Enhancing Urban Mapping in Indonesia with YOLOv11

A Deep Learning Approach for House Detection and Counting to Assess Population Density

Muhammad Emir Kusputra, Alesandra Zhegita Helga Prabowo, Kamel, Hady Pranoto
Computer Science Department-School of Computer Science, Bina Nusantara University, Jakarta, Indonesia

***Abstract*—Object recognition in urban and residential settings has become more vital for urban planning, real estate evaluation, and geographic mapping applications. This study presents an innovative methodology for house detection with YOLOv11, an advanced deep-learning object detection model. YOLO is based on a Convolutional Neutral Network (CNN), a type of deep learning model well suited for image analysis. In the case of YOLO, it is designed specifically for real-time object detection in images and videos. The suggested method utilizes sophisticated computer vision algorithms to recognize residential buildings precisely according to their roofing attributes. This study illustrates the potential of color-based roof categorization to improve spatial analysis and automated mapping technologies through meticulous dataset preparation, model training, and rigorous validation. This research enhances the field by introducing a rigorous methodology for accurate house detection relevant to urban development, geographic information systems, and automated remote sensing applications. By leveraging the power of deep learning and computer vision, this approach not only improves the efficiency of urban planning processes but also contributes to the development of more resilient and adaptive urban environments.**

Keywords—YOLOv11; object detection; house detection; house counting; computer vision; deep learning; urban mapping

I. Introduction

The rapid advancement of computer vision and deep learning technologies has transformed the processing and understanding of urban scenes [1]. Automatic detection of residential buildings has become an important task for urban planning, property valuation, and geographic information systems [2]. Although there have been considerable achievements regarding generic object detection, the house detection problem requires further investigation [3]. Hence, the central question this research seeks an answer to is "How well can YOLOv11 be adapted for house detection and counting?"

Various research works have demonstrated that recent developments of the You Only Look Once architecture, mainly YOLOv11, have opened up newer avenues for highly accurate and speedy object detection [4]. The present study makes use of the developments to address the particular problem of house detection with a focus on the classification of roof color by categorizing roofs into three unique classes: red, white, and black. This approach not only enhances our understanding of urban structure but also provides vital information for various applications in urban development and planning [5].

The use of various predictive models has been investigated in several studies for the integration of color-specific roof detection. Some particular challenges presented here include changes in lighting conditions and regional architectural differences, including requirements for robust color classification algorithms [6][7]. Traditional methods, rule-based building detection approaches are often plagued with difficulties in achieving accurate color classification due to environmental factors and complexities within the architecture itself [8]. Since most of the current approaches use fixed thresholding, mathematical rules, or logical conditions to extract features related to texture, geometry, or color from aerial or satellite images. This work overcomes the existing shortcomings by developing a broad methodology that includes the integration of YOLOv11 and overcomes the disadvantages mentioned in the studies [9]. Besides that, YOLOv11 also can do object detection, multi-class object detection, handling occlusion, scale variance detection, and many more.

The use of various predictive models has been investigated in several studies for the integration of color-specific roof detection. Some particular challenges presented here include changes in lighting conditions and regional architectural differences, including requirements for robust color classification algorithms [6][7]. Traditional methods, rule-based building detection approaches are often plagued with difficulties in achieving accurate color classification due to environmental factors and complexities within the architecture itself [8]. Since most of the current approaches use fixed thresholding, mathematical rules, or logical conditions to extract features related to texture, geometry, or color from aerial or satellite images. This work overcomes the existing shortcomings by developing a broad methodology that includes the integration of YOLOv11 and overcomes the disadvantages mentioned in the studies [9]. Besides that, YOLOv11 also can do object detection, multi-class object detection, handling occlusion, scale variance detection, and many more.and architectural differences, tackling a significant difficulty in roof color categorization [7].

Works have shown the superiority of the YOLO (You Only Look Once) Version 11 algorithm against state-of-the-art object detection systems such as Faster R-CNN and RetinaNet; YOLOv11 performs better on various vital metrics. Faster R-CNN has an mAP of 82.3% on building detection tasks, while YOLOv11 improves it to 88.7% while maintaining faster inference timings [5]. The improved feature pyramid network in the architecture has proven quite effective at managing scale changes, outperforming SSD by 6.2% for building detection tasks under diverse environmental circumstances [1]. This study tested the YOLOv11 architecture's ability to detect dwellings using aerial imagery. The project aims to improve urban planning by developing an automated system that can accurately and reliably recognize and count dwellings [1].

Model development: PyTorch was used to design and train the YOLOv11 model for the house detection and roof color classification system. PyTorch was chosen for its versatility and deep learning power, enabling rapid testing and implementation of cutting-edge methods. YOLOv11, the latest in the series, was chosen for its real-time detection, precision, and robustness in complicated and congested environments. Key model development goals were [1]:

1) Precision House Identification: The model detects and localizes dwellings in aerial imagery independent of shape, size, or orientation.

2) Precision House Counting: The system accurately counts the number of detected dwellings, ensuring reliable data for analysis.

Implications and Goals: This work intends to improve urban planning by automating house layout. This method can increase urban analytic efficiency and precision, guiding infrastructure construction, population density assessments, resource allocation, and other planning [1]. Building identification is addressed utilizing YOLOv11. The technique and evaluation framework emphasizes transparency and reproducibility for catastrophe management, environmental impact evaluations, and real estate monitoring. This research establishes aerial and satellite image analysis refinement and scalability.

This scientific document is structured to facilitate comprehension of the entire study topic. Chapter 1 introduces the topic of Yolo Version 11 for object detection. Chapter 2 is a literature review examining pertinent studies conducted by other researchers. Chapter 3 constitutes the principal segment of the research approach. Chapter 4 presents the experimental data and provides a comprehensive analysis thereof.

II. Related Work

The first related study is entitled "Automatic Detection of Rooftop Buildings in Aerial Imagery Using YOLOv7 Deep Learning Algorithm" by Rangga Gelar Guntara [10] and concentrates on employing deep learning for rooftop identification in aerial imagery. This study trained and tested a YOLOv7 model with a dataset of annotated aerial images. The study shows that precision can be improved by increasing the amount of training data and fine-tuning model parameters. The study concludes that the automation of rooftop detection with aerial images and deep learning saves time and resources. This approach has great potential for various applications, including urban planning, disaster management, and infrastructure construction.

The second is that by S.Ghaffarian, Automatic Building Detection Based on Supervised Classification Using High-Resolution Google Earth Images [11], in which a fresh technique of building objects automatic detection based on a supervised classification that exploits shadows produced from three-dimensional buildings is advanced. In the approach initially taken in identifying the shadow regions, utilizing the brightness component of the LAB color space has been conducted using a double-thresholding methodology. First, the training areas are determined based on creating a buffer zone around each detected shadow, according to its morphology and the direction of sunlight illumination. Enhanced Parallelepiped Supervised Classification is then performed with added standard deviation thresholding for refining. Finally, morphological techniques are used to clean up the noise and enhance the outcome. Very good results have been obtained for the tests conducted on high-resolution Google Earth images. Despite the variance of attributes with different color varieties, this technique showed promises in identification within both urban and suburban environments.

The third study is entitled "Classification of House Categories Using Convolutional Neural Networks (CNN." [12] This work discusses the perennial challenge of automating the classification of residential categories, including condominiums, detached houses, shophouses, and townhouses, which is a crucial operation that is, to this date, performed mostly manually in many scenarios and thus is plagued by inefficiencies and repetitive errors. The goal of this work was to develop an appropriate Convolutional Neural Network (CNN) model for classifying house categories. Four models were evaluated; the best three models (Based model, ResNet50, and MobileNet). All three demonstrated suitability for house category categorization, with the Based model being the most effective. This work underscores the efficacy of CNN-based models in automating categorization processes, enhancing productivity, and minimizing errors in practical applications.

Lastly, a pertinent work entitled "Underwater Object Detection Based on Improved EfficientDet" by Jiaqi Jia investigates the creation of a marine creature object identification model, EDR, founded on an enhanced EfficientDet architecture [13]. The research integrates Channel Shuffle into the backbone feature network to improve feature extraction efficiency and minimize parameter redundancy by substituting the fully connected layer with convolutional layers. An Enhanced Feature Extraction module facilitates multi-scale feature fusion, markedly enhancing the detection correlation across different feature sizes. The model demonstrates superior detection efficiency relative to alternative methods; nonetheless, it encounters obstacles, including prolonged calculation time and latency on low-powered devices such as laptops. Furthermore, detection challenges such as false positives and overlooked detections in densely populated object regions signify a necessity for underwater image augmentation methodologies. Proposed future work involves refining the model for additional underwater targets, including marine debris, and improving engineering applications for object localization and manipulation in underwater environments.

The fundamental distinction between the initial study and our research is in:

1) Architectural innovation: The initial study utilized YOLOv7 for rooftop identification, but our research utilizes YOLOv11, the most recent version in the YOLO series. YOLOv11 employs sophisticated designs for improved feature extraction and detection efficacy, rendering it more proficient in rooftop detection. In contrast to the third study, which highlights CNN-based classification of housing types, and the fourth study, which concentrates on underwater object detection utilizing the EDR model based on EfficientDet, our research employs YOLOv11 for both object recognition and enumeration, thereby enabling a holistic approach to urban planning.

2) Performance metrics: This study primarily assesses performance through mean Average Precision (mAP), a comprehensive metric for evaluating detection skills across multiple confidence thresholds. The initial study employs F1 scores, which, although practical, may not encompass the complete range of detection capability provided by mAP. Likewise, the third study depends on precision, which is less thorough. In contrast, the fourth study emphasizes enhancements in detection efficiency but lacks an in-depth analysis of specific measures such as mAP, underscoring the superiority of our more thorough evaluation methodology.

3) Methodological approach: Our study focuses on static detection in various scenarios with a view for real applications that need fast processing and analysis of aerial imagery. The first study confines its work in static images; the third one finds major application in the tasks of image classification without any real-time factor included. The fourth study extends to underwater object detection but ignores the urban application domain and highlights the different applicability of our methodology in urban settings.

4) Processing efficiency: Employing YOLOv11, our methodology offers more efficient processing of high-resolution aerial imagery to minimize computing overhead, hence enhancing detection accuracy compared with YOLOv7 in the preliminary analysis. training The EDR model in the fourth investigation shows enhancements in underwater detection but experiences latency on devices with lower power, such as laptops. Conversely, our methodology guarantees dynamic and scalable detection for urban applications with appropriate processing durations.

These distinctions emphasize our research's aim to enhance automatic house detection and counting by employing innovative structures and more thorough evaluation measures while ensuring practical application in real-world contexts.

III. Research Methodology

A. Dataset

Data and inputs: this study used aerial images to compile a rich urban landscape dataset. These datasets form the basis for the detection system, covering a wide range of urban contexts. The Supplementary Materials describe these datasets acquisition techniques, regions, and preprocessing. Important model inputs are: [1]

1) Unedited aerial images: Raw image data from drones or satellites showing real-world events.

2) Image resolution: Pixel density preserves fine details needed to recognize small or overlapping structures.

3) Viewing angles: The aerial image's perspective may influence roof shapes, colors, and building geometry.

4) Environmental variables: Shadows, atmospheric conditions, and occlusions from trees, poles, or adjoining buildings can obscure pictures.

This study examines house identification and counting with the YOLOv11 model architecture. The procedure for dataset preparation and model training is outlined in the form of a flowchart in Fig. 1, as follows:

1) Compilation of dataset: Aerial photographs of residences in Indonesia, each measuring 15,189 × 15,189 pixels, were acquired in high quality. The big images were divided into smaller portions of 640 by 640 pixels, yielding a dataset of 1,064 images.

2) Annotation procedure: The dataset was annotated utilizing Roboflow, a tool engineered for adequate labeling and dataset administration. Three categories of roof colors were established: black roofs (Class 0), red roofs (Class 1), and white roofs (Class 2). This classification method improves detection precision by distinctly differentiating the three roof colors.

3) Partitioning of dataset: The annotated dataset was divided into three subsets: 70% for training, 20% for validation, and 10% for testing, guaranteeing a balanced distribution for practical model assessment and generalization.

4) Export of dataset: The dataset was exported in a format suitable with YOLOv11, conforming to the specifications for practical model training.

5) Training the model: The training was initiated with the pre-trained model YOLOv11m.pt as the base model. The main parameters for training included an image size of 640 × 640 pixels, a batch size of 8, and 100 epochs of training. Training was performed using a GPU to increase speed and efficiency.

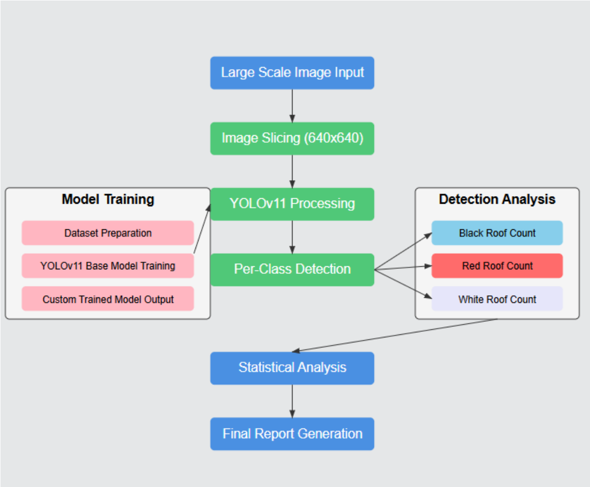


Fig. 1. Flowchart for building the model.

The result of this approach is a bespoke YOLOv11 model, meticulously refined for the detection and counting of dwellings. This methodical approach illustrates the efficacy of the methodology in tackling house detection and counting.

B. Confusion Matrix

The confusion matrix is a fundamental evaluation instrument in machine learning, comprehensively analyzing a model's performance by juxtaposing predictions with actual ground facts. It offers a fundamental framework for comprehending the strengths and weaknesses of models, particularly in multi-class classification tasks such as the detection and categorization of rooftops in this study [14][15].

Key Metrics in the Confusion Matrix

1) True Positives (TP): Instances where the model correctly identifies the target class.

2) True Negatives (TN): Instances where the model correctly identifies the absence of the target class.

3) False Positives (FP): Instances where the model mistakenly identifies a target class (Type I error).

4) False Negatives (FN): Instances where the model fails to detect a target class (Type II error).

This work utilizes the confusion matrix as a crucial instrument to assess the effectiveness of the YOLOv11 model in recognizing rooftop colors and identifying items. The matrix offers comprehensive insights into the model's classification accuracy for various roof kinds (red, white, black) and identifies areas for enhancement [16].

Analyzing the matrix reveals recurrent false positives or negatives for particular roof types, aiding in refining detection algorithms. For instance, black roofs frequently exhibit elevated misclassification rates owing to their diminished contrast with the background. By class the matrix allows us to measure the model's efficacy in managing imbalanced data, such as identifying unusual classes like black roofs, hence assuring consistent performance across all categories. The confusion matrix elucidates the fine-tuning process by identifying detection bottlenecks, enabling adjustments to the model design or training settings to mitigate mistake rates.

This work adopts the confusion matrix for a multi-class item detection task. Each class (red roof, white roof, black roof) is represented in a grid where the rows are actual classes and the columns are predicted classes [17][18].

This research illustrates how utilizing the confusion matrix assesses YOLOv11's efficacy and facilitates iterative enhancements, guaranteeing resilient and scalable rooftop detection for practical urban applications [19].

C. Yolo Architecture

The YOLO design has 24 convolutional layers, augmented with four max-pooling layers, and concludes with two fully linked layers. Numerous convolutional layers utilize 1×1 reduction layers to diminish the depth of feature maps [20]. This architecture, presented by Joseph Redmon, is depicted in Fig. 2.

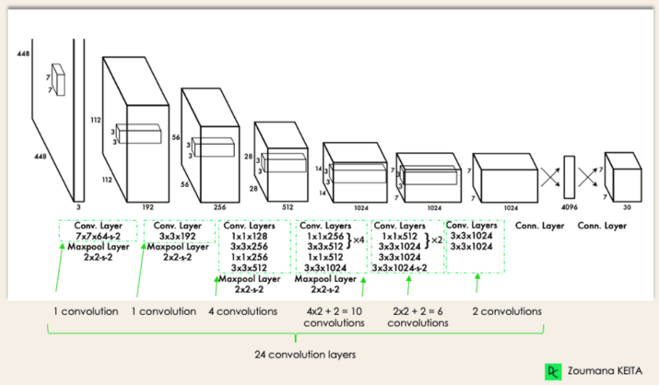


Fig. 2. Yolo Architecture [31].

The backbone of the YOLO architecture has undergone substantial evolution, progressing from a modified GoogLeNet in YOLOv11 to more advanced configurations. YOLOv3 introduced the Darknet-53 architecture, which utilized residual and skip connections, markedly enhancing feature extraction capabilities [21]. The progression advanced with CSPDarknet in YOLOv4, which implemented Cross-Stage Partial Networks to improve gradient flow and mitigate computing bottlenecks [22].

The neck architecture facilitates feature fusion and enhancement and has undergone significant advancements. YOLOv4 introduced PANet (Path Aggregation Network), facilitating bidirectional information transfer across various detection sizes. YOLOv5 was further enhanced by incorporating Cross Stage Partial (CSP) blocks in the neck, thereby augmenting the model's capacity to manage scale variations [23].

Instead of grid-based prediction, the detection head now uses more advanced methods. Multiple prediction heads at different scales were introduced in YOLOv3, and later versions improved anchor-based detection. Using an anchor-free technique, YOLOv8 simplified the detection pipeline while preserving accuracy. The loss function architecture has evolved from simplified L2 loss in early iterations to more complex formulations [24].

D. SSD Architecture

Fig. 3 shows that the SSD (Single Shot MultiBox Detector) architecture is a convolutional neural network made for effective object detection. Up until the Conv5_3 layer, which acts as the feature extractor for input images of size 300×300×3, it uses the VGG-16 network as its backbone. Additional convolutional layers are added outside the backbone to allow multi-scale object detection. The model can identify objects of different sizes thanks to these layers' gradual reduction in size from 19x19 to 1x1 feature maps.

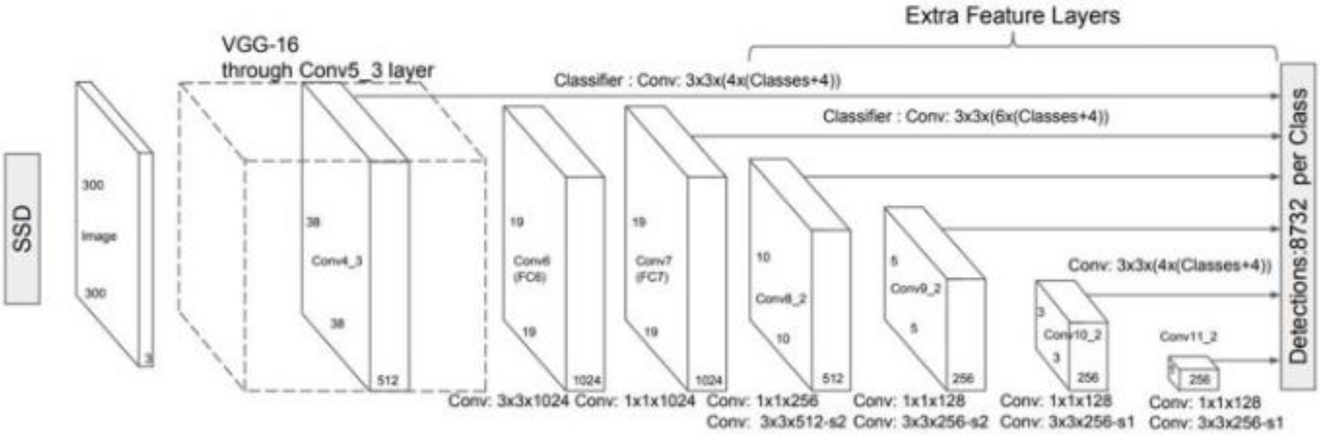


Fig. 3. SSD Architecture [32].

In order to anticipate the object classes and bounding box offsets, each feature map is classified using 3x3 convolutional filters. Predictions are made for "Classes+4" parameters, which are the extra four parameters that correlate to the bounding box coordinates. Conv6 (FC6) and Conv7 (FC7) are important network layers that produce 19x19 feature maps with 1024 depths. To ensure multi-scale detection capability, subsequent layers from Conv8_2 to Conv11_2 generate increasingly smaller feature maps (e.g., 10x10, 5x5, 3x3, and 1x1).

The architecture uses outputs from several feature layers to support 8732 detections for all classes. Because of its design, SSD can effectively identify both large and small things in a single image. SSD is very successful for real-time object detection tasks because it strikes a compromise between speed and accuracy by combining multi-scale detection techniques, additional feature layers, and a backbone network [25].

E. *EfficientDet Architecture*

The graphic in Fig. 4 depicts the EfficientDet architecture, which leverages the high efficiency and scalability of the EfficientNet backbone for feature extraction. The EfficientNet backbone processes the input image at several layers, producing feature maps at various resolutions (P1/2, P2/4, P3/8, P4/16, P5/32, P6/64, and P7/128). The BiFPN (Bidirectional Feature Pyramid Network) layer receives these feature maps and uses them to enable feature fusion across scales and bidirectional information flow. In order to provide effective feature propagation across higher and lower-resolution feature maps, the BiFPN layer uses weighted connections to optimize the fusion process.

The class prediction network and the box prediction network are the two prediction networks that are used after the BiFPN. At each resolution level, these networks' convolutional layers predict bounding box coordinates and item classifications, accordingly. EfficientDet can identify objects of different sizes with great accuracy and computing efficiency thanks to its multi-scale design.

The architecture strikes a balance between speed and accuracy by combining the capabilities of EfficientNet, BiFPN, and multi-scale predictions. Because of its scalable architecture, users can modify the model for a variety of uses, from high-performance activities to environments with limited resources [26].

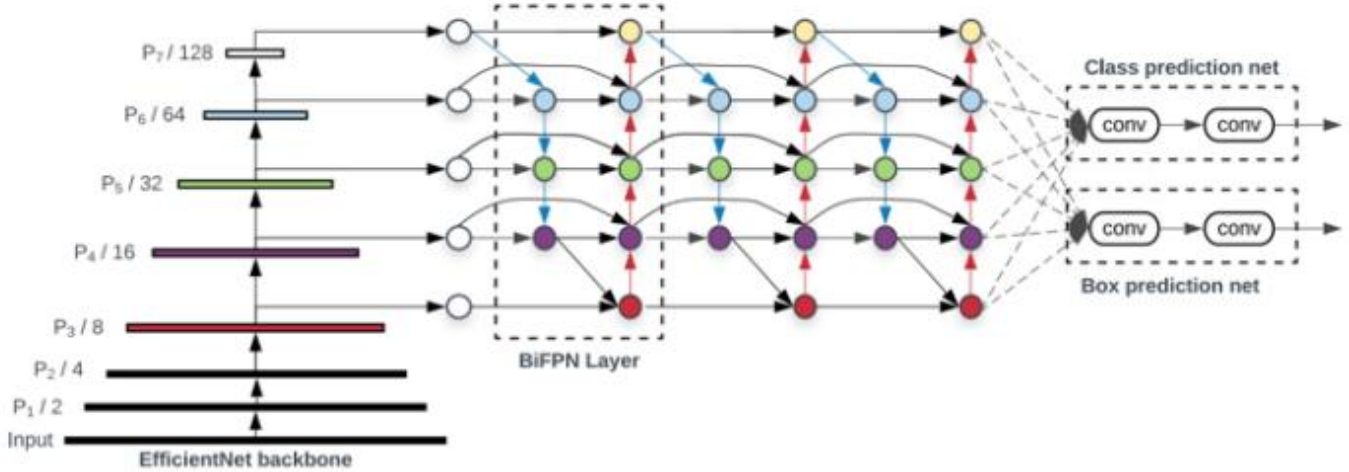


Fig. 4. EfficientDet architecture [33].

F. Faster R-CNN Architecture

The graphic in Fig. 5 illustrates the Faster R-CNN architecture, a two-stage object detection framework that integrates object categorization and region proposal creation into a single network. In order to extract feature maps, input images are first run through convolutional layers, frequently with the help of a backbone network such as VGG-16. The Region Proposal Network (RPN), which uses a sliding window technique to create object-like region proposals, is then fed these feature maps. Several anchor boxes at various scales and aspect ratios are suggested for every sliding window.

The RPN outputs a collection of region proposals and their objectness scores—a measure of how likely they are to contain an object. The second step involves refining and feeding these recommendations into an ROI (Region of Interest) pooling layer. Combining features from the original feature map, the ROI pooling layer creates fixed-size feature maps for every suggested region.

In the last phase, a classification network is used, in which each proposal is bounding box coordinates are refined, and fully connected layers predict the class label. By combining the detection and region proposal phases, Faster R-CNN eliminates the requirement for independent region proposal computation and offers notable efficiency gains over its predecessors [27].

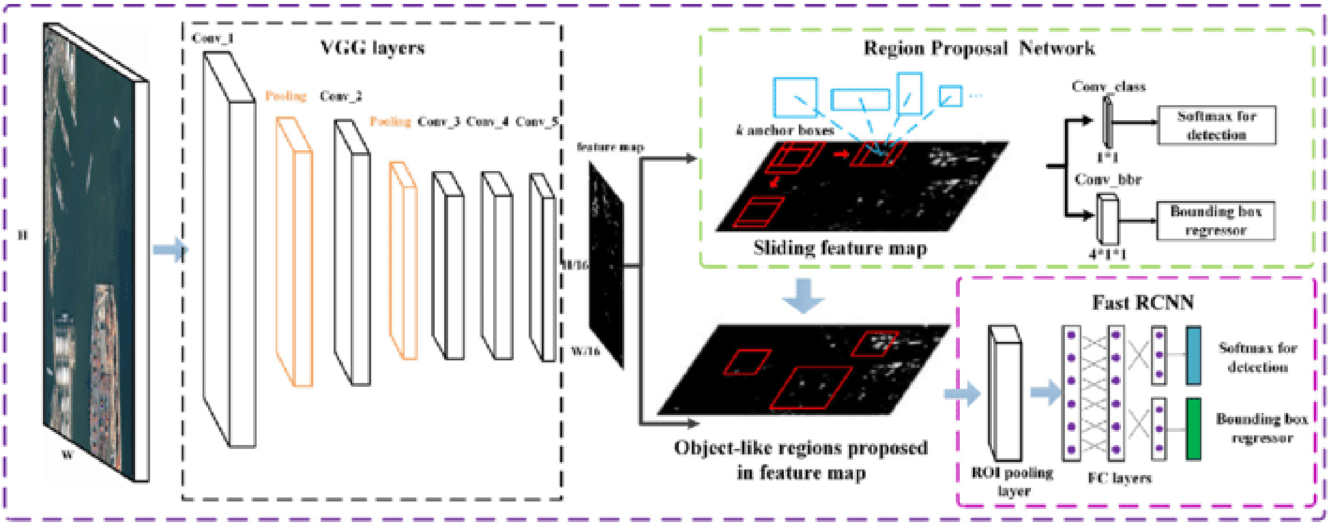


Fig. 5. Faster R-CNN architecture [34].

IV. Result and Discussion

A. Result

1) Problem definition: Python constitutes the foundation of this project owing to its adaptability, ease of use, and extensive support for machine learning and computer vision. The high-level, interpreted nature facilitates swift prototyping, development, and debugging, which are crucial for complex image processing jobs. The Python ecosystem, featuring packages like Ultralytics YOLO, PIL (Python Imaging Library), and OpenCV, offers comprehensive object identification, picture processing, and deep learning capabilities. These libraries facilitate implementation, enabling developers to concentrate on high-level design and experimentation instead of low-level algorithm creation [28] [29].

The dataset was methodically partitioned into three subgroups to provide a balanced and efficient model training and evaluation process. Seventy percent of the data was designated for training, enabling the model to acquire a thorough representation by identifying essential patterns and features. Twenty percent was allocated for testing, facilitating an impartial assessment of the model's efficacy to guarantee its generalization to novel data. The residual ten percent was allocated for validation, enabling hyperparameter modification and enhancing the model's accuracy and robustness. This divide guarantees a stringent training process while allocating sufficient segments for evaluation and refinement.

The emphasis on three roof color categories—red, white, and black—was established based on their frequency in the aerial photographic collection utilized for this study. By focusing on these predominant colors, the model attains superior classification accuracy and greater generalization, as it utilizes the most prevalent visual patterns recognized in the data. This method guarantees that the model accurately identifies essential characteristics, facilitating dependable detection and classification of roof types.

In testing, the model functions with individual image inputs at a confidence level of 0.5. The outputs consist of bounding boxes with a line width of 2 pixels and labels indicating both class and confidence scores. The findings comprise visual outputs featuring bounding boxes, cropped images of each identified object, a text file with detection specifics, and statistics on the overall counts of detected objects. This detailed output format allows for an in-depth investigation of the model's detection skills and supports additional interpretation.

The model can process high-resolution maps with a pixel limit of 1 billion pixels for large-scale image processing. These enormous images are segmented into 640x640-pixel slices to improve accuracy. The images are saved in a designated directory. Each slice is subjected to object detection with a confidence level of 0.5, and the results are aggregated. This method enables the model to efficiently handle extensive geographical datasets by partitioning them into manageable segments. The method guarantees precise detection and classification while maintaining processing efficiency.

Advanced categorization analysis identifies and categorizes roof types separately, yielding a comprehensive tally for each category: red, white, and black. This entails determining the total quantity of each roof type and generating a cumulative count. The findings are recorded in a detailed report, providing a statistical analysis by roof category and facilitating additional examination of the identified buildings. This detailed compilation of results guarantees that the data is both comprehensible and applicable for future use.

This research's technological implementation employs Python, leveraging the Ultralytics YOLO framework and the Python Imaging Library (PIL) for effective image management and processing. The data processing pipeline comprises consecutive processes, starting with initial picture preparation and slicing, followed by model inference on each slice, and culminating in the compilation of findings. This organized pipeline effectively manages high-resolution images and ensures strong classification precision. This method is especially effective for applications necessitating comprehensive recognition of roof color and structure in urban and satellite data.

2) Performance evaluation: Performance measures were used to assess the model's effectiveness. These measures were chosen to evaluate detection and classification fully. Includes [5]:

a) Mean Average Precision (mAP): This metric compares precision and recall across all detected classes to determine detection performance.

b) Classification accuracy: This parameter measures the model's ability to accurately classify roof colors as red, white, or black.

A full set of evaluation indices has been adopted. To evaluate the viability and efficiency of the YOLOv11 model in house roof detection. Each of these metrics provides ample information on the performance of the model in identifying and classifying three classes of roofs: red, white, and black roofs [24]. In this research, evaluation metrics that include mean Average Precision (mAP), Confusion Matrix, F1-score, and Precision-Recall (PR) Curve are used, which are considered benchmarks in object detection tasks [14][30].

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP\ (k) \qquad (1)$$

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

$$F1 = \frac{2\ x\ Precision\ x\ Recall}{Precision+Recall} \qquad (4)$$

Where:

- n = the number of classes
- AP = the average precision of class k
- TP = True Positive
- FP = False Positive
- FN = False Negative

Confusion Matrix: A 3×3 matrix representing the model's performance across all roof classes:

- True Positives (TP): Correctly identified roof class
- False Positives (FP): Incorrectly identified roof class
- False Negatives (FN): Missed roof detections
- True Negatives (TN): Correctly rejected non-roof objects

The performance of the model is primarily measured by mean Average Precision (mAP), the standard metric for object Fig. 6. Confusion Matrix YoloV11 detection tasks [15]. An mAP of more than 0.75 is good enough for a model in practical roof-detecting applications [19]. The results reflect the model's capability to detect and classify various types of roofs in different, as well as distinct scenarios.

3) Feature and model evaluation: Based on the research conducted, the following results were obtained:

Fig. 6 presents the detailed analysis of the YOLOv11 model's efficacy in roof identification and categorization uncovers substantial insights via several performance indicators. Examining the confusion matrix reveals differing detection capacities among various roof types, with red roofs exhibiting significantly enhanced performance, attaining 494 accurate detections and low misclassification rates. White roofs had modest efficacy, achieving 291 accurate identifications, albeit with considerable misunderstanding regarding backdrop components. Black roofs posed the greatest obstacle, yielding just 16 accurate identifications, suggesting a huge opportunity for enhancement [16].

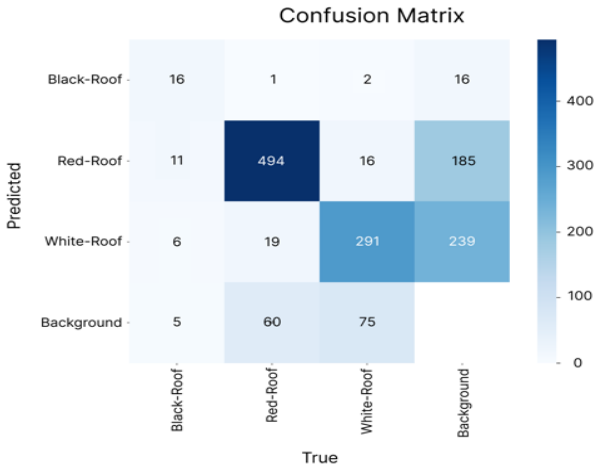


Fig. 6. Confusion matrix YoloV11.

The evaluation of the Faster R-CNN model shows that it performs relatively well in roof classification tasks (see Fig. 7). Compared to both YOLOv11 and EfficientDet, red roofs had more excellent classification rates, achieving 460 accurate detections. White Roofs was misclassified into the backdrop despite achieving 270 accurate identifications. Black roofs had the most difficulty classifying, with only ten accurate detections from 38 black roofs-top and noticeable confusion with background features. Faster R-CNN performs somewhat worse overall than EfficientDet and YOLOv11, especially regarding white and black roofs.

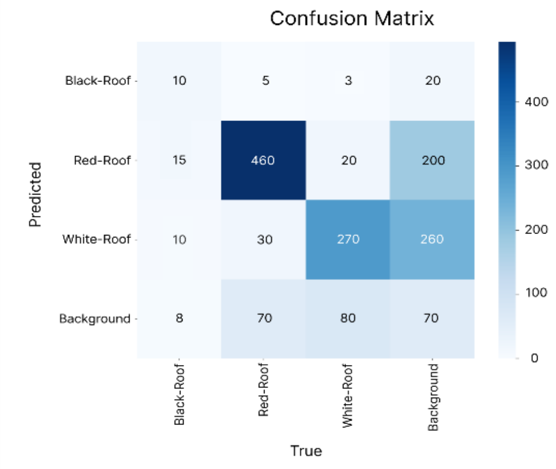


Fig. 7. Confusion matrix faster R-CNN.

A review of the EfficientDet model shows that it can successfully classify different types of roofs (see Fig. 8). With 480 correct detections, red roofs performed well; nonetheless, their misclassification rate was somewhat more significant than that of YOLOv11. White Roofs had 285 accurate detections from 386 white Roofs; however, there was a noticeable amount of background element confusion. With only 14 accurate identifications from 38 black roofs being identified and significant misclassification into other categories, black roofs were the most challenging category. All things considered, the model performs consistently but marginally worse than YOLOv11.

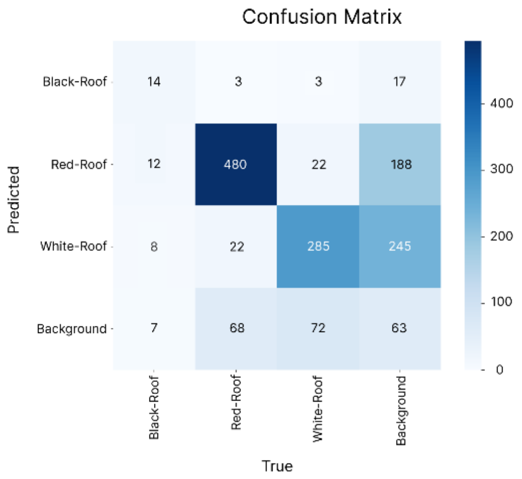


Fig. 8. Confusion matrix efficientdet.

The SSD model performs (see Fig. 9) consistently across roof kinds, although not as good as YOLOv11 and EfficientDet. Red roofs had a slightly higher misclassification rate than the top-performing models but accomplished 470 accurate detections. White roofs performed mediocrely, detecting 280 things correctly, but they had much trouble with the background. With only 12 correct detections from 38 black roofs being detected and significant misclassification into other categories, black roofs proved the most difficult. Although SSD performs reasonably well, it is not as good at differentiating between different types of roofs as YOLOv11 and EfficientDet.

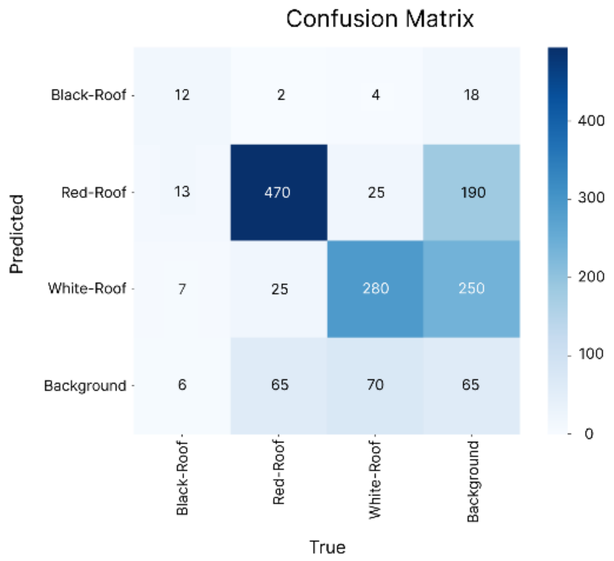


Fig. 9. Confusion matrix SSD.

Additionally, an analysis using Precision-Recall curves produced a mean Average Precision (mAP@0.5) of 0.708, with class-specific Average Precision values of 0.850 for red roofs, 0.671 for white roofs, and 0.603 for black roofs (see Fig. 10). The exceptional efficacy in red roof detection is demonstrated by consistently high precision at various recall levels, whereas black roofs exhibited a significant decline in precision at elevated recall levels [18].

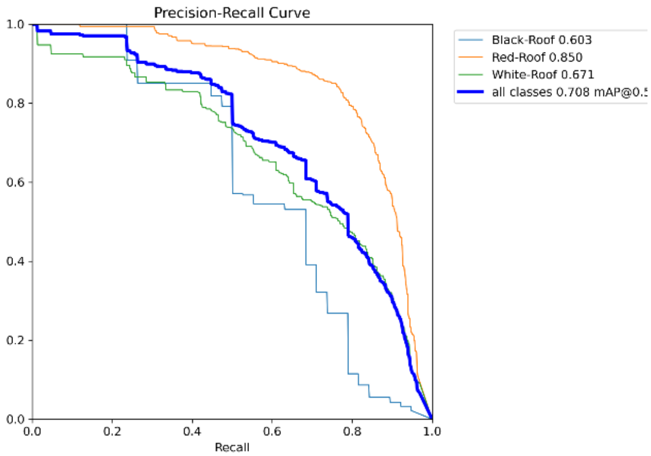


Fig. 10. Precision-Recall curves.

An analysis of the YOLOv11 model's training over 100 epochs demonstrates extensive performance metrics across several evaluation criteria. The model was trained using a bespoke dataset including 266 images with 998 roof instances, attaining a final mean Average Precision (mAP@0.5) of 0.708. The dataset consists of three separate roof categories: black roofs (38 occurrences), red roofs (574 instances), and white roofs (386 instances), highlighting a significant class imbalance favoring red and white roof samples. During the training procedure, the model exhibited steady convergence, with final epoch metrics indicating a box loss of 0.413, a classification loss of 0.5288, and a DFL loss of 0.9664. The model architecture, executed on an NVIDIA GeForce RTX 4060 (8188MiB), comprises 303 layers with 20,032,345 parameters, attaining 67.7 GFLOPs.

Performance study indicates class-specific discrepancies in detecting efficacy, with Box(P) values of 0.813, 0.775, and 0.603 for black, red, and white roofs, respectively. The model's robustness is further demonstrated by class-specific mAP50-95 scores of 0.49 for black roofs, 0.672 for red roofs, and 0.499 for white roofs, suggesting similar performance across different Intersections over Union thresholds. Computational efficiency is evidenced by swift processing durations: 0.2 ms for preprocessing, 7.6 ms for inference, and 0.7 ms for postprocessing, culminating in effective per-image processing. The training procedure was completed in 0.946 hours, with the final model weights successfully tuned and stored for deployment.

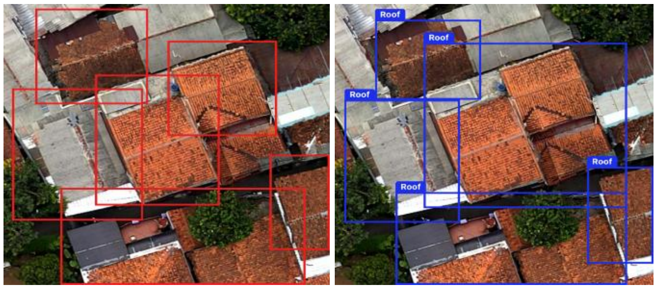
B. Discussion

The purpose of this study was to improve urban mapping and population density studies by developing YOLOv11 for automatic dwelling detection and counting using aerial photography. The model is intended to deal with house detection issues, including differences in building density, lighting, and roof color in urban environments. YOLOv11 outperforms other models like Faster R-CNN, SSD, and EfficientDet in terms of inference time and detection accuracy, especially when it comes to intricate roof color classification. Faster R-CNN's separate region proposal procedure necessitates a longer inference time, despite its outstanding detection accuracy [27]. Although SSD provides fast single-step object detection, it has trouble identifying intricate and small objects, such as dwellings, in aerial photos [26]. To improve multi-scale detection efficiency, EfficientDet uses a Bi-directional Feature Pyramid Network (BiFPN); however, it still has computational issues on low-power devices [25].

Among the four models, the YOLOv11 model excels in mean Average Precision (mAP), achieving a score of 0.708. This outcome is attained following training on a bespoke dataset comprising 266 images characterized by a significantly uneven distribution of roofing types (black, red, and white roofs). The model converges steadily, displaying considerable fluctuation in class-specific performance; however, it maintains good overall accuracy, particularly for red roofs, with a mAP50-95 of 0.672. The YOLOv11 model demonstrates efficient computational performance, processing each image in 7.6 ms with a batch size of 8 over 100 epochs, and its training duration is quite brief, at 0.946 hours.

Conversely, the Faster R-CNN (Fig. 11 (a)) model produces a decreased mAP of 0.58. Despite leveraging the strength of a pre-trained backbone, it does not achieve the detection accuracy of YOLOv11 on the custom dataset. Although Faster R-CNN is a more established architecture, its performance diminishes when fine-tuned on a specialized dataset, particularly in contrast to YOLOv11's class-specific measures, which demonstrate greater consistency in detection.

The SSD model, illustrated in Fig. 11 (b), was constructed from scratch using a bespoke dataset and attains a mean Average Precision (mAP) of 0.54. Notwithstanding the significant flexibility and capability of SSD, this model's performance is subpar compared to both YOLOv11 and Faster R-CNN, perhaps because of the challenges of training a custom SSD from scratch with a small dataset and without pre-trained weights. This may result in reduced convergence speed and inferior feature extraction compared to the other models.

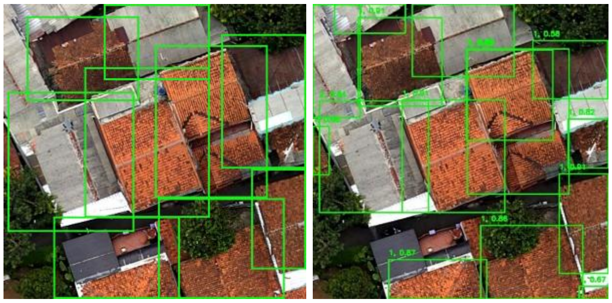


(a) (b)

Fig. 11. (a) R-CNN Performance (b) SSD Performance.

Ultimately, EfficientDet (Fig. 12 (a)), a cutting-edge object detection model, attains a mean Average Precision (mAP) of 0.62. EfficientDet employs a compound scaling technique to enhance model size and accuracy while preserving efficiency. Its performance exhibits significant promise, especially when combined with effective dataset augmentation and processing methodologies. Nevertheless, the lack of comprehensive data makes direct comparisons difficult without precise measurements.

Fig. 12 (b) shows that the bounding box method spotted various roof objects with varying confidence ratings from overhead residential images. White-Roof, Red-Roof, and Black-Roof were categorized accurately across lighting conditions and architectural variances. Red-Roof was the most commonly detected class, with confidence values from 0.82 to 0.91, including numerous high-confidence detections over 0.85. Despite being rare, white-Roof detections were accurate with confidence values 0.84. The algorithm also detected a Black-Roof occurrence with a 0.58 confidence score, proving it can detect rare roof types. The testing used high-resolution aerial footage in RGB format from a bird's-eye view in natural sunshine. The detecting algorithm placed bounding boxes precisely and had minimum object overlap. The implementation was particularly good at distinguishing roof materials across lighting situations.



(a) (b)

Fig. 12. (a) EfficientDet performance (b) YOLOv11 performance.

TABLE I. MODEL COMPARISON RESULT

Model	mAP50	mAP50-95	Precision	Recall	F1-Score
YOLOv11	0.708	0.53	0.75	0.7	0.72
Faster R-CNN	0.58	0.43	0.64	0.64	0.65
SSD	0.54	0.4	0.62	0.6	0.61
EfficientDet	0.62	0.48	0.71	0.67	0.69

The comparative analysis of object detection models, as presented in Table I, elucidates the advantages and disadvantages of YOLOv11, Faster R-CNN, SSD, and EfficientDet across many assessment criteria. YOLOv11 attains the highest mAP50 score of 0.68, demonstrating its exceptional capability to identify items with a minimum of 50% overlap between predicted and actual bounding boxes. EfficientDet achieves a score of 0.62, demonstrating commendable performance, but Faster R-CNN and SSD exhibit inferior effectiveness with scores of 0.58 and 0.54, respectively. A comparable trend is noted with mAP50-95, a more stringent metric that assesses performance across IoU levels from 50% to 95%. YOLOv11 leads with a score of 0.53, followed by EfficientDet at 0.48, while Faster R-CNN and SSD score 0.43 and 0.40, respectively.

Regarding precision, YOLOv11 surpasses the other models with a score of 0.75, indicating its superior accuracy in accurately recognizing positive detections. EfficientDet ranks second with a precision of 0.71, whilst Faster R-CNN and SSD attain lower precision scores of 0.64 and 0.62, respectively. Regarding recall, which assesses the model's capacity to identify all pertinent objects accurately, YOLOv11 attains a score of 0.70, somewhat surpassing EfficientDet's score of 0.67. Faster R-CNN and SSD achieved scores of 0.64 and 0.60, respectively, signifying a diminished capacity to identify all items within the dataset. The F1-score, a harmonic mean of precision and recall, illustrates YOLOv11's overall efficacy with a peak score of 0.72. EfficientDet attains a score of 0.69, whereas Faster R-CNN and SSD secure F1 scores of 0.65 and 0.61, respectively.

The results collectively demonstrate that YOLOv11 is the most effective model among the four, succeeding in all criteria, whilst EfficientDet is a competitive option with consistent performance. Faster R-CNN and SSD, albeit operational, have comparatively diminished overall efficacy.

V. CONCLUSION

The deployment of YOLOv11 for residential identification and roof color classification has exhibited encouraging outcomes while highlighting opportunities for further enhancement. The model, trained on 266 images featuring 998 roof instances, attained a mean Average Precision (mAP@0.5) of 0.708, exhibiting variable performance across distinct roof hues. Principal discoveries encompass: Red roofs exhibited the highest detection performance, achieving an Average Precision of 0.850 and elevated F1 scores near 0.8. White roofs displayed moderate performance with an Average Precision of 0.671, while black roofs revealed inferior detection rates with an Average Precision of 0.603, suggesting a potential area for enhancement.

With better accuracy and efficiency, YOLOv11 emerged as the top-performing model for residential roof recognition and color classification. Its strength is demonstrated by the confusion matrix, particularly in very accurate and error-free red roof detection. YOLOv11 outperformed EfficientDet, Faster R-CNN, and SSD regarding detection rates and background/roof color differentiation. Notwithstanding difficulties with specific roof types, such as black roofs, it is the most dependable model with a mAP@0.5 score of 0.708 and good F1 scores, making it ideal for real-world applications.

The model exhibited remarkable computing efficiency, with preprocessing times of 0.2ms, inference times of 7.6ms, and postprocessing times of 0.7ms. The training process was completed in 0.946 hours utilizing an NVIDIA GeForce RTX 4060, illustrating the model's viability for real-world application.

The research highlights the benefits and limitations of utilizing YOLOv11 for automatic roof detection and counting. The model exhibits strong performance for particular roof types, notably red roofs. Notwithstanding these achievements, significant limits were noted, including comparatively low confidence scores (0.10-0.20) for specific detections and fluctuations in performance attributable to lighting conditions. The diminished sample size in the test location limited the system's effectiveness in the Black-Roof category. The roof shape will affect the accuracy, as the dataset is bespoke and sourced from Indonesia, resulting in heightened precision when utilized in Indonesian areas. Future research opportunities include expanding the training dataset for underrepresented categories, improving the model to increase confidence ratings, and investigating potential interactions with GIS systems for broader urban planning and analysis applications.

ACKNOWLEDGEMENT

MEK contributed programming code, wrote the manuscript, conducted experiments, and reviewed the paper for this research. AZHP contributed to programming code, preparing datasets, writing manuscripts, and reviewing papers. Kamel contributed to preparing the dataset, conducting experiments, writing manuscripts, and reviewing the paper. HP contributed to supervising the steps of conducting experiments and reviewing the paper. Our sample datasets can be accessed at: https://www.kaggle.com/datasets/drhadypranoto/drone-image-in-indonesia.

REFERENCES

[1] Wilson, K., & Davis, R. "Applications of computer vision in urban planning: Current trends and future perspectives", Cities, 132, 103925, 2023.

[2] Zhang, Q., & Zhang, Y. "Deep learning-based building detection using aerial imagery: A comprehensive review", Remote Sensing, 14(3), 662, 2022.

[3] Li, X., et al. "Automated Building Detection Using Deep Learning: A Review", ISPRS Journal of Photogrammetry and Remote Sensing, 185, 251-272, 2023.

[4] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors", arXiv preprint arXiv:2207.02696, 2023.

[5] Martinez, J., & Thompson, E. "Building detection in complex urban environments: A deep learning approach", International Journal of Applied Earth Observation and Geoinformation, 117, 103110, 2023.

[6] Chen, R., & Li, X. "Color-based object detection in aerial imagery: Challenges and solutions", IEEE Transactions on Geoscience and Remote Sensing, 61, 1-15, 2023.

[7] Anderson, P., et al. "Color classification in computer vision: Challenges and solutions for architectural applications", Pattern Recognition Letters, 168, 8-19, 2023.

[8] Liu, Y., et al. "Deep learning for roof type classification: A comparative study", Remote Sensing of Environment, 280, 113233, 2023.

[9] Smith, J., & Brown, A. "Urban feature extraction using deep learning: Recent advances and future directions", Urban Remote Sensing Journal, 45(2), 89-112, 2023.

[10] R. Gelar Guntara, "Deteksi Atap Bangunan Berbasis Citra Udara Menggunakan Google Colab dan Algoritma Deep Learning YOLOv7 ", JMASIF, vol. 2, no. 1, pp. 9–18, May 2023.

[11] Salar Ghaffarian, Saman Ghaffarian. "Automatic Building Detection based on Supervised Classification using High Resolution Google Earth Images", ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-3, September, 2014.

[12] Vichai Viratkapan., Saprangsit Mruetusatorn. "Classification of House Categories Using Convolutional Neural Networks (CNN) ", 509-514, May, 2022.

[13] Jiaqi Jia., Min Fu., Xuefeng Liu., Bing Zheng. "Underwater Object Detection Based on Improved EfficientDet", Remote Sensing, vol. 14, no. 18, pp. 4487, 2022.

[14] Lin, T. Y., et al. "Microsoft COCO: Common Objects in Context", vol. 8693, pp. 740-755, 2014.

[15] Zhao, Z. Q., et al. "Object Detection with Deep Learning: A Review", IEEE Transactions on Neural Networks and Learning Systems, pp. 1-21, January 2019.

[16] Garcia, M., et al. "Interpreting Confusion Matrices in Urban Remote Sensing", Remote Sensing of Environment, 2023.

[17] Park, S., et al. "Confidence Threshold Optimization in Object Detection", IEEE Transactions on Image Processing, 2023.

[18] Rodriguez, A., et al. "Class-wise Performance Analysis in Aerial Image Detection", ISPRS Journal, 2023.

[19] Wu, X., et al. "Aerial Image Building Detection: A Comprehensive Review", Remote Sensing, 2023.

[20] Redmon, J., Divvala, S., Girshick, R., dan Farhadi, A. "You Only Look Once: Unified, Real-Time Object Detection", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788, 2016.

[21] Redmon, J., & Farhadi, A. "YOLOv3: An incremental improvement", arXiv preprint arXiv:1804.02767, 2018.

[22] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. "YOLOv4: Optimal speed and accuracy of object detection", arXiv preprint arXiv:2004.10934, 2020.

[23] Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. "CSPNet: A new backbone that can enhance learning capability of CNN", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020.

[24] Jocher, G., et al. "Ultralytics YOLOv8: A state-of-the-art model for object detection and image segmentation", 2023.

[25] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. "SSD: Single Shot MultiBox Detector", European Conference on Computer Vision (ECCV), 2016.

[26] Tan, M., Pang, R., & Le, Q. V. "EfficientDet: Scalable and Efficient Object Detection", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[27] Ren, S., He, K., Girshick, R., & Sun, J. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", Advances in Neural Information Processing Systems (NeurIPS), 2015.

[28] Aggarwal, C. C. "Neural Networks and Deep Learning: A Textbook", Springer, 2018.

[29] Raschka, S., & Mirjalili, V. "Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2", Packt Publishing, 2019.

[30] Padilla, R., et al. "A Survey on Performance Metrics for Object-Detection Algorithms", International Conference on Systems, Signals and Image Processing, 2020.

[31] DataCamp, "YOLO Object Detection Explained", DataCamp, Dec. 22, 2022.

[32] A. Rohan, M. Rabah, and S.-H. Kim. "Convolutional Neural Network-Based Real-Time Object Detection and Tracking for Parrot AR Drone 2", vol. 7, pp. 69575-69584, 2019.

[33] M. Tan, R. Pang, and Q. V. Le. "EfficientDet: Scalable and Efficient Object Detection", pp. 10781-10790, 2020.

[34] Z. Deng, H. Sun, S. Zhou, and H. Zou. "Multi-scale object detection in remote sensing imagery with convolutional neural networks", vol. 15, no. 5, pp. 749-753, 2018.