Iraqi Journal of Science, 2025, Vol. 66, No. 10, pp: 4529– 4544
DOI: 10.24996/ijs.2025.66.10.38

ISSN: 0067-2904

Constructing a Dataset for House Detection in Baghdad Boundaries Using Satellite Imagery and Deep Learning

Omar Abdul-Latif Abdul-Jabbar [1]*, Mudhar N. Abdullah[2], Ahmed T. Sadiq[1]
[1]*Department of Computer Sciences, University of Technology, Baghdad, Iraq*
[2]*CGIS of Qatar, Doha, State of Qatar*

Received: 13/5/2023 Accepted: 5/10/2024 Published: 30/10/2025

Abstract

Unplanned residential expansion into agricultural areas has been driven by factors like population growth, internal displacement, and expensive housing options. This article presents the Baghdad Houses detection dataset. The dataset includes 1627 manually annotated images of the Dorra area in Baghdad, along with bounding box annotations that define houses in the satellite image. This dataset will be used by deep learning algorithms to understand and analyze human population expansion by detecting the spread of houses in the newly developed area in the boundaries of Baghdad (object detection). After the training and evaluation process, the chosen algorithms were able to detect houses and visualize residential expansion with a reasonable degree of success. This method can be used to monitor and understand the population expansion in Iraq.

Keywords: Satellite images, deep learning, computer vision, Population expansion in Baghdad.

بناء مجموعة بيانات للكشف عن المنازل في مناطق اطراف بغداد باستعمال صور الأقمار الاصطناعية والتعلم العميق

عمر عبد اللطيف عبد الجبار[1]* , مضر نعمان عبدالله[2], احمد طارق صادق [1]
[1] قسم علوم الحاسوب, الجامعة التكنولوجية, بغداد, العراق
[2]مركز نظم المعلومات الجغرافية, الدوحة, قطر

الخلاصة

كان التوسع السكني غير المخطط له في المناطق الزراعية مدفوعًا بعوامل مثل النمو السكاني والنزوح الداخلي وخيارات الإسكان الباهظة الثمن. يعرض هذا المقال مجموعة بيانات الكشف عن منازل بغداد والتي تحتوي على 1627 صورة مشروحة يدويا لمنطقة الدورة، بغداد، شروح المربع المحيطي المستعملة لتحديد المنازل في صورة القمر الاصطناعي. سيتم استعمال مجموعة البيانات هذه بواسطة خوارزميات التعلم العميق لفهم وتحليل التوسع السكاني البشري من خلال الكشف (الكشف عن الكائنات) عن انتشار المنازل في المنطقة المطورة حديثًا في حدود بغداد، وبعد عملية التدريب والتقييم، تمكنت الخوارزميات المختارة من اكتشاف المنازل،

*Email: cs.22.26@grad.uotechnology.edu.iq

تصور التوسع السكني بدرجة معقولة من النجاح. يمكن استعمال هذه الطريقة لرصد وفهم التوسع السكاني في المدن العراقية مع استمرارها في النمو.

1. Introduction

Population: The total number of individuals residing in a region, for example, in a country or on the planet, constantly changing through births, immigrations, and deaths. Just as resource supply and pathogen conditions affect other biological populations, so do human population sizes. The human population growth on this planet in the 20th century has been nothing short of spectacular. At the beginning of the century, the Earth had an estimated 1.6 billion inhabitants; that number grew to 6.1 billion by the end of the century, and the population of humans on Earth grew at a rate never witnessed before. By the end of the century, there were an estimated 6.1 billion people on Earth, up from an estimated 1.6 billion at the start of the century [1].

Most countries continue to lack reliable statistics on individual society situations, such as population, health, economy, and infrastructure. Conventional techniques for gathering and monitoring this kind of information involve labor- and money-intensive on-site surveys. Conversely, high-resolution satellite photos and other remote sensing data are becoming widely accessible, and computer vision combined with deep learning has been effectively applied to raw satellite imagery to overcome community problems at a high granularity [2]. One of the most important remote sensing tasks is identifying building footprints and extracting road networks, as they are critical for nations to understand the effects of urban growth on various ecosystems [3].

Satellite images are finely detailed images of the Earth's surface captured by sophisticated cameras and sensors carried by satellites in orbit. These photos provide a detailed view that is not available from the ground and are essential sources of information for many applications.

Successful attempts to study and analyze societal conditions by computer solutions were made using applications of Geographic Information System (GIS) and remote sensing. GIS can be defined as a computer system that can collect, store, analyze, process, and visualize geographic information such as satellite imagery, maps, and coordinates according to their position on Earth [4].

Satellite imagery offers environmental data on a spatial and temporal scale in many applications that are not achievable with conventional techniques. Satellites have given us a fresh perspective that allows us to perceive our planet as a global system by enabling us to view huge areas of it repeatedly [5].

The properties of optical satellite photos differ significantly from typical images taken on the Earth's surface [6]. The optical sensors, which are usually made up of millions of pixels, can often cover a huge area in one image since they are located on board satellites in the Earth's orbit, which ranges between 160 and 37,000 km in height.

The branch of computer vision known as artificial intelligence (AI) gives machines the ability to perceive, interpret, and comprehend visual data. Machines can accurately detect and classify objects using deep learning models, which are combined with digital photos and videos captured by cameras. Many tasks, including object detection, face recognition, action and activity recognition, and human position estimation, are based on computer vision [7].

Many computer vision issues, including object detection [8, 9], motion tracking [10, 11], action recognition [12, 13], human posture estimation [14, 15], and semantic segmentation, have

advanced significantly as a result of deep learning. In essence, deep learning allows computational models composed of multiple processing layers to learn data representations with various levels of abstraction. It has significantly improved results in speech recognition, natural language processing, and other applications [16].

Deep learning algorithms have shown promising success in satellite image analysis, understanding, and then investing in new fields of applications [17]. A single satellite image can be considered a huge data source for deep learning algorithms because it's formed as a dimension matrix with millions of pixels ready to be digested by deep learning algorithms. Today, satellite imagery is a vital resource for learning about our globe. However, it frequently takes assistance to extract useful information from these enormous databases. Deep learning is thought to be one of the most promising techniques for object detection using satellite images. Using deep learning with satellite images is considered a suitable solution for many complicated issues such as weather monitoring, disaster management, change detection, and agriculture—infrastructure planning. Combining deep learning algorithms with the massive amount of data stored in satellite images represents a solution for analyzing, understanding, and demonstrating residential expansion, as well as assisting decision-makers in making accurate decisions in both the planning and executing phases.

Deep learning algorithms are distinguished for recognizing patterns in complex data because they are modeled after the structure and operations of the human brain. These algorithms can automatically recognize and categorize things of interest, such as buildings, roads, and cars, when they are applied to satellite images.

The cornerstone for using deep learning algorithms is the availability of information, known as the dataset. Any proposed satellite image dataset for object detection using deep learning comprises a collection of satellite images that encompass the specified area. Because deep learning is classified under supervised learning, satellite images must have an early step of processing called labeling; this is the way to direct the model to the specific region of interest in the satellite image.
The dataset for this objective must consist of several satellite images covering the study area (Dorra, Baghdad) and annotations that determine the required class, which in this case are the houses. Unfortunately, after a search, no public dataset that can be used for house detection in Baghdad with deep learning was found. This paper outlines the process of creating a dataset that academic researchers and developers can use to train, test, and develop deep learning algorithms for house detection in Baghdad and Iraq, or for similar applications.

2. Related Works

Object detection algorithms in normal images play a crucial role in a wide range of applications, from security surveillance to autonomous driving and even healthcare, but they require a lot of work to maintain the same accuracy when used to detect objects in satellite images because of the wide difference between the normal images and the satellite images as a result of the buildings' different sizes, orientations, and backgrounds. Traditional deep learning algorithms used for object detection are initially designed to work with normal images like camera images; therefore, despite their outstanding performance, these models perform poorly on satellite photos with inputs of roughly 16000 × 16000 pixels. Adam Van Etten added a pipeline named You Only Look Twice [28] (YOLT) to the YOLOv2 framework in May 2018 so that it could analyze huge satellite photos at a pace of 0.2 km^3/s. Later that year, Van Etten incorporated additional models into a recently developed pipeline known as Satellite Imagery Multiscale Rapid Detection with Windowed Networks (SIMRDWN) [29], resulting in an improvement in the efficiency of the previous model. These models included YOLOv3, SSD, Faster R-CNN,

and R-FCN. In 2021, YOLTv4, a revised version of SIMRDWN, became available. It combined the pre- and post-processing scripts of SIMRDWN with the upgraded model YOLOv4 for increased accuracy and frames per second (FPS). Building datasets for deep learning-based object detection of satellite images is another important research focus, with expansive areas of use across several sectors such as urban development, emergency management, and environmental surveillance, among others. Several datasets have been created in this area of work. For example, the xView dataset, proposed by Lam et al. (2018), is considered one of the largest in the world for satellite imagery. It has more than 1 million annotated objects from 60 classes, thus helping in training models for efficient and accurate object detection, especially when developed in urban and rural areas [18]. In the same vein, Xia et al. (2018) introduced the DOTA: Dataset for Object Detection in Aerial Imagery, which includes a set of high-resolution aerial images in fifteen categories annotated with the object set that tries to overcome some of the issues concerning object size, orientation, and density [19]. The SpaceNet initiative described by Van Etten et al. (2018) provided an open dataset with labeled satellite imagery regarding building footprints and road networks, which stimulates research on the subject of geospatial machine learning. These datasets are perfect illustrations of the kind of cooperation that has characterized the development of the remote sensing and machine learning databases [20].

In 2018, Ovidiu Csillik and colleagues successfully differentiated citrus trees from other trees using a basic convolutional neural network, with their project progressing smoothly. The team attained impressive results, with a general accuracy of 96.24, precision (positive predictive value) of 94.59, and recall of 97.94. This marked the inaugural application of CNN technology to drone imagery specifically targeting citrus trees [21].

Further, the AID (Aerial Image Dataset) is the dataset introduced by Xia et al. (2017), organized in over 10,000 samples on 30 classes to support scene classification and recognition missions [30]. The UC Merced Land Use Dataset, developed by Yang and Newsam (2010), has 21 classes with 100 images each and is commonly used for experimental studies that focus on land use classification [22].

Maggiori et al. (2017) constructed the INRIA Aerial Image Labeling set, a series of large-scale aerial images with annotations, to aid in semantic annotation and building extraction algorithms in urban settings. Such datasets illustrate the recent cooperative work within the communities of remote sensing and machine learning to produce large-scale, high-quality datasets that would allow the promotion of new techniques in object detection and other related tasks in the fields of satellite and aerial imagery [23].

3. Methodology:

The first step in building the required dataset is the data collection, which includes satellite images that should cover the study area (Dorra). The River of Tigress surrounds Dorra (E: 44.44615234897068, N: 33.23073811485445), which features a wide expanse of agricultural areas recently transformed into residential areas, as depicted in figure [1]. The selected image from WorldView-2 [24] was used, and it covers 50 sq. KM.

Figure 1: Study area in Dorra, Baghdad

The detailed information of the selected satellite image is displayed in table [1]:

Table 1: Satellite image information

Image Source	*World view 2*
Extent: Top Left	44.4092520°E 33.2722887°N
Extent: Bottom right	44.4747182°E 33.1989280°N
Resolution	0.3 meters per pixel
spectral bands	Red, Green, Blue
Covered Area	50 Sq. KM
Projected Coordinates system	WGS 1984 UTM Zone 38N
Geographic coordinates system	WGS 1984

Identifying houses in satellite images presents a complex task. The selected satellite images are (14740 x 16384) pixels, so there is a need to divide the satellite images into smaller parts. The satellite image was divided into equal (640 x 640) parts to be suitable for deep learning algorithms like the YOLO family. The result was a (591) sub-image. Figure 2 presents four samples of the sub-images.

Figure 2: 640 x 640 image samples

The next step is image annotation. Annotation refers to the process of meticulously annotating raw data, such as images, text, or video, with specific labels. These labels (bounding boxes) provide context and meaning to the data, enabling deep-learning models to grasp the underlying patterns and relationships [25]. Bounding boxes were drawn around the targeted class (houses) in the 591 images, trying to cover as many houses as possible in every image. The output from the labeling process is called the annotation files, which store the central coordinates of the bounding box and the height and width of that box so it can be related to its target (house). The platform used for the annotation process is Robflow [26].
The final result of the labeling process is presented in Figure 3.

Figure 3: labeled images

The results of the annotation process were three folders (train, validate, test), respectively. In each folder, there are two nested folders (images, labels); the images folder stores the images, and the labels folder stores all the annotation files for images in the same directory folder. The structure of the annotation file is:

<Object class> <x_center> <y_center> <x_width> <y_height>

The annotation XML file was considered irrelative and removed, and the annotation file example figure was also removed.
This describes the format for data about a detected house within an image:

- <Object class>: This represents the category of the detected object. In this case, since it's the only class, it will always be a number representing "house".
- <x_center>: This value indicates the horizontal center position of a rectangle drawn around the detected house, relative to the entire image width. It's likely a value between 0 (left edge) and 1 (right edge).
- <y_center>: Similar to <x_center>, this represents the vertical center position of the bounding box around the house, relative to the image height. It will likely be a value between 0 (top edge) and 1 (bottom edge).
- <x_width>: This value signifies the width of the bounding box surrounding the house, expressed as a proportion of the entire image width. So, a value of 0.2 would mean the box is 20% as wide as the image.
- <y_height>: Similar to <x_width>, this represents the height of the bounding box relative to the image height.

In simpler terms, this format uses a bounding box to provide information about the location and size of a detected house within an image. The center coordinates (<x_center> and <y_center>) and relative dimensions (<x_width> and <y_height>) help pinpoint the house and its size compared to the whole image.

Pre-processing & Augmentations

Deep learning necessitates a substantial volume of data to be utilized for training, validation, and testing to enhance the accuracy of predictions. To compensate for the lack of images, a set of preprocessing and augmentation tools was used to increase the size and diversity of the data set. These operations are:

1) Pre-processing :
a) Auto-Orient (90°, 180°, 270° of rotation).
b) Grayscale.
2) Augmentations :
a) Flip: Horizontal.
b) Hue: Between -25° and +25°.
c) Saturation: Between -25% and +25%.
d) Bounding Box: 90° Rotate: Clockwise, Counter-Clockwise, Upside Down.

The pre-processing and augmentation processes resulted in a dataset containing 1627 labeled images organized in three sets according to the data split ratio of 70:30, as shown in table 2.

Table 2: Dataset image organization

Training set	Validation set	Testing set
1554 images (%70)	48 images(%20)	25 images (%10)

The data split ratio was added to the text and table.

The final dataset is now published at Roboflow, and it's available for researchers and developers to download, use, and train their models for different deep-learning applications. https://universe.roboflow.com/baghdad-houses-detection/baghdad-houses-detection/dataset/10

Data Set training:

The model training step is now underway. YOLO (v5 & v8) deep learning algorithms were selected for this step. The core of the YOLOv5 architecture presented in figure 4 is based on a modified version of CSPDarknet53, which incorporates a stem and a large-window stride convolution layer to optimize memory usage and computational efficiency. Convolutional layers are then applied to the input image to extract important features. The Spatial Pyramid Pooling Fast (SPPF) layer processes data at multiple scales, while subsequent convolutional and upsampling layers enhance the resolution of feature maps. The SPPF layer pools features from various scales into a fixed-size feature map, improving the network's computational efficiency. Each convolutional layer is followed by batch normalization (BN) and the SiLU activation function. The architecture's head resembles that of YOLOv3, while the neck incorporates SPPF and an adapted CSP-PAN. YOLOv5 uses several augmentations from the augmentation package, including Mosaic, copy-paste, random affine transformations, MixUp, HSV augmentation, random horizontal flip, and others. Additionally, it improves grid sensitivity, making it more robust against runaway gradients [27].

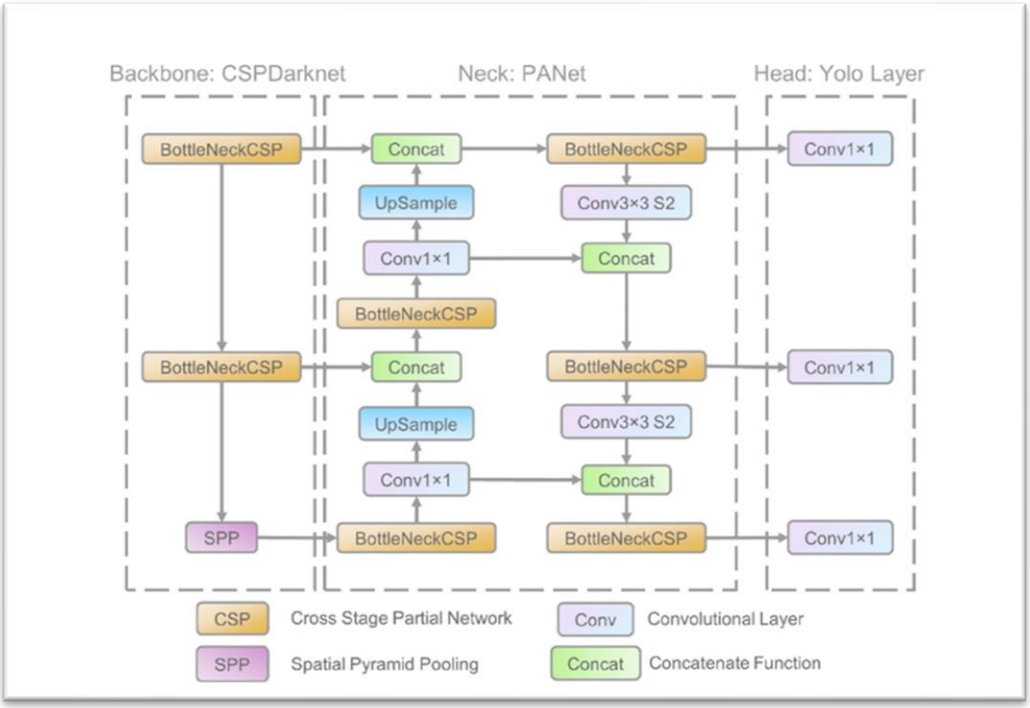

Figure 4: YOLOV 5 architecture

YOLO v5's multiple network architectures are more user-friendly, include a lightweight model size, and offer accuracy levels comparable to the YOLO v4 test. Compared to the Darknet framework used in YOLO v4, the Pytourch framework is easier to use with the data set, making it easier to use in production.

YOLOv8, also known as You Only Look Once version 8, is a single-stage object detection model that has gained recognition for its exceptional speed and accuracy when deployed in real-time applications. It builds upon previous YOLO iterations, achieving improvements in

both aspects while introducing novel architectural components. Ultralytics has announced the development of YOLOv8, aiming to provide more features and operate faster on both CPU and GPU devices than its predecessors. The new API in YOLOv8 simplifies the process of training and inference. Although the framework continues to support previous YOLO versions, the developers are in the process of preparing a scholarly publication that will provide a detailed explanation of the model's architecture and performance [27]. Figure 5 depicts the YOLOv8's work and architecture.

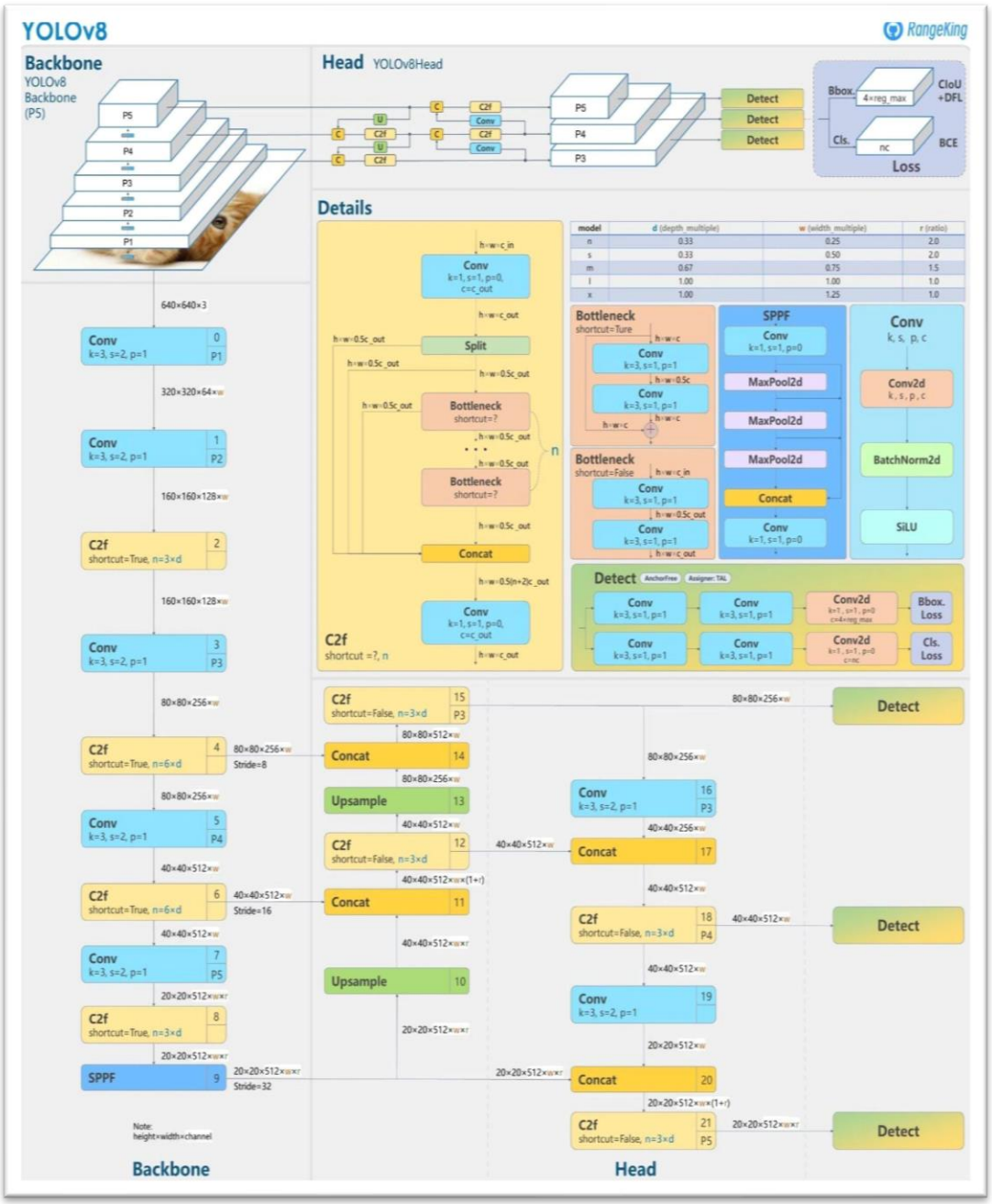

Figure 5: YOLO v8 work and architecture

On the dataset, Python and Google Collab were used to train YOLO V5 and V8. Both algorithms were trained for 25 epochs, and the trained models are currently deployed with the dataset and can be used and tested at the Roboflow site.

4. Results and Discussion

As previously mentioned, given a single satellite image and the unplanned residential expansion in Baghdad without a proven unified method or map for designing and constructing homes and buildings, contracts are obligated to use a standard plane or area based on the results of the training process for YOLOv8 and YOLOv5, as illustrated in figures (6) and (7), respectively. The model couldn't make higher accuracy after many different training attempts. This modest dataset can be a first step. At the moment, work is underway to collect more satellite images and perform data labeling, preprocessing, and augmentation to build a dataset suitable for deep learning algorithms specified for houses and building detection in Baghdad and Iraq. Before discussing the results of both (V5 and V8), here is a short definition of what the different metrics mean:

- **Box Loss** (train/box loss, Val/box loss) evaluates the model's alignment between the predicted bounding boxes and the actual bounding boxes in the training dataset. A decrease in the values of this metric signifies a higher level of accuracy and effectiveness in the model's performance.
- **Cls Loss** (train/cls loss, Val/cls loss): This metric measures how well the model can classify the objects within the bounding boxes. Lower values indicate better performance.
- **Dfl Loss** (train/dfl loss, Val/dfl loss): This metric is likely measuring the deformation of the bounding boxes. A lower loss indicates that the predicted bounding boxes have a similar shape to the ground truth boxes.
- **mAP** (metrics/mAP50 (B), Val/mAP50 (B)): The most commonly used metric for object detection model evaluation is the **mean average precision** at the intersection over the union of 0.5. In other words, it simply indicates how well a model ranks objects appropriately while considering bounding boxes that overlap with the ground truth box simultaneously. The higher the mAP0.5 value, the better the performance.
- **mAP50-95** (metrics/mAP50-95(B), Val/mAP50-95(B)): This metric is similar to mAP0.5 but uses a stricter threshold of 0.95 for Intersection over Union.

4.1 YOLO V8 training results:

Figure 6 represents a visualization of the training process for a YOLOv8 object detection model on the dataset. Finally, it is evident that the model, using the available satellite image and the construction pattern of houses in Baghdad, performed satisfactorily. However, it is certain that it requires additional data, pre-processing, and training operations.

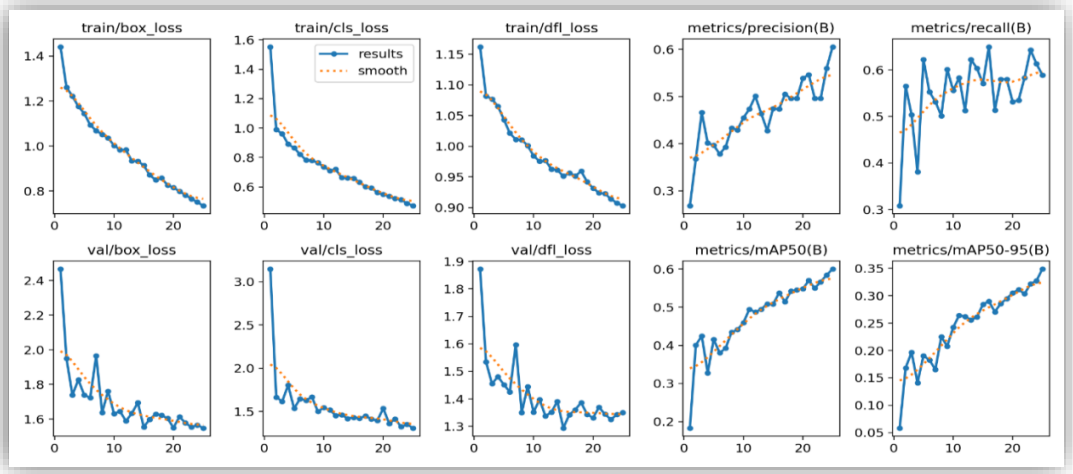

Figure 6: training Graphs for YOLOv

Figure 7 is a confusion matrix visualizing the performance evaluation of a YOLOv8 model on how well it can distinguish houses from backgrounds. Here is a short explanation of it:

- Rows: Represent the actual labels of the data (ground truth). In this case, there are two rows: house and background.
- Columns: Illustrate the forecasted classifications produced by the model. Once more, the data is presented in two separate columns: one for 'house' and the other for 'background'.
- Values in the table: the value at the top left corner (225) indicates that there were 225 instances where the model correctly predicted a house.

However, here are some general observations we can make based on Figure 7:

- The model seems to be good at predicting houses (high value in the top left corner). There were 225 correct house predictions.
- There seems to be a low number of false positive house detections (background predicted as a house). The value in the top right corner is 0.

Overall, the confusion matrix suggests that the YOLOv8 model performs well at detecting houses with a low number of false positive detections. Table 3 also displays the information from the confusion matrix in Figure 7.

Table 3: Confusion matrix

Confusion matrix				
predicted	house	255	208	225
				200
				175
				150
				125
	background	106		100
				75
				50
				25
		background	house	0
		true		

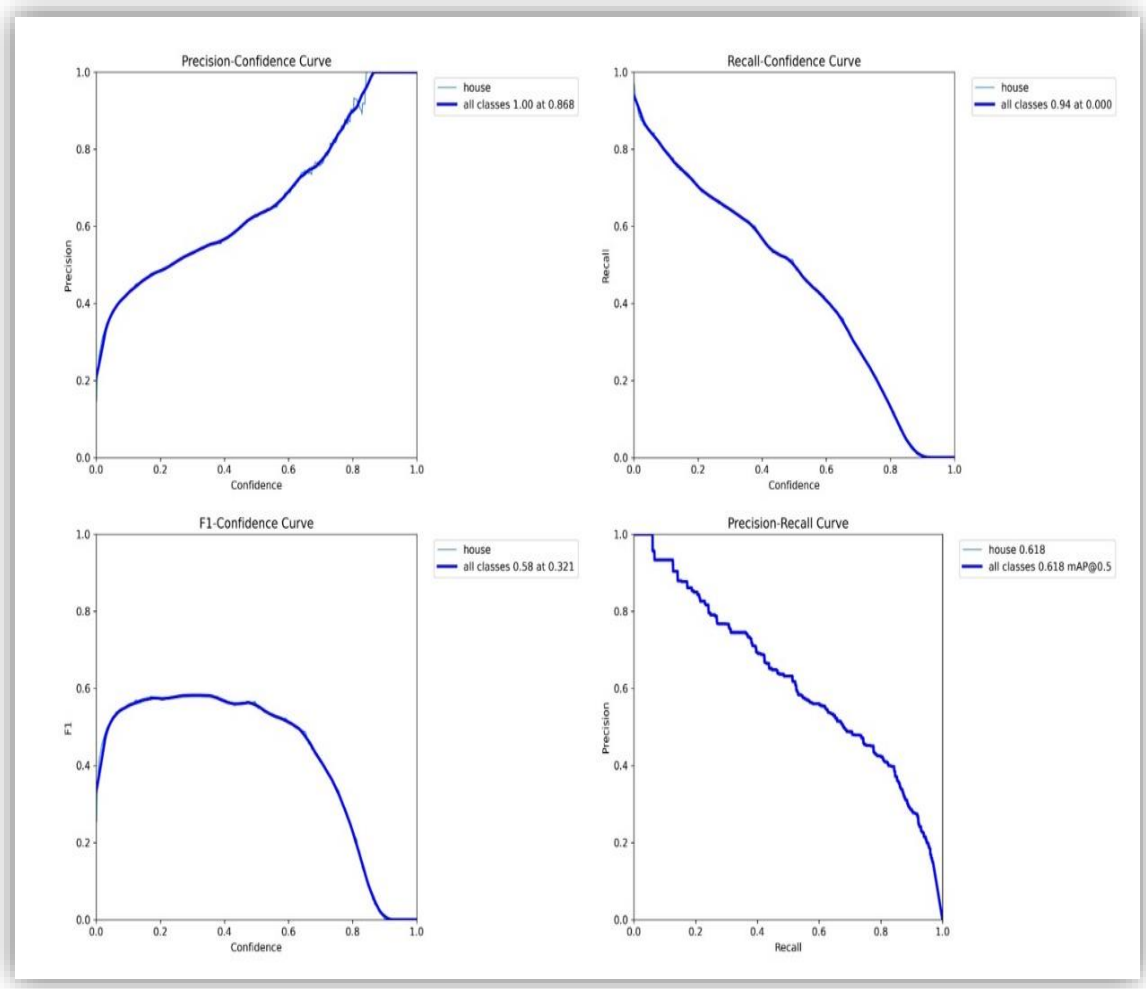

Figure 8: training curves for YOLOv8

Figure 8 shows the YOLOv8 model's performance on object detection (house). In the context of house detection, Figure 8 would help to understand how well the model is performing at detecting houses with different confidence levels.

- **Precision-Confidence Curve and Recall-Confidence Curve:** These curves show how good the model's precision and recall are at different confidence levels. Here, the precision is the detection of houses and the recall of real houses detected by the model. The higher one sets the confidence threshold level, the more assured the model is with its detection; however, it may reject some actual houses. Such curves show the model's variation trend, with a change in precision and recall confidence threshold.
- **F1-Confidence Curve**: The F1 score is a harmonic mean of precision and recall, and it provides a single measure of the model's performance. This curve plots the F1 score of the model at different confidence thresholds.
- **Precision-Recall Curve:** The curve shows the model's precision on the x-axis and the recall on the y-axis. An ideal graph would show the curve at the upper left corner, representing that the model has a high value of both precision and recall.
- **mAP**: mean Average Precision, is a measure of overall detection performance that takes into account precision and recall across all confidence thresholds. The value in the image, 0.5, suggests an average precision of 0.5.

Figure 9 shows the final step, which is house predicting with a collection of predicted images marking predicted houses with red boxes.

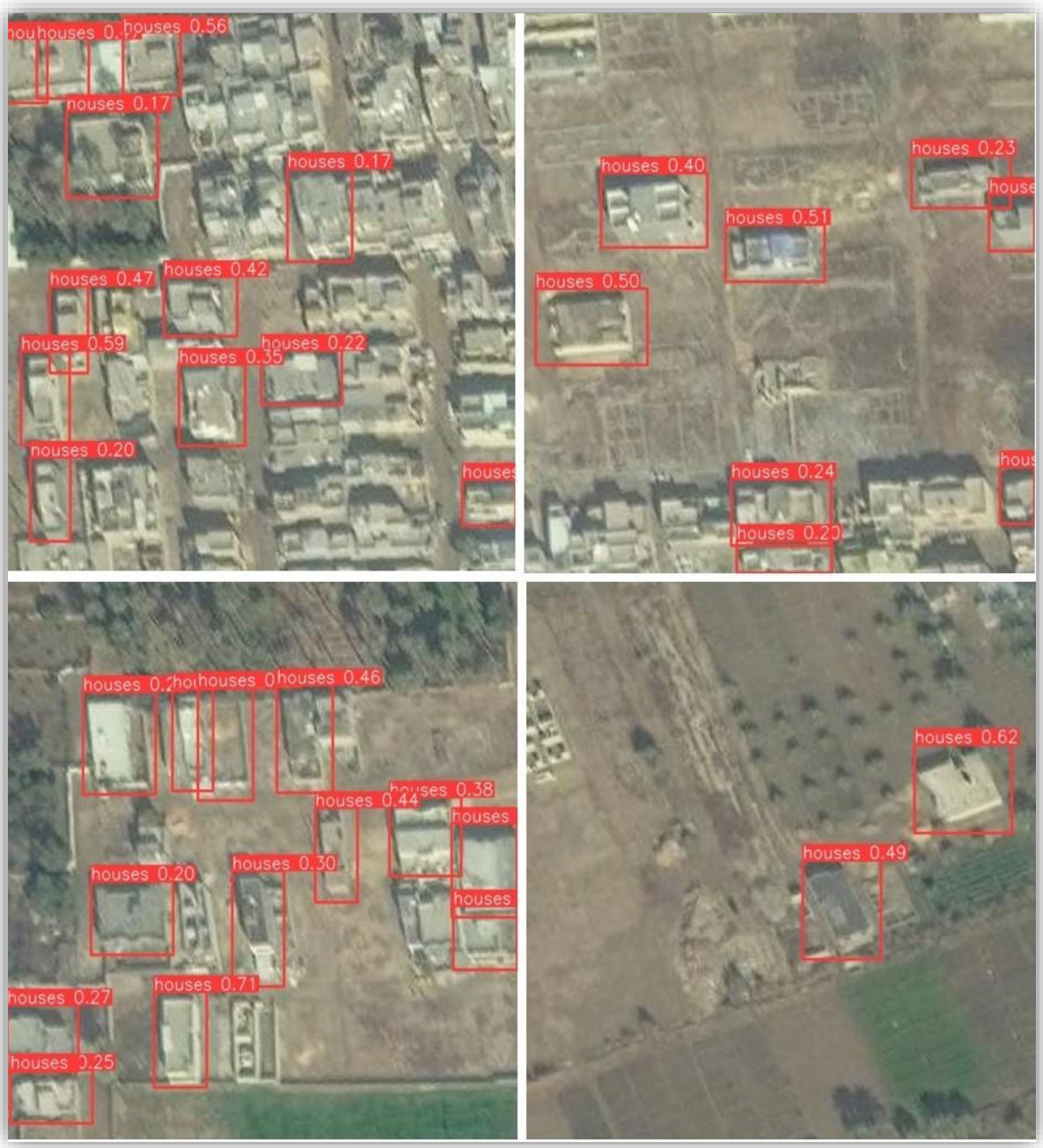

Figure 9: Samples of predicted images from YOLOv8

4.2 Dataset metrics for YOLO V5:

We will only list the training results of YOLO v5 without describing the detailed information because it's almost similar to YOLO v8 training results, which we discussed earlier. Figure 10 represents the training results of YOLO v5, and Figure 11 displays samples of predicted images.

Table 4: YOLO 5 & 8 training results

version	mAP	Precision	Recall
YOLO V5	45.0%	39.1%	65.6%
YOLO V8	60.0%	60.6%	58.9%

Table 4 and Figure 12 present a comparison of YOLOv5 and YOLOv8's object (house) detection performance in the dataset.

- **mAP (mean average precision):** This is the primary metric used to compare the overall performance of object detection models. It considers both precision and recall across all object classes being detected (houses in this case). A higher mAP value indicates better overall performance. In this case, YOLOv8 outperforms YOLOv5 with a mAP of 60.0% compared to 45.0%.
- **Precision:** The metric reflects the part of correct detections among all made detections. In other words, this value measures how many things that are said to be houses are houses. The higher the precision value, the fewer false positives the model produces. Here, YOLOv8 again performs slightly better with 60.6% precision compared to YOLOv5's 39.1%.
- **Recall:** This metric tells the proportion of actual houses in the image that the model correctly detects. In simpler terms, it represents how many actual houses the model didn't miss. A higher recall value indicates the model makes fewer false negative mistakes. Here, YOLOv5 performs better with a recall of 65.6% compared to YOLOv8's 58.9%.

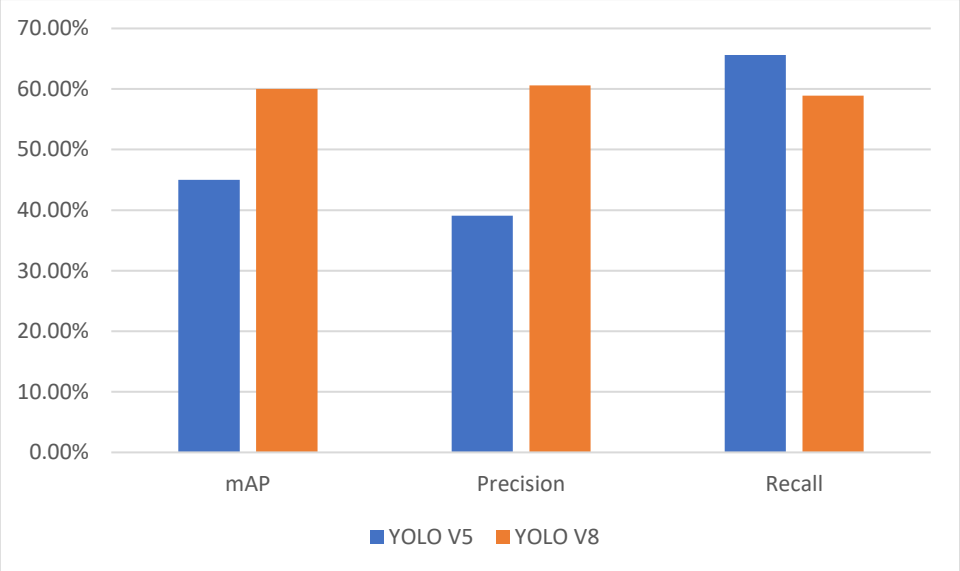

Figure 12: YOLO 5 & 8 training results performance comparison

5. Conclusion:

As previously mentioned, YOLOv5 and YOLOv8 algorithms were utilized to evaluate the constructed dataset. Both algorithms produced moderate results in detecting houses from satellite images of Baghdad. The findings indicate that the dataset (Baghdad-houses-detection) has potential for improvement and could be refined to support more complex house detection systems.

At the current time, there is a continuous process to add and annotate more images to the dataset. YOLOv8 demonstrated a better balance between precision and recall, resulting in a higher mAP score, which suggests it performs more accurately overall in detecting houses.

However, YOLOv5 exhibited a higher recall, meaning it identified more actual houses and missed fewer compared to YOLOv8.

References

[1] R. M. Sibly and J. Hone, "Population growth rate and its determinants: an overview," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 357, no. 1425, pp. 1153–1170, Sep. 2002, doi: https://doi.org/10.1098/rstb.2002.1117.

[2] S. Piaggesi *et al.*, "Predicting City Poverty Using Satellite Imagery," *International Journal of Computer Science and Information Technology Research*, vol. 11, no. 2348–1196, pp. 90–96, Jan. 2019.

[3] C. Ayala, R. Sesma, C. Aranda, and M. Galar, "A Deep Learning Approach to an Enhanced Building Footprint and Road Detection in High-Resolution Satellite Imagery," *Remote Sensing*, vol. 13, no. 16, p. 3135, Aug. 2021, doi: https://doi.org/10.3390/rs13163135.

[4] P. A. Burrough, R. Mcdonnell, and C. D. Lloyd, *Principles of geographical information systems*. Oxford: Oxford University Press, 2015.

[5] S. Nath, *An introduction to remote sensing*. Birmingham: Koros, 2014.

[6] E. D. Conway and M. Space, *An Introduction to Satellite Image Interpretation*. JHU Press, 1997.

[7] F. Liarokapis, A. Voulodimos, N. Doulamis, and A. Doulamis, *Visual computing for cultural heritage*, 1st ed. Cham, Switzerland: Springer, 2020.

[8] W. Ouyang, "DeepID-Net: Object Detection with Deformable Part Based Convolutional Neural Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1320–1334, 2017.

[9] A. Diba, V. Sharma, A. M. Pazandeh, H. Pirsiavash, and L. Van Gool, "Weakly Supervised Cascaded Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 5131–5139, doi: https://doi.org/10.1109/cvpr.2017.545.

[10] N. Doulamis and A. Voulodimos, "FAST-MDL: Fast Adaptive Supervised Training of multi-layered deep learning models for consistent object tracking and classification," *IEEE Xplore*, Oct. 01, 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7738244.

[11] N. Doulamis, "Adaptable deep learning structures for object labeling/tracking under dynamic visual environments," *Multimedia Tools and Applications*, vol. 77, no. 8, pp. 9651–9689, Nov. 2017, doi: https://doi.org/10.1007/s11042-017-5349-7.

[12] L. Lin, K. Wang, W. Zuo, M. Wang, J. Luo, and L. Zhang, "A Deep Structured Model with Radius–Margin Bound for 3D Human Activity Recognition," *International Journal of Computer Vision*, vol. 118, no. 2, pp. 256–273, Dec. 2015, doi: https://doi.org/10.1007/s11263-015-0876-z.

[13] S. Cao and R. Nevatia, "Exploring deep learning based solutions in fine-grained activity recognition in the wild," *IEEE Xplore*, Dec. 01, 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7899664.

[14] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660, doi: https://doi.org/10.1109/CVPR.2014.214.

[15] Y. Ke *et al.*, "A Rapid Object Detection Method for Satellite Image with Large Size," in *Multimedia Information Networking and Security, 2009. MINES'09. International Conference on*, 2009, vol. 1, pp. 637–641.

[16] C. Robinson, F. Hohman, and B. Dilkina, "A Deep Learning Approach for Population Estimation from Satellite Imagery," in *GeoHumanities '17: Proceedings of the 1st ACM SIGSPATIAL Workshop on Geospatial Humanities*, no. 9781450354967, pp. 47–54, Oct. 2017, doi: https://doi.org/10.1145/3149858.3149863.

[17] M. Pritt and G. Chern, "Satellite Image Classification with Deep Learning," in *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, no. 2332–5615, pp. 1–7, Oct. 2017, doi: https://doi.org/10.1109/aipr.2017.8457969.

[18] D. Lam *et al.*, "xView: Objects in Context in Overhead Imagery," *arXiv*, Jan. 2018, doi: https://doi.org/10.48550/arxiv.1802.07856.

[19] T. Blaschke, "Object-based image analysis for remote sensing," *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 1, pp. 2–16, 2010.

[20] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, "SpaceNet: A Remote Sensing Dataset and Challenge Series," *arXiv*, Jul. 14, 2019. [Online]. Available: https://arxiv.org/abs/1807.01232.

[21] G.-S. Xia *et al.*, "AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017, doi: https://doi.org/10.1109/tgrs.2017.2685945.

[22] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10*, 2010, doi: https://doi.org/10.1145/1869790.1869829.

[23] Z. H. Jarrallah and A. Khodher, "Satellite Image Classification using Spectral Signature and Deep Learning," *Iraqi Journal of Science*, vol. 64, no. 6, pp. 4053–4063, Jun. 2023, doi: https://doi.org/10.24996/ijs.2023.64.6.42.

[24] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2017, doi: https://doi.org/10.1109/igarss.2017.8127684.

[25] P. K. Kakani and S. Vyas, "Automated Catalog Generation using Deep Learning," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 05, no. 08, Aug. 2023, doi: https://doi.org/10.56726/irjmets44010.

[26] B. Petrovska *et al.*, "Aerial Scene Classification through Fine-Tuning with Adaptive Learning Rates and Label Smoothing," *Applied Sciences*, vol. 10, no. 17, p. 5792, Aug. 2020, doi: https://doi.org/10.3390/app10175792.

[27] Rusul Hussein Hasan, Rasha Majid Hassoo, and Inaam Salman Aboud, "Yolo Versions Architecture: Review," *International Journal of Advances in Scientific Research and Engineering*, vol. 09, no. 11, pp. 73–92, Jan. 2023, doi: https://doi.org/10.31695/ijasre.2023.9.11.7.