models eliminate the need for fully connected layers, which means that models can be used for segmentation tasks using fully convolutional networks (FCNs) [8]. The encoder-decoder-based approach has become one of the most popular approaches for semantic segmentation. It has been used to achieve state-of-the-art results on a variety of datasets, including the PASCAL VOC [9] dataset and the Cityscapes [10] dataset.

An FCN is composed of an encoder and a decoder. The encoder is responsible for extracting features from the image, while the decoder is responsible for upsampling the features and generating the final segmentation mask.

Below is a brief overview of each of the models that are compared in this paper.

3.1.1 ResNet18 + Pyramid Pooling Module

This model, also called ResNet18_PPM, has ResNet18 as the encoder and Pyramid Pooling Network as its decoder. **ResNets** [11] or residual networks solve the problem of vanishing gradients when training very deep neural networks. ResNets tackle this issue by utilizing skip connections, also known as shortcut connections or identity mappings. These connections allow the gradient to bypass one or more layers and directly flow to

deeper layers, ensuring that the gradients remain strong and enabling better optimization. The key idea behind ResNets is to learn residual functions, meaning the network learns to approximate the difference between the desired output and the current output of the layers being bypassed.

The residual block, which is the basic building block of a ResNet, consists of a set of convolutional layers followed by an element-wise addition operation that combines the input with the output of those layers. The input to the block is called the "identity" or the "shortcut connection", while the output is referred to as the "residual". By adding the residual to the identity, the network can learn to adjust the output of the convolutional layers, effectively allowing them to model the residual function.

ResNet18 is a specific variant of the ResNet architecture that consists of 18 layers, including convolutional layers, batch normalization, ReLU activation functions, and pooling layers.

The accuracy of convolutional networks depends on the global context information or features that they capture. The number of features captured relies on the receptive field of a network. There is often a mismatch between the receptive field of the network in theory and in practice, **Pyramid Pooling Module** (PPM) [3] addresses this problem by capturing multi-scale contextual information from an input feature map. It does so by feature fusion under different scales. It addresses the challenge of effectively incorporating global context information into the segmentation process. By aggregating information from multiple scales, the PPM enables the network to make more informed predictions about the semantic class of each pixel.

Downsampling the input images by ResNet18, followed by feature extraction by PPM, and the final upsampling of the pooled features, the model can be successfully trained for semantic segmentation tasks.

3.1.2 ResNet50 + UperNet

This model, which is abbreviated as ResNet50_Upernet, has ResNet50 as the encoder and a network based on **Feature Pyramid Network** (FPN)[12] and Pyramid Pooling Module (PPM) as the decoder.

ResNet50 is just another variant of a class of neural networks called ResNets. It consists of 50 layers, including convolutional layers, batch normalization, ReLU activation functions, and pooling layers. ResNet50 has significantly more layers compared to ResNet18, allowing it to capture richer and more diverse features from input images.

The decoder is inspired by **UPerNet** [4] (Unified Perceptual Parsing Network), which combines both FCN and PPM to preserve high-quality semantic features and increase the empirical receptive field. In UPerNet, a PPM head is appended in the last layer of the back-bone network, before feeding the features to FCN.

3.1.3 HRNetV2

This model is the second version of the original HRNet [5] (High-Resolution Network) which preserves the high-resolution representations by connecting high-to-low-resolution convolutional feature maps in parallel. The model architecture employs repeating multi-resolution blocks. A multi-resolution block consists of a multi-resolution group convolution and a multi-resolution convolution.

In the original approach **HRNetV1**, only the feature maps from the high-resolution convolutions are considered, therefore only a subset of output channels from the high-resolution convolutions is used and other subsets from low-resolution convolutions are lost. This was tackled in **HRNetV2** with a small modification by exploiting other subsets of channel output from low-resolution convolutions. This was achieved by rescaling the low-resolution feature maps through bilinear upsampling and concatenating with the high-resolution feature maps. The benefit is that the capacity of the multiresolution convolution is fully utilized, and it adds only a small parameter count.

3.1.4 UNet (With ResNet18 encoder)

UNet is a specific type of encoder-decoder architecture that is widely used for semantic segmentation tasks. It is known for its symmetric structure which resembles the letter “U” (hence the name UNet) and skip connections that enable the fusion of low-level and high-level features.

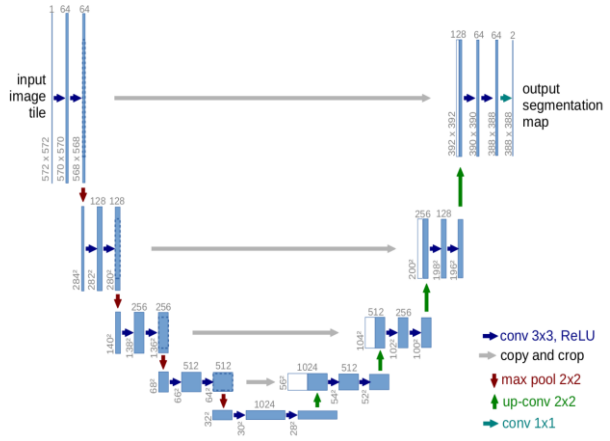


Figure 3: U-Net architecture from the original paper [6].

UNet [6] architecture consists of two parts: **encoder** and **decoder**. The contracting path/encoder of UNet consists of multiple downsampling blocks, each typically composed of convolutional layers followed by pooling or strided convolutions. These blocks progressively reduce the spatial resolution of the feature maps while increasing the number of channels, capturing high-level semantic information. The expanding path/decoder of UNet consists of multiple upsampling blocks, each composed of upsampling layers followed by convolutional layers. The upsampling blocks progressively increase the spatial resolution of the feature maps while reducing the number of channels. Importantly, the UNet architecture includes skip connections that directly connect corresponding layers between the contracting and expanding paths. These skip connections enable the fusion of low-level details and high-level context, helping to refine the segmentation output.

3.2 Modifications made to the models

To get decent results on a different dataset. Some modifications to the models (mentioned in Section 3.1) are made. Each modification is explained below.

Loss Function

For the models in Section 3.1.1, 3.1.2, and 3.1.3 the default **negative log-likelihood loss** (NLL loss) [13] function is replaced with **dice loss** [14]**.** NLL loss is calculated by the negative of the log of the probability that the model predicts the correct label.
Dice loss is calculated using the Dice coefficient. The Dice coefficient is twice the intersection of two sets divided by the sum of the two sets. The loss is calculated as,

$$Dice\ Loss\ =\ 1\ -\ (\frac{2\times|X\cap Y|)}{|X|\ +\ |Y|})$$

Dice loss has many advantages over NLL loss when it comes to semantic segmentation:

1. Handling Class Imbalance: Semantic segmentation datasets typically exhibit class imbalance, where the number of background pixels far exceeds the number of foreground pixels. This class imbalance can lead to biased models that prioritize background predictions. Dice loss addresses this issue by explicitly emphasizing the correct prediction of foreground pixels, promoting accurate segmentation results even in imbalanced datasets. NLL loss, on the other hand, treats each class equally and may not effectively handle class imbalance.

2. Similarity Measure: Dice loss directly measures the similarity between the predicted and ground truth segmentation masks using the Dice coefficient. It evaluates the overlap of foreground regions, making it more aligned with the evaluation metric used in semantic segmentation tasks. In contrast, NLL loss focuses on the probability distribution of different classes and may not directly capture the similarity or alignment between segmentation masks.
3. Robustness to False Positives and Negatives: Dice loss is less sensitive to false positives and false negatives compared to NLL loss. NLL loss penalizes both types of errors equally, which can be problematic when dealing with segmentation tasks where false positives or false negatives have different consequences. Dice loss, by considering the overlap of the predicted and ground truth masks, encourages models to produce segmentations that have a better balance between false positives and false negatives, leading to more accurate and visually plausible results.

4. Gradient Behavior: Dice loss often exhibits smoother and more stable gradients

compared to NLL loss. This smoothness can be beneficial for optimization during training, enabling more stable convergence and potentially avoiding issues like vanishing or exploding gradients.

5. Direct Optimization of Evaluation Metric: Dice loss directly optimizes the Dice coefficient or a similar similarity measure. This means that minimizing the Dice loss during training encourages the model to directly improve the segmentation accuracy according to the evaluation metric. In contrast, NLL loss optimizes the probability distribution and may not explicitly correlate with the segmentation performance metric.

Last Layer

For the models in Sections 3.1.1, 3.1.2, and 3.1.3, the final softmax layer is replaced with a sigmoid layer. Softmax layers are commonly used in multi-class classification problems, where there are more than two mutually exclusive classes. Sigmoid layers are primarily used in binary classification problems, where there are two classes. They are also used in multi-label classification tasks, where each input can belong to multiple classes simultaneously. In multi-label classification, each sigmoid output can be independently interpreted as the probability of the corresponding class being present.

Evaluation Stack

New functions are introduced like binary accuracy and area similarity, which give insight into the model's performance. The default IoU function, which was intended for multi-class segmentation, is also updated to support binary class segmentation. As most of the area in the image belongs to the negative class, IoU function is adjusted to only consider positive labels when evaluating the model.

4. Dataset

The commercialization of the satellite industry has led to an increase in the amount of satellite data being collected. Increasing accessibility of state-of-the-art deep learning algorithms have enabled developers to extract great insight from these satellite datasets.

Dataset used in this study is hosted on the AIcrowd website for a mapping challenge [15]. This dataset is built on SpaceNet (v1) [16]. SpaceNet (v1) is built for building detection. Private organizations such as CosmiQ Works, Radiant Solutions and NVIDIA have partnered to open source the SpaceNet dataset. The original SpaceNet dataset, introduced in the SpaceNet challenge in 2016, covers 2544 square kilometers of Rio De Janeiro with a **ground sampling distance** of 50cm. Ground sampling distance measures the area of ground covered in one pixel. The size of the full dataset is around 5.4 GB, with training (3.77 GB), validation (830 MB) and test (805 MB) splits consisting of 280741, 60317 and 60697 tiles respectively. Each tile is a 300x300 RGB image and all the annotations are in MS COCO format.

As the compute resources are limited, a small fraction of the dataset is used, consisting of only 300 training images and 50 validation images. Each image is of size 300x300 pixels. For the models used in this paper, the original labels which are in MS COCO format are changed to binary masks, where each pixel is labeled either 1 or 0 based on whether that pixel is a part of a building or not.

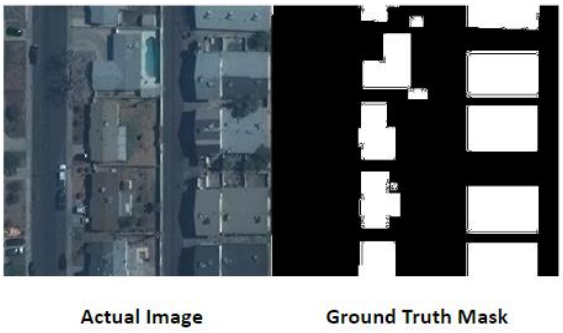


Figure 4: Ground truth image vs ground truth binary mask.

5. Training and Evaluation

The training and evaluation pipeline is fairly straightforward and is divided into two parts. The first training-evaluation pipeline is common for the first 3 models given in Table 1 and the second one is for the fourth model which is the UNet model.

In the first training-evaluation pipeline, a configuration file is fed as input. This file contains parameters of a specific model architecture and other details like root directory of the dataset, location where the weights need to be saved and training hyperparameters like batch size, learning rate, number of epochs etc. Using these configurations, a segmentation model object is created along with a data loader object. The loss

function is then passed to the segmentation object. Following this, optimizers are instantiated, and subsequently, the model is trained for the number of epochs specified in the config file. The model is saved after each epoch, and the best model is utilized for evaluation. A low evaluation threshold is maintained due to the lack of high confidence in the output probabilities. This can be tackled with further tweaking of the models.

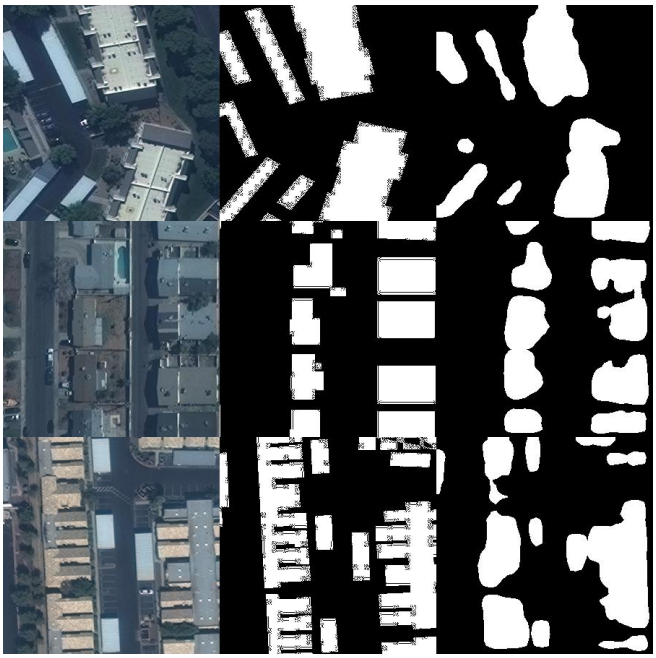
All the models are trained on 300 images for 10 epochs, and the model weights are saved after each epoch. Subsequently, the model is evaluated by calculating the IoU for the positive class. Training and evaluation are done on a laptop with Nvidia GTX1650 Ti with 4GB of VRAM.

6. Result and discussion

Training the dataset on four models requires some preprocessing as mentioned in sections above, the labels require converting to an appropriate format which is a binary segmentation mask. Due to limited compute resources, the original dataset is reduced to only 300 images. Additionally, models that are lightweight and have a small size are chosen.

As seen in Table 1. UNet outperforms all models despite being the one with a smaller number of parameters. The reason why UNet might be outperforming the ResNet18 model in this case could be attributed to its ability to capture fine-grained details and spatial context. The skip connections in UNet enable the model to leverage both high-level and low-level features effectively, helping to improve segmentation accuracy. Additionally, UNet's architecture is specifically designed for semantic segmentation tasks.

Actual Image Ground Truth Mask Predicted mask



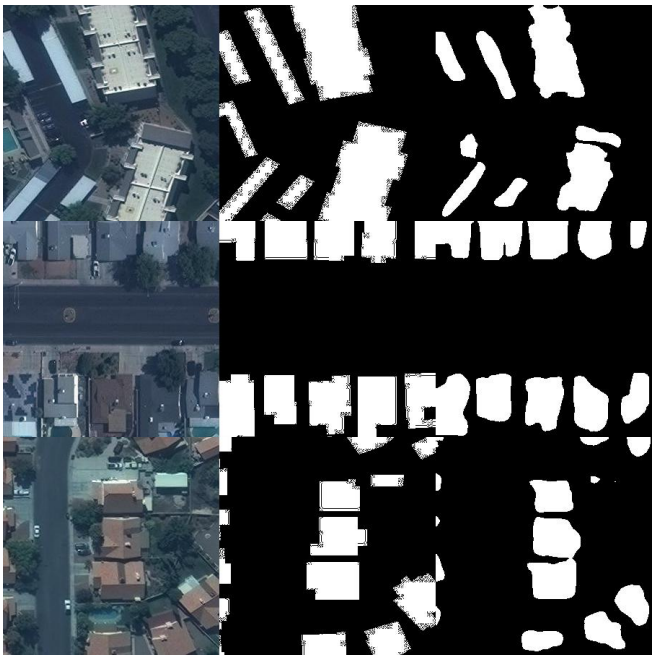
(a) ResNet18_PPM

(b) *HrnetV2*

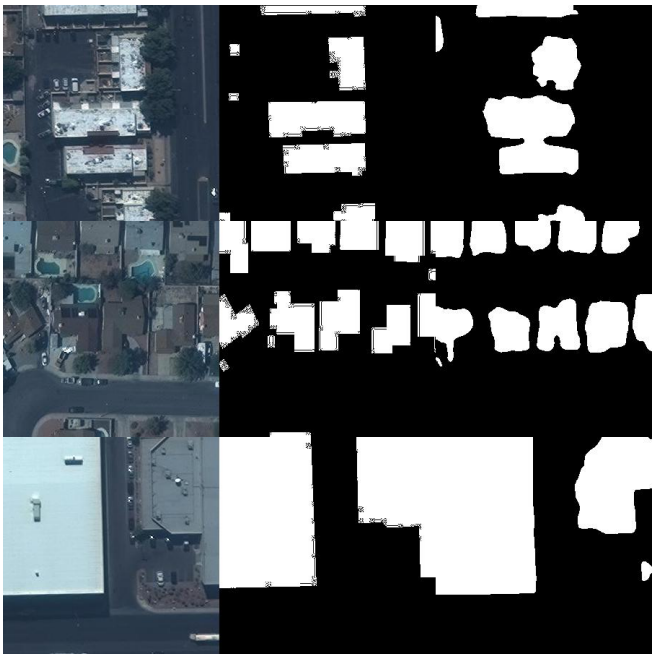
Figure 5: Comparing the binary masks predicted by (a) ResNet18_PPM and (b) HrnetV2 to the ground truth masks.

On the other hand, models like ResNet with a pyramid pooling module might be more suited for image classification. On the other hand, HRNetV2's architecture is specifically designed for pose estimation, therefore its low IoU score than UNet can be attributed to the different nature of the dataset.

Actual Image Ground Truth Mask Predicted mask



(c) ResNet50_Upernet



(d) *UNet (ResNet18 encoder)*

Figure 6: Comparing the binary masks predicted by (c) ResNet50_Upernet and (d) U-Net to the ground truth masks.

As for the function loss, dice loss is a suitable loss function that can be used to optimize the model weights, for reasons (discussed above) like handling class imbalance, similarity measure, robustness, gradient behavior.

As seen in Table 1 below, the models are compared based on mIoU scores (mean IoU) and the number of parameters (M = million).

TABLE I. MODEL COMPARISON

S.No	*Model Name*	*mIou*	*Num Params*	*Pixel accuracy (%)*
1	ResNet18_PPM	0.62	24M	64.5
2	ResNet50_Upernet	0.57	64M	54.4
3	HrnetV2	0.71	66M	78.3
4	Unet (ResNet18 encoder)	0.83	14M	85.1

7. Conclusion

This study showcases the effectiveness of deep learning models in accurately estimating building areas from satellite imagery in urban environments. Among the evaluated models, the UNet model with ResNet18 encoder emerges as the most suitable, achieving an impressive mean Intersection over Union (mIoU) score of 0.83.

The findings of this study contribute to the advancement of building area estimation techniques using deep learning. By leveraging multi-scale contextual information, high-resolution feature maps, and skip connections, significant improvements in accuracy have been achieved. Particularly, the UNet model with ResNet18 encoder demonstrates great potential for accurate and efficient building area estimation in urban areas.

These insights have practical implications for urban planners and policymakers, providing them with a reliable method for assessing building areas using satellite imagery. Accurate building area estimation plays a crucial role in addressing the challenges of urbanization and facilitating sustainable development. The outcome of this study offers valuable guidance for decision-making processes related to urban planning and resource allocation.

This study underscores the effectiveness of deep learning models for building area estimation from

satellite imagery. The identified UNet model with ResNet18 encoder stands out for its superior performance, with the potential to contribute to the progress in this field and providing actionable insights for urban planners and policymakers.

References

[1] LeCun, Y. *et al.* (1998) *Convolutional networks for images, speech, and Time Series: The handbook of brain theory and neural networks, Guide books*. Available at: https://dl.acm.org/doi/10.5555/303568.303704 (Accessed: 27 June 2023).

[2] Minaee, S. *et al.* (2020) *Image segmentation using Deep Learning: A Survey, arXiv.org*. Available at: https://arxiv.org/abs/2001.05566 (Accessed: 27 June 2023).

[3] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network." Available: https://arxiv.org/pdf/1612.01105.pdf

[4] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified Perceptual Parsing for Scene Understanding." Accessed: June 27, 2023. [Online]. Available: https://arxiv.org/pdf/1807.10221.pdf

[5] K. Sun *et al.*, "High-Resolution Representations for Labeling Pixels and Regions." Available: https://arxiv.org/pdf/1904.04514.pdf

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015. Available: https://arxiv.org/pdf/1505.04597.pdf

[7] Liu, X., Deng, Z. and Yang, Y. (2018) *Recent progress in Semantic Image Segmentation - Artificial Intelligence Review, SpringerLink*. Available at: https://link.springer.com/article/10.1007/s10462-018-9641-3 (Accessed: 27 June 2023).

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," arXiv.org, 2014. https://arxiv.org/abs/1411.4038

[9] Everingham, M. et al. (2009) The Pascal Visual Object Classes (VOC) Challenge - International Journal of Computer Vision, SpringerLink. Available at: https://link.springer.com/article/10.1007/s11263-009-0275-4 (Accessed: 27 June 2023).

[10] M. Cordts et al., "The Cityscapes Dataset for Semantic Urban Scene Understanding," arXiv:1604.01685 [cs], Apr. 2016, Available: https://arxiv.org/abs/1604.01685

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv.org, Dec. 10, 2015. https://arxiv.org/abs/1512.03385

[12] Lin, T.-Y. *et al.* (2017) *Feature Pyramid Networks for Object Detection, arXiv.org*. Available at: https://arxiv.org/abs/1612.03144 (Accessed: 27 June 2023).

[13] D. Zhu, H. Yao, B. Jiang, and P. Yu, "Negative Log Likelihood Ratio Loss for Deep Neural Network Classification," arXiv.org, Apr. 27, 2018. https://arxiv.org/abs/1804.10690 (accessed Jun. 27, 2023).

[14] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations," Springer Link, 2017. https://link.springer.com/chapter/10.1007%2F978-3-319-67558-9_28

[15] Mohanty, S.P. *et al.* (2020) *Deep learning for understanding satellite imagery: An experimental survey, Frontiers*. Available at: https://www.frontiersin.org/articles/10.3389/frai.2020.534696/full (Accessed: 27 June 2023).

[16] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, "SpaceNet: A Remote Sensing Dataset and Challenge Series," arXiv.org, Jul. 14, 2019. https://arxiv.org/abs/1807.01232 (accessed Jun. 27, 2023).