e-ISSN 2621-8062
Elipsoida : Jurnal Geodesi dan Geomatika
Vol 08 No 02, (2025)

ANALYSIS OF BUILDING DENSITY USING DEEP LEARNING MODEL SEMANTIC SEGMENTATION

Kris Junida Herindra Nuranda*, Moehammad Awaluddin, Firman Hadi
Department Geodetic Engineering, Faculty of Engineering, Diponegoro University, Indonesia
Corresponding author(s) e-mail: krisjunidahn@alumni.undip.ac.id*

(Received 1 June 2025, Accepted 26 November 2025, Published December 2025)

ABSTRACT

Densely populated settlements are one of the urban problems with building density that requires special attention. This research aims to detect and analyze the spatial distribution of building density, especially in detecting building density in residential areas using the Semantic Segmentation deep learning model method with a research dataset sourced from the entire DKI Jakarta Province area. The analysis was conducted using typology criteria in the form of building density levels based on PERMEN PUPR No. 14 of 2018 concerning the Prevention and Improvement of the Quality of Slums and Slum Settlements, which was processed through the Kaggle Notebook and Google Colaboratory platforms using the Python programming language and based on the U-Net architecture. The segmentation results show that using the U-Net architecture is capable of classifying image pixels with an accuracy of 70% in distinguishing between dense and Sparse buildings, which indicates fairly good accuracy performance. The output produced in this final project research is a web interface for detecting dense and Sparse buildings that can be used as a tool to aid in decision-making for regional planning. This research shows that the Semantic Segmentation deep learning model approach can be an efficient and objective solution in satellite image-based spatial analysis.

***Keywords:** Deep Learning, Building Density, Semantic Segmentation*

1. INTRODUCTION

Urbanization is one of the activities included in urban structural patterns that involve changes in land use. This condition has implications for changes in spatial patterns and environmental management in a region, so it can be concluded that urbanization is closely related to changes in urban spatial structures that have a significant impact on urban areas themselves.

Rapid urbanization is occurring in the Province of DKI Jakarta. Stiles the "Metropolitan City," this province is one of the areas that cannot escape the consequences of urbanization, one of which is the emergence of slums. This is triggered by the growth and inequality of development infrastructure facilities between rural and urban areas. As a result, the DKI Jakarta Province has become a magnet for urbanites to "migrate" (Rahmawati, 2014). The rapid growth of urbanization is also supported by a large population growth. In addition, the limited availability of land for building decent housing has also encouraged the spread of slums. Uncontrolled densification of settlements has also resulted in changes to the structure of existing housing in DKI Jakarta Province (Bolay, 2006).

This research focuses its analysis on the Building Density indicator in the DKI Jakarta Province area, which is one aspect of the criteria and typology of slums based on Article 19 paragraph of PERMEN PUPR No. 14 of 2018 concerning the Prevention and Improvement of the Quality of Slum Housing and Settlements, with the consideration that the determination of the Building Density class has a distance between buildings or roofs of less than 1.5 meters. Therefore, this research was conducted with the aim of analyzing Building Density detection using the U-Net architecture-based Semantic Segmentation deep learning model with using the DKI Jakarta Province area as the research dataset. Through the application of the U-Net architecture on Semantic Segmentation, it will be easier to identify, classify, and label each image pixel to determine the Building Density pattern based on the interpretation of the tested images. The results obtained through the application of the deep learning model Semantic Segmentation method for Building Density detection can be analyzed in terms of visual aspects and spatial analysis to better identify Building Density patterns based on the dataset used.

Previous research, such as that conducted by Rohit (2019) using the Semantic Segmentation method to monitor the development of slums in Mumbai, presented efforts to identify slum areas temporally, relying on visual interpretation and spatial analysis of settlement characteristics. However, this method tends to be subjective, especially in determining the research dataset, and the final output of the research is a closed-source platform. In addition, research that specifically examines Building Density based on Building Density parameters using a deep learning approach to Semantic Segmentation based on U-Net architecture is still very limited. This encourages the need for a new, more accurate approach, one of which is to use Semantic Segmentation based on U-Net architecture, which is capable of classifying image pixels precisely according to the Building Density category.

This research aims to analyze Building Density using Pleiades Neo satellite imagery data from 2023 covering the entire DKI Jakarta Province. The results of this research are implemented in the form of a Gradio-based web interface for visualizing the results of Building Density classification based on the provisions of PERMEN PUPR No.14 of 2018 concerning the Prevention and Improvement of the Quality of Slums and Slum Settlements.

In general, this research answers the main questions regarding the accuracy, final system integration, and results of the Building Density model analysis. The benefits of this research include scientific aspects, namely strengthening the role of Geodesy in satellite image processing and analysis, as well as engineering aspects, namely the development of a settlement classification system to support spatial planning. The scope of the research is limited to pixel-based Building Density analysis from Pleiades Neo 2023 satellite imagery without focusing on specific areas in DKI Jakarta Province or other regions.

This research produced a final output in the form of a web interface for Building Density detection that visualizes the condition of Building Density and its classes in an area with image testing provisions based on PERMEN PUPR No.14 of 2018 concerning the Prevention and Improvement of the Quality of Slums and Slum Settlements.

2. DATA AND METHODS

The research implementation for detecting building density conditions within a residential area, using the Province of DKI Jakarta as the research area, was conducted employing the following data and methodology.

2.1 Data and Location

This research uses primary data in the form of Pleiades Neo satellite imagery of DKI Jakarta Province in 2023 with a spatial resolution of 30 cm obtained from the Detail Spatial Plan (RDTR) of the DKI Jakarta Provincial Office of Public Works, Spatial Planning, and Land. This primary data will be used as a dataset that functions as the main data for training and testing the Semantic Segmentation model. However, it should be noted that this research was conducted only to design a building density detection system without referring to a specific area. The location of the case study in this research can be seen in Figure 1.

Figure 1. Location of research dataset

The primary data was then processed using software including QGIS Desktop Version 3.40.0 with the zooming scale of 1:300 to determine and create research datasets in accordance with PERMEN PUPR No.14 of 2018 concerning Prevention and Quality Improvement of Slums and Slum Settlements in the Building Density class, Label Studio web tools to annotate images in the research dataset, and the Google Colaboratory and Kaggle Notebook platforms for pre-processing and cloud-based processing.

2.2 Methodology

The research methodology for this research related to the implementation of semantic segmentation is visualized in a flowchart as shown in Figure 2. Meanwhile the U-Net architecture used in this research as shown in Figure 3.

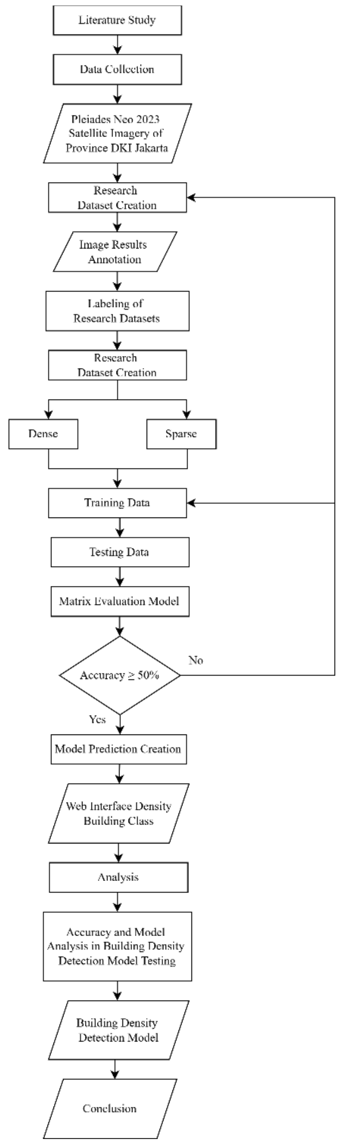

Figure 2. Research flowchart

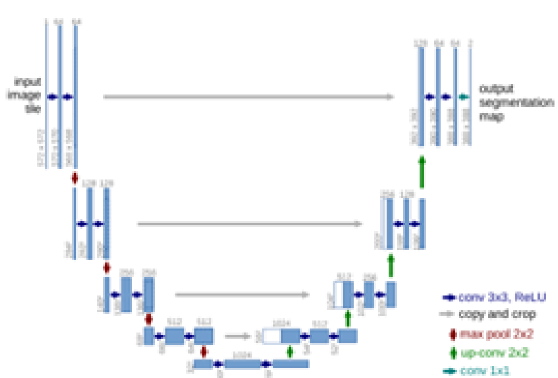

Figure 3. U-Net architecture

2.2.1 Preparing Phase

The preparatory phase of this research began with a literature study to enrich the theory relevant to the research using previous studies. In addition, the preparation also included collecting relevant data to be used in this research so that it could be used as a research dataset.

2.2.2 Data Collection (Image Collection)

The dataset creation phase begins with collecting image data that will later be further processed on the Label Studio web tools. The images collected are of dense and sparse buildings scattered throughout the DKI Jakarta Province. The collection of these images is in accordance with the provisions stated in PERMEN PUPR No.14 of 2018 concerning the Prevention and Improvement of the Quality of Slums and Slum Settlements, specifically in the Building Density category. In the Building Density class, the images captured are images of settlements with a distance between buildings of <1.5 meters because they are categorized as dense buildings according to PERMEN PUPR No.14 of 2018 concerning the Prevention and Improvement of the Quality of Slums and Slum Settlements. The application of these provisions was carried out using QGIS software by inputting Pleiades Neo 2023 satellite imagery of the DKI Jakarta Province area at a scale of 1:300 through the Measure Line tool by setting the area to be captured and then drawing a line between two buildings so that information about the distance between the buildings could be seen in the Measure window. The total of 2000 images were collected for each Building Density class.

2.2.3 Data Annotation

Data annotation is the process of labeling image data to provide specific information. In this research, image annotation was performed using the Label Studio web tool. Images uploaded to Label Studio were then labeled with two types of labels: dense settlements and sparse settlements.

2.2.4 Generate dan Export Dataset

Images that have been successfully annotated on the Label Studio web tool are then used to generate a dataset so that they can be used immediately by exporting them as Brush Labels as PNG files. This format was chosen because it is more effective for use with the Semantic Segmentation deep learning model, which can perform image segmentation well. The generated and exported dataset results will be saved in .h5 and .png formats.

2.2.5 Pre-Processing Dataset Python

Data pre-processing was performed on the Google Colaboratory platform, beginning with loading the dataset that had been created and installing packages on the Google Colaboratory environment. The dataset consisted of two types: an image dataset and a mask dataset. The initial phases of processing focused on handling the image dataset. This research uses the U-Net deep learning model architecture for image segmentation. The entire model training process is carried out with the support of the Tesla T4 GPU available on Google Colaboratory to improve computational efficiency and its speed.

2.2.6 Image Processing

Image processing was carried out after the entire image dataset was loaded into the Google Colaboratory environment, where a data pre-processing procedure was conducted to ensure that all datasets could be properly recognized by the model. The first phase involved dividing the images into smaller parts or patches using the patchify method, which aimed to simplify data complexity and enhance the model's generalization capability while minimizing the risk of overfitting in the deep learning model. All images that had been divided into patches were then normalized using the MinMaxScaler technique to standardize pixel value scales, thereby avoiding numerical instability during the training process. This procedure was applied consistently to both the image dataset and the mask dataset. Subsequently, the images were read using the OpenCV library and converted from BGR to RGB format to match the format used by PIL (Python Imaging Library). The conversion was performed using the Image.fromarray() function after color transformation with cv2.cvtColor. To maintain process stability, a try-except block was implemented to handle potential errors during data conversion. This process also included checking image dimensions and adjusting patch sizes according to the predefined value of 128×128 pixels. In the next phase, the image and mask datasets were converted into NumPy array format using np.array() to enable efficient data manipulation during the training process. The consistency of data quantity between images and masks was verified, and the array dimensions were modified using np.reshape() according to the model's input requirements. Visualization of image–mask pairs was performed using matplotlib to ensure alignment between the image and its corresponding label. Labeling of the mask dataset was conducted to classify Building Density into two categories: Dense Building and Sparse Building. This process involved reading mask files from the storage directory, binarizing the images using NumPy, and converting the visual information into a numerical format that could be processed by the model. The code or syntax used also displayed the total number of data successfully processed, as well as examples of labeling results as an initial validation step before training. Subsequently, a transformation from RGB-formatted mask images to numerical labels was performed using the rgb_to_label_with_text() function, with the color black (0,0,0) mapped to label 1 (Dense Building) and white (255, 255, 255) mapped to label 2 (Sparse Building). This process was carried out by creating an empty array with the same dimensions as the image and replacing pixel values based on the colors detected in the mask. The conversion results were stored in the variable labeled_data, which was used as the ground truth for model segmentation training.

2.2.7 Training dan Testing Dataset

The training and testing processes were carried out using the final processed image dataset with a total of 5,643 samples. The training was conducted on the Kaggle Notebook platform utilizing dual T4 GPUs. The deep learning architecture employed in this research utilSized a U-Net model; therefore, TensorFlow with the tf.Keras API was used to facilitate the construction of the U-Net architecture, which produces outputs in the format (batch_size, height, width, channels). The dataset for Building Density detection consisted of batches containing 5.643 images (128×128 pixels) in a two-class configuration, while the master training data were converted into three classes. The division of training and testing data was performed automatically with a 90:10 ratio using scikit-learn (train_test_split) with random_state=100 to ensure experimental reproducibility. Subsequently, the components required to build the CNN model were imported through Keras, including various layers such as Input, Conv2D, MaxPooling2D, upsampling (Conv2DTranspose), and merging layers such as Concatenate, BatchNormalization, Dropout, and Lambda. Model evaluation was conducted using the Jaccard Index matrix, employing a combined loss function consisting of dice loss, focal loss, and total loss to minimize loss values and improve prediction accuracy. Data augmentation on x_train was performed using ImageDataGenerator with a variety of transformations (rotation, shifting, flipping, and scaling) to increase data diversity and reduce the risk of overfitting. During the training phase, the model.fit() function was utilized with a batch size of 16 (for Building Density) for a maximum of 50 epochs, equipped with the EarlyStopping and ReduceLROnPlateau callbacks to adjust the learning rate when stagnation occurred. Model performance evaluation was carried out through visualization of the Intersection over Union (IoU) matrix and comparison of segmentation results between predictions and ground truth. After the training process was completed, the model was saved in HDF5 (.h5) format for further evaluation, prediction, and visualization using GradioUI. The IoU value was obtained using Equation 1.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} = \frac{Area\ (B_p \cap B_{gt})}{Area\ (B_p \cup B_{gt})} \quad (1)$$

2.2.8 Model Evaluation

The performance evaluation of the segmentation model was conducted to assess its generalization capability on previously unseen data and to ensure that the model did not experience overfitting. The assessment was carried out using the matrices of Accuracy, Precision, Recall, and F1-score, which were adjusted to the objective of object detection in the form of Building Density. The model evaluation process was visualized through evaluation syntax and the model network architecture, which illustrated the complete convolutional structure, including convolution, activation, dropout, max pooling, upsampling, and concatenation layers. These elements play a crucial role in feature extraction, dimensionality reduction, and accurate reconstruction of segmentation predictions. The calculations for the Accuracy, Precision, Recall, and F1-score matrices are expressed in Equations 2, 3, 4, and 5.

$$Accuracy = \frac{Correctly\ Classified\ Sample}{All\ of\ Sample} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Precision = \frac{True\ Sample}{Negative\ Classified\ Sample} = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{True\ Positive\ Sample}{Correctly\ Classified\ Sample} = \frac{TP}{TP+FN} \quad (4)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision \times Recall} = \frac{2 \times TP}{2 \times TP+FP+FN} \quad (5)$$

2.2.9 Image Prediction

The evaluation of prediction results is a crucial stage in assessing the model’s performance on test data, particularly in detecting Building Density. The evaluation was conducted by comparing the prediction results with the ground truth images to measure the model’s ability to accurately map relevant features. In the context of image segmentation, this comparison helps identify the extent to which the model can distinguish between Dense Building and Sparse Building areas. In addition to quantitative matrices, visual analysis through plotting of prediction results was also performed to identify potential errors such as over-segmentation and under-segmentation. This process was carried out by sampling data from the 2023 Pleiades Neo satellite imagery of the DKI Jakarta region using QGIS Desktop Version 3.40.0, in accordance with the provisions stated in the Ministry of Public Works and Housing Regulation (PERMEN PUPR) No. 14 of 2018 concerning the Prevention and Quality Improvement of Slum Housing and Slum Settlements, particularly in the context of Building Density.

2.2.10 Model Visualization

The TensorFlow/Keras environment on the Google Colaboratory platform was utilized to implement the Semantic Segmentation model. The installation of libraries such as segmentation-models, OpenCV, and Gradio was carried out to support the processes of modeling, image processing, and the development of a web-based user interface. The model used had been previously trained and saved in .h5 format, and it was executed with compatibility under the tf.keras framework. The main functions within this system include the computation of the Intersection over Union (IoU) using the jaccard_coef function as the primary metric for assessing segmentation accuracy, as well as the definition of a combined loss function consisting of Dice Loss and Categorical Focal Loss to address class imbalance in the data. The model receives input in the form of preprocessed satellite imagery (resized, RGB converted, and normalized) to generate predictions in the form of probabilistic tensors, which are subsequently transformed into pixel labels through the argmax function. The subsequent stages involve the visualization of segmentation results and the analysis of building classification based on the number of pixels identified as Dense or Sparse. To facilitate user interaction, an interactive interface was developed using Gradio on the Google Colaboratory platform, enabling users to upload images, process them using the model, and visually display the segmentation results along with their classification information. The code used to activate the visualization interface of the Building Density detection model can be accessed via the following link: https://github.com/krisjhn/FinalProject_DenseandSparseBuilding_with-SemanticSegmentation.git.

2.2.11 Model Prediction Results Evaluation

In this section, a final evaluation was conducted on the performance of the building image segmentation model that had been implemented in the web interface using GradioUI. The evaluation focused on the use of a confusion matrix to assess the classification accuracy of the model, which included the number of correct predictions (true positives and true negatives) and incorrect predictions (false positives and false negatives) for each class, namely Dense and Sparse. This confusion matrix provides a comprehensive overview of the model's ability to distinguish between each category and to identify possible biases or error patterns in the classification process. The construction of the confusion matrix began with random sampling consisting of 200 test samples, evenly divided into 100 samples for each Building Density category. This number was determined based on Paul Leedy's formula, which is a sample size estimation formula for proportions with a 95% confidence level and a 10% margin of error, resulting in a minimum sample size of 96. Therefore, the selection of 100 samples for each Building Density class was considered adequate and representative to support the model performance evaluation process through the confusion matrix in this Final Project research. The captured samples were then inputted into the available Input Image section of the web interface based on GradioUI, followed by an evaluation of the generated output. Instances of misclassification occurred during the system's detection of Building Density; therefore, an evaluation was conducted by calculating the accuracy of the tested samples through the confusion matrix.

3. RESULTS AND DISCUSSION

3.1 Building Density Class Accuracy Performance

The CNN model used in this research is the U-Net architecture for detecting Building Density. The resulting model was developed using a dataset derived from the 2023 Pleiades Neo satellite imagery of DKI Jakarta Province. The training process was conducted over 50 iterations, producing a graphical plot visualization as shown in Figure 4.

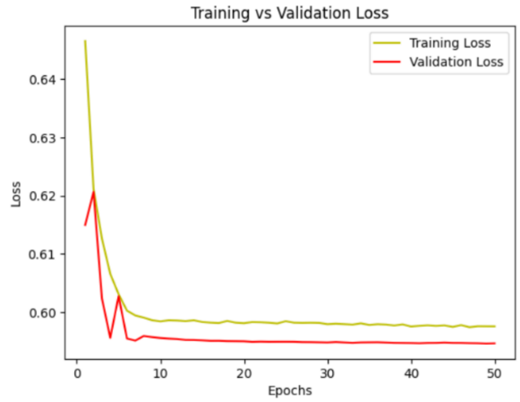

Figure 4. Training and validation loss plot of building density

Figure 4 presents a comparative graph of training loss and validation loss over 50 training epochs. The horizontal axis represents the number of epochs, while the vertical axis indicates the loss values. The yellow curve illustrates the training loss, whereas the red curve represents the validation loss. In the initial phase of training, both loss values are at high levels, indicating that the model is still in the early stages of learning. As the number of epochs increases, both curves exhibit a significant decrease, reflecting the model's ability to adjust its weights to minimize prediction errors. Minor fluctuations in the validation loss during the early training phase were likely caused by the model's adaptation to the different data distribution patterns in the validation set. After approximately 10 epochs, both curves began to display a stable pattern, with the training loss slightly lower than the validation loss, indicating model convergence and good generalization capability without signs of overfitting. This performance stability was reinforced through the implementation of two callback techniques, namely EarlyStopping and ReduceLROnPlateau. EarlyStopping monitored the validation loss and terminated training if no improvement was observed for 10 consecutive epochs, aiming to prevent overfitting and conserve computational resources. Meanwhile, ReduceLROnPlateau adaptively reduced the learning rate during stagnation, allowing the model to fine-tune its weight adjustments more smoothly. Nevertheless, during the training of the Building Density class, EarlyStopping was not activated because the validation loss continued to improve until the end of training at the 50th epoch. This finding confirms that the model continued to experience performance improvement and had not yet reached a stagnation point.

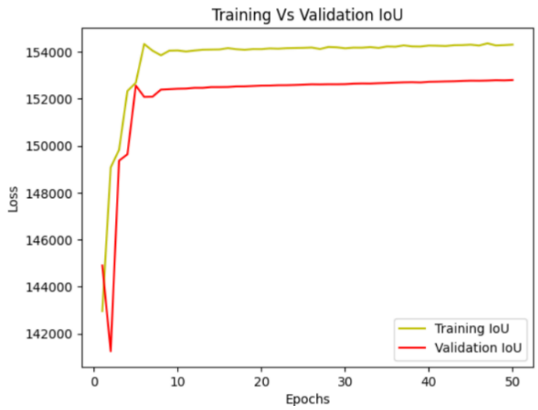

Figure 5. Training and validation IoU plot of building density

Figure 5 presents a comparison of the Intersection over Union (IoU) values between the training and validation data over 50 epochs for the Building Density segmentation model. IoU was used as the primary metric to measure the correspondence between the predicted results and the ground truth labels. At the beginning of the training process, both the training IoU and validation IoU exhibited a sharp increase, reflecting the model's effectiveness in adjusting its weights to produce more accurate segmentation results. However, in the early epochs, some fluctuations were observed, particularly in the validation IoU, which could be attributed to significant adjustments in the model's parameters. After approximately the 10th epoch, the graph demonstrated a stable pattern, with the training IoU consistently higher than the validation IoU. This difference indicates that although the model learned very well from the training data, its ability to maintain performance on unseen data was slightly lower. The gap between the two IoU values remained within a reasonable range and thus did not indicate severe overfitting. Nonetheless, the presence of this gap should still be noted, as it may suggest mild overfitting, where the model becomes overly adapted to the training data. Despite this, the relatively stable pattern of the validation IoU indicates that the model's generalization performance remained well preserved. A quantitative summary of the training IoU and validation IoU values throughout the training process is presented in Table 1.

Table 1. Training and validation IoU values of building density

Epoch	Training IoU	Validation IoU
1	0.9303	0.9084
2	0.9574	0.9798

Table 2. Training and validation IoU values of building density (continued)

Epoch	Training IoU	Validation IoU
3	0.9715	0.9896
4	0.9902	0.9931
5	0.9943	0.9992
6	1	0.9976
7	0.9967	0.9978
8	0.9982	0.9982
9	0.998	0.9984
10	0.9979	0.9984
11	0.9978	0.9985
12	0.998	0.9987
13	0.998	0.9987
14	0.9982	0.9988
15	0.9988	0.9989
16	0.9987	0.999
17	0.9984	0.999
18	0.9984	0.9991
19	0.9987	0.9991
20	0.9987	0.9992
21	0.9985	0.9991
22	0.9986	0.9992
23	0.9988	0.9993
24	0.9992	0.9993
25	0.9989	0.9993
26	0.9987	0.9993
27	0.9988	0.9993
28	0.9989	0.9994
29	0.9988	0.9995
30	0.999	0.9995
31	0.9989	0.9996
32	0.999	0.9996
33	0.9992	0.9996
34	0.9992	0.9996
35	0.9992	0.9996
36	0.9991	0.9996
37	0.9989	0.9997
38	0.9994	0.9997
39	0.9992	0.9997
40	0.9991	0.9997
41	0.9992	0.9997
42	0.9991	0.9998
43	0.9989	0.9998
44	0.9992	0.9998
45	0.999	0.9998
46	0.9994	0.9999
47	0.9994	1

Table 3. Training and validation IoU values of building density (continued)

Epoch	Training IoU	Validation IoU
48	0.9997	1
49	0.9991	1
50	0.9993	1

Based on Table 1, the Intersection over Union (IoU) values for both the training and validation data show a consistent upward trend as the number of epochs increases, approaching the maximum value of 1.0. This value indicates that the model achieved a very high segmentation performance, with prediction results that were almost completely aligned with the ground truth. In the early phase of training, the sharp increase in IoU values demonstrates that the model was able to quickly learn and understand the underlying patterns within the data. After reaching approximately the 5th to 10th epoch, the IoU values began to approach saturation, with relatively small changes in the subsequent epochs. At its peak, in the final epoch, the validation IoU reached a perfect value of 1.0, indicating that the model was able to produce highly accurate predictions on the validation data. Overall, the IoU range between 0.93 and 1.0 reflects an exceptionally high level of agreement between the predicted results and the reference data, suggesting that the model possesses near-perfect segmentation capability. In the context of image segmentation for building detection, these results indicate that the object boundaries predicted by the model closely match the actual structures on the ground. In addition to the IoU metric, other matrices such as Precision, Accuracy, Recall, and F1-Score were also used to obtain a more comprehensive evaluation of the model's performance in distinguishing and recognizing variations in Building Density levels. All these matrices were calculated based on Equations 2–5, and the complete results are presented in Table 4.

Table 4. Evaluation metrics of the building density detection model

Label class	Precission	Recall	F1-score
Dense building	0.7	0.93	0.8
Sparse building	0.71	0.3	0.42
	Accuracy		0.7
Macro average	0.7	0.62	0.61
Weighted average	0.7	0.7	0.66

Table 4 shows that the model achieved a Precision value of 70% for the Dense Building class and 71% for the Sparse Building class. This Precision value indicates that most of the model's predictions for each class tend to be correct. However, when viewed from the Recall perspective, a significant performance difference is observed, where the model demonstrates a much better ability to detect Dense Buildings with a Recall value of 93%, compared to only 30% for Sparse Buildings. This indicates that the model is more sensitive to the Dense Building class and has difficulty recognizing samples from the Sparse Building class, resulting in a high number of false negatives for Sparse Buildings. The imbalance between Precision and Recall is also reflected in the F1-score, which is 80% for the Dense Building class and only 42% for the Sparse Building class. The lower F1-score for the Sparse Building class indicates that the model has not yet achieved an optimal balance between detection and precision for that class. Meanwhile, the overall accuracy of the model is 70%, meaning that the model correctly classified 70% of the total tested samples. To provide a more comprehensive overview of performance, both macro average and weighted average were also utilized. The macro average F1-score of 61% shows that if each class is considered to have equal weight, the model's performance remains uneven. In contrast, the weighted average F1-score of 66% reflects that the model's performance is more influenced by the class with a larger number of samples, namely the Dense Building class. Overall, this analysis indicates that although the model demonstrates excellent spatial segmentation performance, as evidenced by its high IoU values, it still faces challenges in inter-class classification. The high IoU values suggest that the model's predicted building areas align closely with the actual conditions; however, the lower classification accuracy reveals weaknesses in distinguishing between building density classes, particularly in an imbalanced dataset. This occurs because of the differing approaches between IoU calculations, which focus on the overlap of areas regardless of object class, and classification metrics such as accuracy and F1-score, which evaluate the correctness of predicted labels between classes. Therefore, further model optimization is needed to enable more accurate and balanced recognition of the Sparse Building class.

3.2. Integration of the Building Density Detection System

Based on the results of this research, a web-based interface system (Gradio) was developed to detect the level of Building Density using satellite imagery. The system's sample testing follows the provisions stated in the Regulation of the Minister of Public Works and Public Housing (PERMEN PUPR) No. 14 of 2018, which defines buildings as *Dense* when the distance between structures is less than 1.5–3 meters. Two examples of the testing results are presented in Figure 6 & 7.

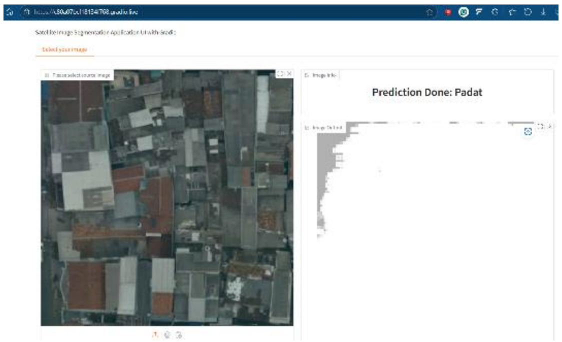

Figure 6. Density building detection system interface visualization

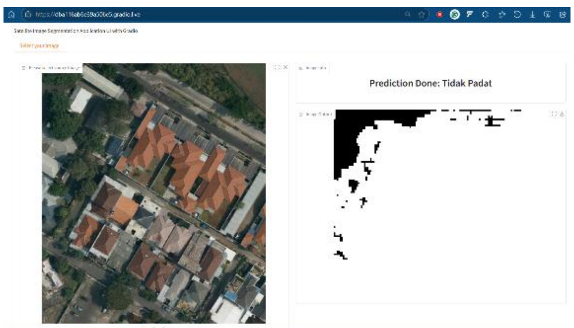

Figure 7. Sparse building detection system interface visualization

In Figure 6, the distance between buildings ranges from 0.265 to 0.444 meters, allowing the system to accurately classify the area as "Dense." Conversely, in Figure 7, where the distances between buildings ranges from 3.465 to 4.465 meters, the system classifies the area as "Sparse" in accordance with the regulatory reference. This system employs a deep learning–based Semantic Segmentation model that has undergone a training process to recognize the spatial distribution of buildings within satellite imagery. The satellite images input into the system are processed to generate an output in the form of building density classification, along with a visual representation of dense areas highlighted in white. The results shown in Figures 6 and 7 demonstrate that the system can accurately distinguish residential areas based on building density. The system's success in differentiating between Dense and Sparse areas indicates a strong generalization capability of the model across various spatial patterns in imagery. This suggests that the model was trained on a representative dataset, enabling it to effectively recognize diverse settlement characteristics across different regions.

3.3 Evaluation of the Building Density Detection Model

In addition to the model evaluation metrics, a confusion matrix for the Building Density classes is also presented. This confusion matrix provides information regarding the detection performance of the Building Density classes within the GradioUI-based web interface, indicating whether the detection system correctly predicts the Building Density class or experiences misclassification, as shown in Figure 8.

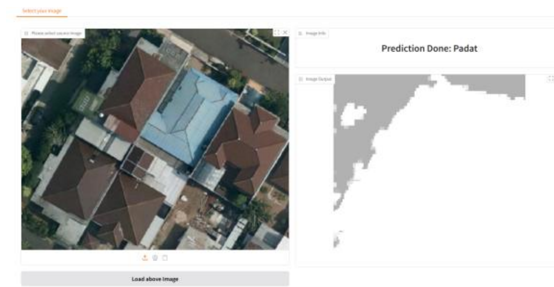

Figure 8. Model prediction errors of building density

Figure 8 illustrates an example of a misclassification made by the model in classifying Building Density levels. In the Input Image, the satellite imagery depicts a residential area with several buildings located relatively close to each other but not forming a uniformly dense pattern. However, the prediction result displayed in the Output Image shows that the model classified most of the area as Dense Buildings, indicated by white-shaded regions. This misclassification demonstrates that the model tends to generalize the presence of buildings as Dense areas, even though not all parts visually meet the Density criteria. This indicates that the model still struggles to distinguish spatial density comprehensively, particularly in areas with uneven building distribution or small open spaces between structures. Another possible cause of this misclassification is the limited variation in the training data, where the model predominantly learned to recognize satellite imagery characteristics of highly dense or very sparse areas, making it less sensitive to regions with intermediate characteristics. These findings are consistent with the quantitative evaluation presented in Table 3, where the omission error for the Sparse Building class is relatively high, supporting the observation that the model is more effective at recognizing Dense Buildings than Sparse ones. The confusion matrix report for the Building Density classes is presented in Table 5.

Table 5. Confusion matrix of density building class

Label class	Sample		Rows total	Ommision error
	Dense building	Sparse building		
Dense building	61	3	64	0.05
Sparse building	24	12	36	0.67
Columns total	85	15	100	
Commission error	0.28	0.2		
Overall accuracy	0.73			
Kappa accuracy	0.33			

Based on the report check results presented in Table 5, the model achieved a high true positive value for the Dense Building class, with 61 out of 64 samples correctly identified. This indicates that the model has strong capability in recognizing Dense Building objects, which may be attributed to the more consistent visual features and the dominance of data from this class. Conversely, the high false negative value for the Sparse Building class, with 24 out of 36 samples misclassified, indicates that many objects in this class were incorrectly predicted as Dense Buildings. This directly contributes to the high omission error value of 0.67. Meanwhile, the relatively low false positive rate, with only 3 out of 64 samples, shows that the model rarely misclassified an object as a Dense Building, resulting in a commission error of 0.28, which can be considered fairly good. Furthermore, the overall accuracy of 0.73 is relatively high; however, the Kappa value of only 0.33 suggests that part of the model's success may be influenced by the imbalance in the number of samples rather than purely by its generalization capability. Overall, the model's performance in classifying Dense Building areas can be considered good, as indicated by the low omission error of 0.05 and a high number of correct predictions (61 out of 64 samples). However, the model's performance in recognizing Sparse Building areas remains suboptimal, as reflected by the high omission error, indicating that many samples from this class were misclassified as Dense Buildings. In addition, the results imply that the model's classification ability is not significantly better than random classification. This limitation is likely caused by the imbalance in sample distribution between classes, where Dense Building areas are more easily detected by the system compared to Sparse Building areas. In this research, the success of the model in classifying Building Density demonstrates that the system possesses a good generalization capability toward spatial patterns within satellite imagery. This indicates that the model was trained using a sufficiently representative dataset, enabling it to recognize the characteristics of regions with varying levels of Building Density effectively.

4. CONCLUSIONS AND SUGGESTIONS

The Building Density detection model developed using a U-Net-based Semantic Segmentation approach demonstrated excellent performance with an Intersection over Union (IoU) value approaching 1, yet it still encountered difficulties in accurately identifying the Sparse Building class. This is evidenced by the lower Recall and overall accuracy for that class, as well as a Kappa accuracy of 33%, indicating the model's reliability is not yet optimal. Evaluation via the confusion matrix confirmed the model performs very well in detecting Dense Buildings but has a high error rate for Sparse Buildings, likely due to class imbalance and visual similarities between the categories. Despite this limitation, the model was successfully implemented in a web-based system using Gradio, capable of classifying building density into two main categories with accurate segmentation outputs.

For further development, it is recommended to enhance dataset quality by incorporating more diverse satellite imagery from various regions and applying data augmentation techniques to improve the model's generalization capability. Furthermore, the model should be applied to a specific area, such as at the sub-district level, to evaluate its effectiveness under real-world conditions. It is also advisable to conduct a comparative analysis with other image-based deep learning methods or to perform experiments using different frameworks, such as TensorFlow and PyTorch, to assess relative performance and potential variations in results.

ACKNOWLEDGMENTS

The author would like to express sincere gratitude to the academic supervisor for their guidance and direction, to the Head of the Department and all members of the Department of Geodetic Engineering, Faculty of Engineering, Diponegoro University, for their support throughout the research process, and to the author's parents and friends for their continuous encouragement and motivation.

REFERENCES

Bolay, J. (2006). *Slums and Urban Development: Questions on Society and Globalisation.* Swiss: The European Journal of Development Research.

Chollet, F. (2021). *Deep Learning with Phyton.* New York City: Manning Publications Co.

Ciresan, D., Giusti, A., Gambardella, L.M., & Schmidhuber, J. (2012). Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. *NIPS*, 2852-2860.

Cristina, & Andreanus, J. (2018). Aplikasi Sistem Informasi Geografis Berbasis Web Pemetaan Lokasi Tempat Makan Vegetarian di Kota Batam. *Jurnal Telematika, Vol. 13 No. 1, Institut Teknologi Harapan Bangsa Bandung*, 55-56.

Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision, 88 (2)*, 303-338.

Herold. M., Joseph, S., & Clarke, K. (2002). The Use of Remote Sensing and Lanscape Metrics to Describe Structures and Changes in Urban Land Uses. *Environment and Planning A*, 1443-1458.

Hicks, S., Strümke, I., Thambawita, V., Hammou, M., Riegler, M.A., Halvorsen, P., & Parasa, S. (2022). On Evaluation Metrics for Medical Applications of Artificial Intelligence. *Scientific Reports*, 2.

Kohli. D., Sliuzas, R., & Stein, A. (2016). Urban Slum Detection Using Texture and Spatial. *Journal of Spatial Science*, 405-426.

Liu. H., Huang, X., Wen, D., & Li, J. (2017). The Use of Landscape Metrics and Transfer Learning to Explore Urban Villages in China. *Remote Sensing MDPI*, 1-23.

McGarigal, K., & Marks, B. (1995). FRAGSTATS: Spatial Pattern Analysis Program for Quantifying Landscape Structure. *USDA Forest Service General Technical Report PNW-351*, 110-119.

Menteri PUPR RI. (2018). *Peraturan Menteri Pekerjaan Umum dan Perumahan Rakyat Republik Indonesia No.14/PRT/M/2018 Tentang Pencegahan dan Peningkatan Kualitas Terhadap Perumahan Kumuh dan Permukiman Kumuh.* Jakarta: Sekretariat Negara. Menteri PUPR RI. (2018). *Peraturan Menteri Pekerjaan Umum dan Perumahan Rakyat Republik Indonesia No.14/PRT/M/2018 Tentang Pencegahan dan Peningkatan Kualitas Terhadap Perumahan Kumuh dan Permukiman Kumuh Pasal 25:1.* Jakarta: Sekretariat Negara.

Menteri PUPR RI. (2018). *Peraturan Menteri Pekerjaan Umum dan Perumahan Rakyat Republik Indonesia No.14/PRT/M/2018 Tentang Pencegahan dan Peningkatan Kualitas Terhadap Perumahan Kumuh dan Permukiman Kumuh Pasal 26:1.* Jakarta: Sekretariat Negara.

Menteri PUPR RI. (2018). *Peraturan Menteri Pekerjaan Umum dan Perumahan Rakyat Republik Indonesia No.14/PRT/M/2018 Tentang Pola Penanganan Perumahan dan Permukiman Kumuh Pasal 44:4.* Jakarta: Sekretariat Negara.

Menteri PUPR RI. (2018). *Peraturan Menteri Pekerjaan Umum dan Perumahan Rakyat Republik Indonesia No.14/PRT/M/2018 Tentang Pola Penanganan Terhadap Perumahan Kumuh dan Permukiman Kumuh Pasal 36:2.* Jakarta: Sekretariat Negara.

Mugiraneza. T., Hafner, S., Haas, J., & Ban, Y. (2022). Monitoring Urbanization and Enviromental Impact in Kigali, Rwanda Using Sentinel-2 MSI Data and Ecosystem Service Bundles. *International Journal of Applied Earth Observations and Geoinformation*, 1-16.

Rahman, M., Alamsah, D., Darmawidjaja, M.I., & Nurma, I. (2017). Klasifikasi untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN). *J. Inform, 11 (1)*, 36.

Raj, A., & Mitra, A. (2024). Deep Learning for Slum Mapping in Remote Sensing Images: A Meta-analysis and Review. *arXiv*, 2-17.

Rohit, S. (2019). *Mapping and Monitoring Slum Growth Using Deep Learning.* Mumbai: Esri R&D Center India.

Verma, D., Jana, A., & Ramamritham, K. (2019). Transfer Learning Approach to Map Urban Slums Using High and Medium Resolution Satellite Imagery. *ELSEVIER*, 3-11.

Williams, T., Wei, T., & Zhu, X. (2019). Mapping Urban Slum Settlements Using Very High-Resolution Imagery and Land Boundary Data. *IEEE Journal*, 2-10.

Wurm, M., Stark, T., Zhu, X.X., & Weigand, M. (2019). Semantic Segmentation of Slums in Satellite Images Using Transfer Learning on Fully Convolutional Neural Networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59-69.

APPENDIX

Appendix 1 – Code Repository and Prediction Results

All program codes, segmentation models, and prediction results can be accessed through the following link: https://github.com/krisjhn/FinalProject_DenseandSparseBuilding_with-SemanticSegmentation.git.

This repository includes all preprocessing scripts up to the generation of the output from the Building Density Detection system.