DOI 10.1590/s1982-21702024000100011

ORIGINAL ARTICLE

Automatic detection of urban infrastructure elements from terrestrial images using deep learning

Jaime Carlos Macuácua[1,2] - ORCID: 0000-0002-4822-6098

Jorge António Silva Centeno[1] - ORCID: 0000-0002-2669-7147

Fernando Alves Barros Firmino[1] - ORCID: 0000-0002-2155-5449

Jorgiana Kamila Teixeira Do Crato[1] - ORCID: 0009-0005-3648-4160

Kauê de Moraes Vestena[1] - ORCID: 0000-0003-1225-2371

Caisse Amisse[3] - ORCID: 0000-0001-9458-5510

[1]Departament of Geomatics, Federal University of Paraná, Curitiba, Brazil.
E-mail: macua.gis@gmail.com, centeno@ufpr.br, fernando.firmino@ifpa.edu.br,
jorgianakamila@gmail.com, kauemv2@gmail.com, caamisse@gmail.com

[2]Eduardo Mondlane University. Av. Julius Nyerere, 3453, Maputo - Mozambique.
E-mail: macua.gis@gmail.com

[3]Rovuma University, Av. Josina Machel nº 256, Nampula - Mozambique.
E-mail: caamisse@gmail.com

Received in 03rd April 2023.
Accepted in 10th April 2024.

Abstract:

Urban infrastructure element detection is important for the domain of public management in large urban centres. The diversity of objects in the urban environment makes object detection and classification a challenging task, requiring fast and accurate methods. Advances in deep learning methods have driven improvement in detection techniques (processing, speed, accuracy) that do not rely on manually crafted models, but, instead, use learning approaches with corresponding large training sets to detect and classify objects in images. We applied an object detection model to identify and classify four urban infrastructure elements in the Mappilary dataset. We use YOLOv5, one of the top-performing object detection models, a recent release of the YOLO family, pre-trained on the COCO dataset but fine-tuned on Mappilary dataset. Experimental results from the dataset show that YOLOv5 can make qualitative predictions, for example, the power grid pole class presented the mean Average Precision (mAP) of 78% and the crosswalk class showed mAP around 79%. A lower degree of certainty was verified in the detection of public lighting (mAP=64%) and accessibility (mAP=61%) classes due to the low resolution of certain objects. However, the proposed method showed the capability of automatically detection and location of urban infrastructure elements in real-time, which could contribute to improve decision-making.

Keywords: Deep learning; Urban infrastructure detection; Terrestrial images; YOLO algorithm.

How to cite this article: MACUÁCUA JC, CENTENO JAS, FIRMINO FAB, CRATO JKT, VESTENA KM, AMISSE C. Automatic detection of urban infrastructure elements from terrestrial images using deep learning. *Bulletin of Geodetic Sciences*. 30: e2024011, 2024.

1. Introduction

The citizens' quality of life in urban environments depends, in part, on urban mobility facilities. Traffic control and guidance measures are proposed to organise the movement of people and vehicles, enabling citizens to move, for example, from their homes to schools, workplaces, or supermarkets. However, the urban environment is constantly developing due to socioeconomic and demographic growth, forcing managers of public spaces to rethink and evaluate the current scenario to propose rational measures. Every planning and management action must be based on quality spatial data, including the road network and the different existing elements of infrastructure. However, in many cases, once projects are carried out in urban spaces, there is no permanent control over the functioning or even the existence of urban infrastructure elements. In general, the competent public agencies find it difficult to map the elements of urban infrastructure and, consequently, to manage these spaces. If there is control, it is usually conducted with the help of the population, which informs the competent body of the need to take the necessary measures and, generally, replacement takes a long time due to bureaucracy. These structures include crosswalks, power grid poles, with or without public lighting, accessibility ramps, bike paths, sidewalks, traffic signals and signs, among others.

With the development of new sensors, technologies have emerged, based on mobile platforms, to support mobile mapping using global positioning GNSS/GPS sensors, inertial sensors, LiDAR and cameras, which allow the collection of a large amount and quality of data, point clouds and images. It is also possible to note that mobile land mapping systems offer the opportunity to prepare topographic maps and create a database of images of georeferenced urban infrastructure elements. Therefore, the current problem lies in extracting useful information from this data. The present work aims to develop a methodology for the automatic detection and identification of urban infrastructure elements in images obtained from terrestrial platforms, using deep network methods. As an example, this article uses public images of power grid poles, public lighting, crosswalk signalling on public roads and accessibility for wheelchair users, obtained from the Mapillary API collaborative platform associated with OpenStreetMap.

Object detection in images is a vast field of research undergoing constant development, with applications in the real world and even surpassing human capacity in solving some visual tasks with the use of artificial intelligence, deep learning and image processing (Bai et al., 2020; He et al., 2021; Soori et al., 2023). The application of deep neural networks has proven to be efficient in several areas, including the medical field (Sarker et al., 2021; Dildar et al., 2021; Muchuchuti and Viriri, 2023), robotics (Soori et al., 2023), autonomous car technologies (Ning et al., 2021; Elallid et al., 2022), people monitoring (Amisse et al., 2021), and agriculture (Linaza et al., 2021), among others. It is considered by Oguine et al., (2022) as one of the Deep Learning research areas that have fostered the recurrent improvement of object detection models in several interdisciplinary studies.

In this context, when using the images from the Mapillary platform in the context of computer vision, it is possible to identify the presence of structures consistent with urban infrastructure, to provide this information and improve urban management conditions. This contribution can be made through the development of tools and algorithms such as the automatic extraction of geographic features from landscape images to enrich and improve data quality. To overcome specificity issues, which include the need for high quality and precision in detection, the pursuit of greater efficiency in the accurate localisation and recognition of objects, as well as speed-related limitations in traditional detection methods, several research studies have been conducted. For instance, Liu et al. (2020) address the challenges in detection accuracy by comparing traditional models with deep learning methods. Other studies have also employed traditional methods, such as Sangeetha and Deepa (2017), who proposed using the Histogram of Oriented Gradients (HOG) for object detection; Guo et al. (2018), who explored the Scale-Invariant Feature Transform (SIFT) algorithm for image detection and matching; and Umar et al. (2017), who utilised Speeded-Up Robust Features (SURF) and Support Vector Machines (SVM) for object detection in aerial images. Nevertheless, to overcome these limitations and achieve improved results in terms of quality, precision, and efficiency, we propose the adoption of deep learning methods in this study. These methods stand out due to providing better performance and speed, as described in recent research, such as the works of Jiang and Hao (2018) and Cheng (2020), which discuss the efficacy of model families such as CNN and YOLO. These studies compare the structure, computation speed, and efficiency in object identification, highlighting the potential benefits of deep learning methods over traditional approaches.

The utilisation of the YOLOv5 model for detecting urban infrastructure elements in images offers several advantages that make it a solid choice compared to more recent versions. Firstly, it demonstrates greater inference speed and accuracy, leading to efficient and reliable performance. The model is optimised to run effectively on Graphics Processing Units (GPUs) and utilises memory more efficiently, ensuring a smoother execution even in resource-constrained environments. One of the main strengths of YOLOv5 is its advanced architecture, incorporating CSPDarknet53 and PANet, significantly enhancing the precision and accuracy of object localisation and recognition. This sophisticated approach contributes to the overall quality of detections, resulting in more reliable and precise outcomes. Additionally, YOLOv5's modularity, implemented in PyTorch, adds to its appeal. This modularity enables easy customisation for specific use cases and ensures efficient memory utilisation. Such flexibility is particularly valuable for devices with limited memory capacity, making YOLOv5 well-suited for a wide range of practical applications. Notably, YOLOv5 is a general-purpose model, which means it can be adapted for various purposes. Its ability to implicitly encode contextual information about object classes and appearances, combined with its advanced architecture and modularity, makes it a powerful tool for addressing diverse challenges in the field of urban infrastructure detection and beyond. However, to leverage its full potential in new problem domains, the network must undergo specific training tailored to the target task (Redmon et al., 2016; Bochkovskiy et al., 2020). This adaptability further adds to the model's value and makes it a valuable asset in various fields of application.

2. Related works

Widely used approaches for urban infrastructure element recognition are based on the use of LiDAR systems or images. When using data derived from a LiDAR survey, the analysis is limited to the geometry of the objects by segmenting and separating objects to identify their nature (de Andrade Peixoto and Centeno, 2020; Li et al., 2019). When using close-range images, the methods can be distinguished from those using colour or grey images (Cao et al., 2019; Krišto et al., 2020). Approaches based on grayscale images allow us to effectively define a large part of the objects' geometry, but it is obvious that the use of colour enables to prevent or reduce false positives and improve their semantic interpretation. However, when the image database is properly integrated and uses modern techniques, it can provide better planning and analysis related to urban infrastructure elements. More recently, the advent of artificial intelligence methods, especially deep learning, has facilitated the task of object detection and segmentation in images (Cheng, 2020; Wu et al., 2020).

The vast list of deep learning models for object detection includes two-stage detectors such as R-CNN (Region-Based Convolutional Neural Network), Fast R-CNN, Faster R-CNN, Mask R-CNN, and those involving one stage, namely: SSD (Single Shot Detector), RetinaNet, YOLO (You Only Look Once) and its extensions like YOLOv2, YOLOv3, YOLOv4, YOLOv5, to name a few.

Both one- and two-stage detectors present advantages and disadvantages (Liu et al., 2020). Two-stage detectors, such as R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), Faster R-CNN (Girshick, 2015), and Mask R-CNN (He et al., 2017), train end-to-end convolutional neural networks (CNNs) for region classification (Liang et al., 2015). First, they generate a set of locations where the object is suspected to be found (candidate regions) through a Proposed Region Network (RPN) on the feature map. They then classify each candidate region as either an "object" or "background" while simultaneously running location regression (Girshick et al., 2014; Erhan et al., 2014; Szegedy et al., 2014; Li, 2021). The two-stage detectors have improved in a variety of ways (accuracy, speed, performance, etc.). However, some problems persist, such as the time required to train the network, the selective search algorithm generating proposals for bad regions and the fact that it is not efficient when implemented in real-time.

Region-based Convolutional Neural Networks (RCNN) and their faster variants extract many region proposals from the input image and use a CNN to perform forward propagation on each region proposal to extract descriptors and then use these descriptors to predict the class and bounding box of each proposal. Their success is largely due to the transfer of supervised pre-trained image representations to image classification for object detection. These methods, however, require a direct pass through the convolutional network to extract features for each proposed object, leading to a heavy computational load (Girshick et al., 2014, Jiang and Learned-Miller, 2017).

From R-CNN to Faster R-CNN, many improvements have occurred. One of the key improvements from R-CNN to Fast R-CNN is that direct CNN propagation is performed on the entire image, instead of feeding each distorted proposal image region to the CNN. It also introduces the region of interest clustering layer, so that descriptors can be extracted in the same manner for regions of interest, even if they have different shapes.

Faster R-CNN, proposed by Ren et al. (2015), is an improvement of Fast R-CNN. It uses a new Proposed Region Network (RPN) to generate proposed regions, which saves time compared to former algorithms like selective search. It uses the RoIPool (Region of Interest Pooling) layer to extract a fixed-length feature vector from each proposed region and performs classification and bounding box regression.

Following recent DL developments in object detection and classification, the Mask R-CNN model was proposed by He et al. (2017). Mask R-CNN adds a third branch that generates the object's mask. Additionally, mask output is different from class and box output, requiring a much finer spatial layout to be extracted from an object.

Alternatively, one-stage detectors have been used, such as the SSD or the YOLO family, proposed by Joseph Redmon and Ali Farhadi in 2015, which directly predict the output without going through the region proposal stage (Redmon et al., 2016). The single-stage detector approach applies a single neural network to the full image, then divides the image into regions, predicting bounding boxes and probabilities for each region, i.e., the bounding boxes are weighted by the predicted probabilities.

YOLO family algorithms only look at an image once to predict which objects are present and where they are (Redmon et al., 2016; Bochkovskiy et al., 2020). YOLO treats the object detection problem as a regression problem. In the YOLO detector, a downsampled feature map is divided into grid cells. For each grid cell, fully connected layers are trained to detect objects that are centred within this cell using the entire image as spatial support. YOLO has weaknesses for small objects and object groups, that cluster within a single cell. So, YOLOv2 (Redmon and Farhadi, 2017) is an improvement over the original YOLO algorithm. The main improvement in YOLOv2 is the use of anchor boxes. Anchor boxes are a set of predefined bounding boxes of different aspect ratios and scales.

The YOLOv3 model uses logistic regression to predict the objectivity score of each bounding box (Redmon and Farhadi, 2018; Cheng, 2020). In YOLOv3, three different layers with three different strides are used to predict classes and precise positions for the anchor boxes. YOLOv4 (Bochkovskiy et al., 2020) versions have optimal speed and accuracy compared to the previous versions of object detectors. YOLOv4 was specifically designed for production systems and optimised for parallel computations.

YOLOv5 is a one-stage detector proposed by Glenn Jocher in 2020. YOLO-v5 comes in several variants with respect to the computational parameters as presented in Table 1. It consists of four different versions, namely a: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, classified based on their memory storage size, but their underlying principle remains the same. YOLOv5x has the largest memory storage size, while YOLOv5s has the smallest (Jocher, 2021; Jung and Choi, 2022). The main improvement in the YOLOv5 architecture is the integration of the focus layer, represented as a single layer, which replaces the first three layers of YOLOv3. This integration reduced the number of layers and parameters, while also increasing the forward and backward speed with minimal impact on mAP (mean Average Precision). As a result, YOLOv5 is highly advantageous for object detection and recognition in terms of detection accuracy and computational complexity.

Table 1: YOLOv5 internal variant compararison.

Model	Average precision (@50)	Parameters	FLOPs
YOLOv5s	55.8%	7.5M	13.2B
YOLOv5m	62.4%	21.8M	39.4B
YOLOv5l	65.4%	47.8M	88.1B
YOLOv5x	66.9%	86.7M	205.7B

Source: Jocher (2021).

Other versions such as YOLOv6 (Li et al., 2022), YOLOv7 (Wang et al., 2023) and YOLOv8 (Reis et al., 2023) appeared, aiming to increase and optimise speed and accuracy. This illustrates the current research trend, where speed and accuracy are the two prerequisites for which object detection algorithms are examined. YOLO is a general-purpose model, which means that it can be used for different purposes, and the network must be trained to adapt to new problems (Redmon et al., 2016; Bochkovskiy et al., 2020).

Table 2 shows some studies of various machine learning model architectures, including performance evaluation metrics and processing time. The evaluation of these models was conducted using different datasets, making a direct comparison with the results of the current study challenging. Despite the shared similarities, such as the use of high-resolution RGB images, this analysis aims to provide a comprehensive understanding of the various approaches employed and generate valuable insights into the performance and effectiveness of different models.

The core challenge highlighted in the studies presented in Table 2 are based in improving the precision of detection and classification, implying improvements to the models, whether through combination or fine-tuning, as demonstrated in the research of Amisse et al. (2021) and Li et al. (2021). Additionally, the diversity in data sources employed for the detection task is noteworthy, covering a specific database comprising short-range images, as shown in the Amisse et al. (2021) and He et al. (2021). In the Li et al. (2021) and Yang et al. (2022), public databases were used, while Liu et al. (2022), Chen et al. (2022) and Yu et al. (2022) used aerial images. The amount of data plays a crucial role in improving the performance of deep learning models, and the single-stage models have shown relatively lower instance inference times.

Table 2: Description of different studies comparing the proposed model.

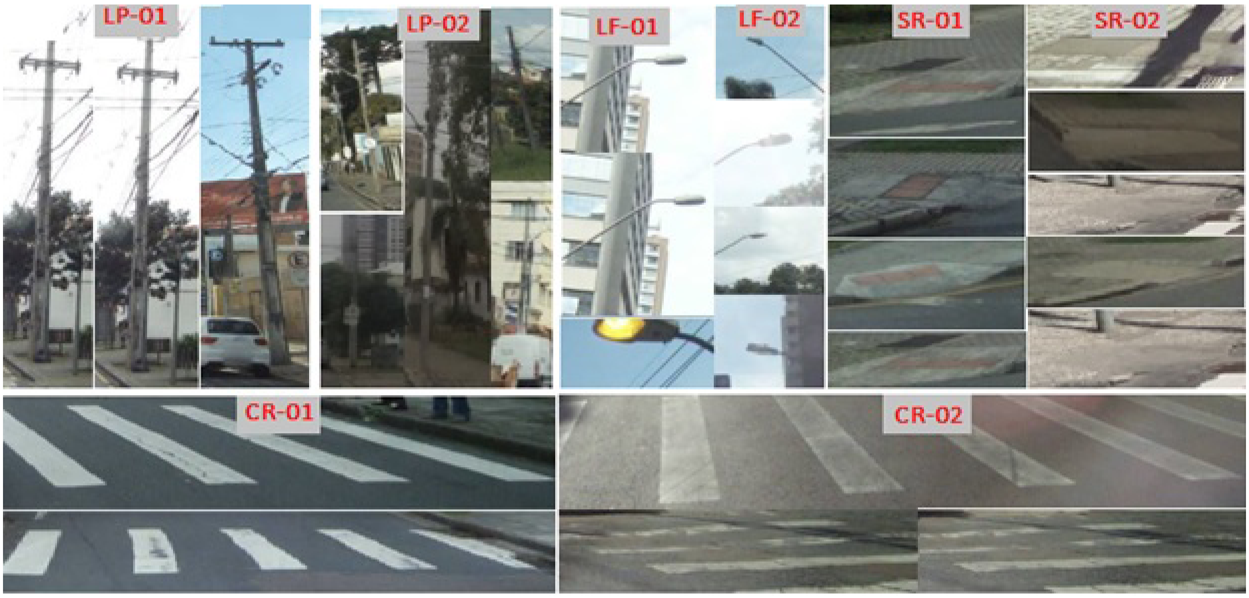
Authors	Model	Dataset	Precision	Recall	mAP	Average speed (ms)
Amisse et al., 2021	Faster RCNN Inception v2	700 RGB images of Pedestrian	82.5	76.4	-	-
	SSD Inception v2		70.7	64.0		
	SSD Mobilenetv2		62.1	59.1		
He et al., 2021	SSD	1350 RGB images of railway track	-	-	85.9	0.11
	YOLO V4				88.0	0.12
Liu et al., 2022	FRCNN	UAV Images	52.5	48.0	49.4	96
	YOLOv5(s)	6471 images	79.5	44.2	47.9	46
Chen et al., 2022	Yolov5	UAV Images	-	-	44.7	-
		742 images				
Yang et al., 2022	Original YOLOv5	Public dataset DOTA -2806 images	-	-	56.0	-
	Improved Yolov5				69.0	
Yu et al., 2022	YOLOv5(x) YOLOv5(x6)	Aerial images	-	-	75.3	-
					76.0	
Li et al., 2021	YOLOv3	KAIST dataset (infrared images, single channel)	-	-	79.6	25
	YOLOv4				81.0	37
	YOLO-FIR -Infrared images				93.1	12
	YOLO-FIRI (improved)				98.3	14

3. Material and Methods

3.1 Data source

For this research, Mapillary API images associated with OpenStreetMap were used (accessed in: https://www.mapillary.com/dataset/vistas). Mapillary is a global VGI (Volunteered Geographic Information) project, an initiative that was launched in Sweden in April 2014 to provide open-access images of places on the planet (Seto and Nishimura, 2022). Therefore, API Mapillary has become one of the largest sources of images taken from cars or by people on foot, from a horizontal perspective, shared around the world. Mapillary's images, show multiple objects in an urban scene, among them, elements of urban infrastructure, people and vehicles. With the images from the Mapillary collaborative platform, computer vision can be used to scale and automate the mapping.

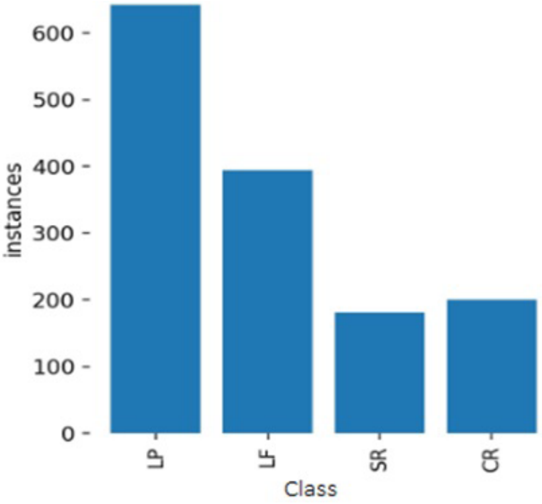
Figure 1 presents the general mosaic of some characteristics of the urban infrastructure elements considered in the present work. Therefore, urban infrastructure elements (*LP- Power grid pole*, *LF- Public lighting*, *SR- Accessibility*, *CR-Crosswalk*) have different visual characteristics, for example, classes close to the camera at the time of image capture (higher resolution) and others far from the camera (lower resolution) and also some objects in a state of degradation. The visual characteristics of the objects in an image contribute to class typification. This contributes to the model's performance at predicting the classes.



SR-01- Accessibility with signage; SR-02 - Accessibility without signs; CR-01 - Crosswalk with visible signage; CR-02 - Crosswalk without visible signage; LP-01 - Power grid pole with high resolution; LP-02 - Power grid pole with lower resolution; LF-01- Public lighting with better visual representation; LF-02 - Public lighting with low visual representation.

Figure 1: Different characteristics of urban infrastructure elements.

To assess the feasibility of automatically extracting elements of urban infrastructure from these images, a database was created, composed of only 4 classes of objects (Figure 2), namely: power grid poles, crosswalks, accessibility features, and public lighting. Thus, 615 images were downloaded from the API Mapillary platform, taken on main urban roads in the city of Curitiba. During the acquisition process, some images were discarded, allowing for a broader coverage of the study area and capturing different image scenarios. The dimensions of the images used were 1600 x 1200 pixels (width and height). The dataset for the experiment was randomly selected and partitioned into three proportions: 70% (496 images) for training, 20% (126 images) to compose the validation set, and approximately 10% (20 images) for model testing.



LP - Power grid pole; LF - Public lighting; SR - Accessibility; CR - Crossawalk.

Figure 2: The number of instances of each class.

To implement the model, the bounding boxes of the different instances in the images were manually annotated using MAKESENSE (Piotr Skalski, MakeSense, 2019, provided in: https://github.com/SkalskiP/make-sense/), an open-source and free tool that supports output file formats such as YOLO, VOC XML, VGG JSON, and CSV. The experiment was carried out in the Google Colab environment, based on the graphics processing unit (GPU) using python language.

3.2. Methods

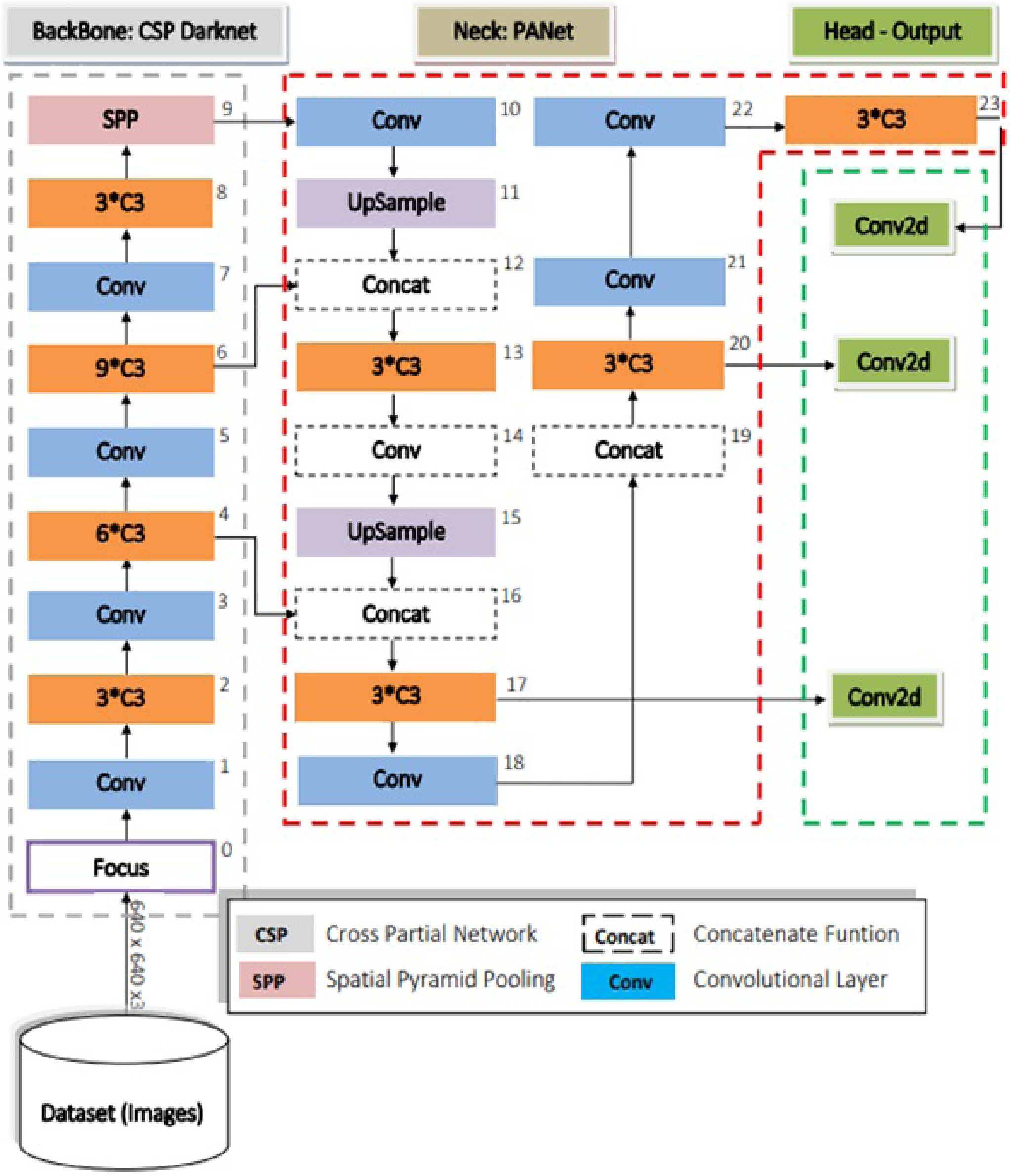
To detect urban infrastructure elements, the YOLOv5 deep neural network was used. YOLOv5 is a single-stage detector that includes three important parts (Figure 3), namely: (i) Backbone: CSPDarknet; (ii) Neck: PANet and; (iii) Head: Yolo Layer.

The backbone module is mainly used to extract important features from the input image. This is accomplished using Cross Stage Partial Networks (CSPNet) as the backbone, which provides a large amount of descriptors from an input image (Jocher et al., 2020; Li et al., 2021; Lawal et al., 2021; Hassan et al., 2023). An advantage of CSPNet is that it provides a significant improvement in processing time with deeper networks (Jocher et al., 2020; Li et al., 2021; He et al., 2022). In addition, an SPP (Spatial Pyramid Pooling) layer is used to concatenate the results obtained through four clustering windows to solve the alignment problem for anchors and feature maps (Jocher et al., 2020; Li et al., 2021; He et al., 2022). The process uses a neck module to generate feature pyramids. The feature pyramids enable models to generalise in size, allowing them to identify the same object with different sizes and scales. They also allow models to perform well even when objects are partially occluded (Jocher et al., 2020; Li et al., 2021; Lu et al., 2022). Other models use different types of feature pyramid techniques such as FPN (Feature Pyramid Networks for object detection), Bi-directional feature pyramid network (BiFPN), Path Aggregation Networks (PANet), etc. Finally, the head module is used to perform the final part of detection, applying anchor boxes on features, generating final output vectors with class probabilities, objectivity scores and bounding boxes. The head module used in YOLOv5 is the same as in previous versions of YOLOv3 and YOLOv4 (Jocher et al., 2020; Li et al., 2021; Lawal et al., 2021; He et al., 2022; Lu et al., 2022; Hassan et al., 2023).

In the present model, both activation and optimization functions were carefully considered. Activation functions, such as Leaky ReLU and sigmoid, play a crucial role in deep networks, influencing the model's ability to learn complex patterns (Jocher et al., 2020; Li et al., 2021; Lu et al., 2022). Leaky ReLU was used in intermediate/hidden layers, while sigmoid was applied in the final detection layer.

The backbone network, often including CSPDarknet53 or CSPDarknet53-SPP, uses Leaky ReLU activations to introduce non-linearity and aid in learning rich representations. The neck of the YOLOv5 model continues to use Leaky ReLU activations for further feature processing. In the YOLOv5 head, which contains the detection layers, the sigmoid activation function is crucial for bounding box regression and objectness score prediction. It normalizes output values between 0 and 1, making them interpretable as probabilities (Jocher et al., 2020).

The standard optimization function SGD (Stochastic Gradient Descent) was chosen for training, updating model weights based on gradients. According to Jocher et al. (2020), Li et al. (2021), and Lu et al. (2022), the composite loss is computed based on Objectivity Score (confidence for object presence), Class Probability Score (probability of object class), and Bounding Box Regression Score (refinement of box coordinates). The Binary Cross-Entropy with Logits Loss function was used for class probability and object scores, suited for binary classification tasks in object detection.



Source: Adapted from: Jocher et al. (2020).

Figure 3: The Yolov5 network architecture. Main parts of Yolov5: (i) Backbone; CSPDarknet; (ii) Neck: PANet; (iii) Head: Yolo Layer. Data are first entered into CSPDarknet for feature extraction and then fed into PANet for feature fusion. Finally, Yolo Layer generates detection results (class, score, location, size).

The parameter settings included the learning rate: 0.0005; Momentum of: 0.9; Batch size: 32; Steps per epoch: 50; Number of epochs: 200. The setting of hyperparameters was an iterative process to achieve the best results based on the specific characteristics of the dataset and the observed training performance.

To evaluate the model's performance, the following metrics were computed (equations 1, 2 and 3): precision (P), recall (R) and mean average precision (mAP). These metrics were computed from the number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) results. Precision represents the model's ability to identify successfully classified objects, that is, the percentage of correct predictions. Recall is the model's ability to find all relevant objects, representing the percentage of true positives detected in the ground truth. The *mAP* incorporates precision and recall, which makes mAP a suitable metric to evaluate most object detection applications.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

$$\mathrm{mAP} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{APi} \tag{3}$$

$$\mathrm{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{4}$$

where *APi* is the average precision of classes and *N* is the number of classes. The *AP* metric relies on Intersection Over Union (IoU), represented by Equation 4, and describes how boxes overlap in object detection. It provides an output box that wraps objects perfectly and allows the calculation of location errors in object detection models, in which the grid cell is responsible for predicting the bounding boxes and their confidence scores. Therefore, *IoU = 1* if the predicted bounding box equals the actual box.

4. Results

The dataset instances include four categories of objects referring to urban infrastructure elements. Figure 4 shows the location and detection probabilities of the classes. The algorithm has difficulty detecting certain classes in some images, especially when the classes are located far from the camera, or have a low visual representation. The algorithm has good localisation ability since it can easily detect the object, but on the other hand, the model presents difficulty in categorising certain classes. The lower classification rate of certain classes may be associated with the difficulty in typifying some classes, probably attributed to a certain state of degradation and lower resolution. This contributes to the reduction of the IoU value, thus reducing the model's ability to predict these classes.

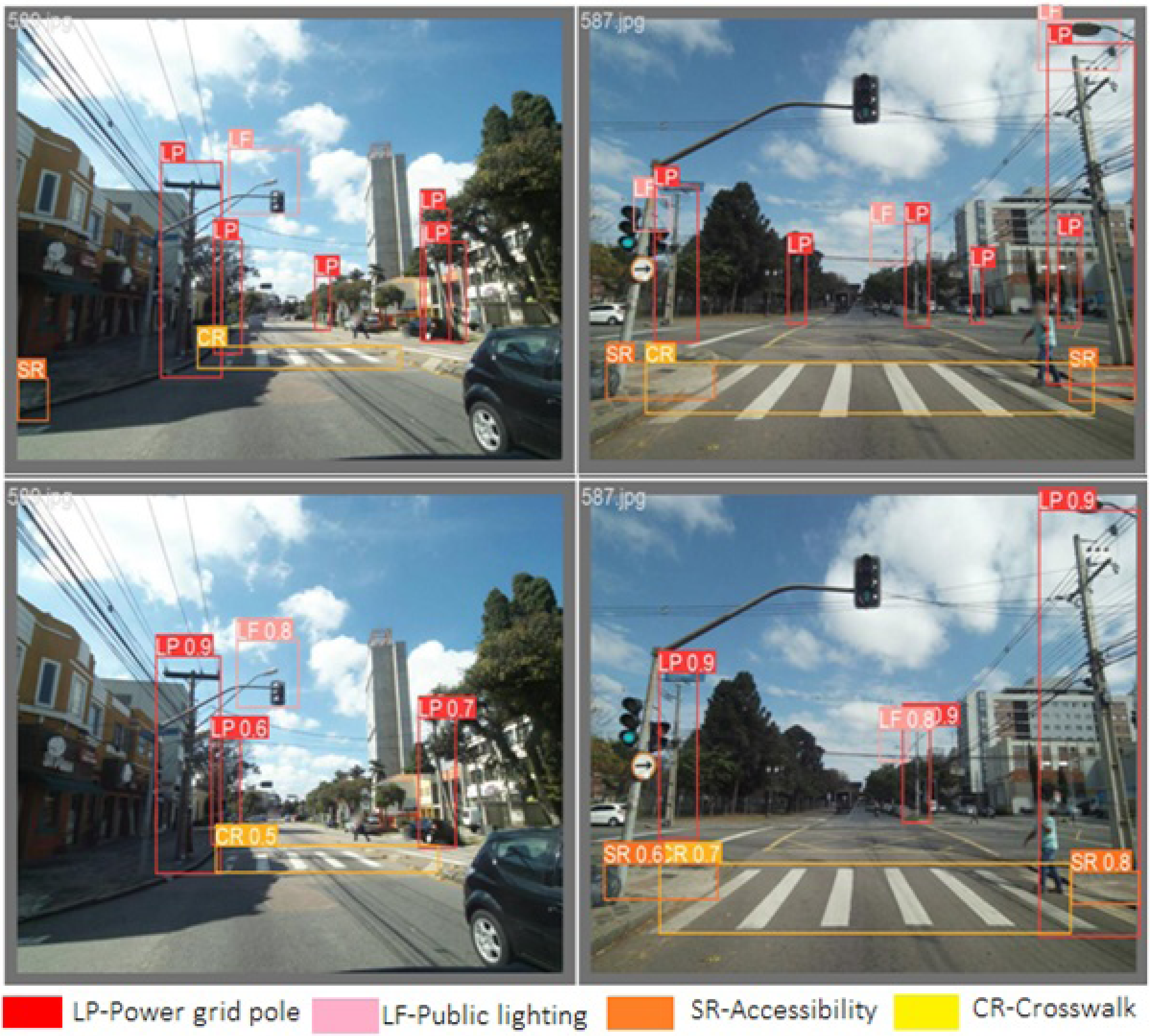


Figure 4: Model output of the evaluated classes.

The numerical results of the model are presented in Table 3, which reports the precision, recall, mAP and processing time.

Table 3: Classification performance of different objects.

Class	Validation					Test			
	Instances	P(%)	R(%)	mAP(%)	T(ms)	P(%)	R(%)	mAP(%)	T(ms)
Power grid pole	516	84	69	78	0.383	82	67	78	0.371
Crosswalk	281	89	71	79	0.403	87	70	79	0.410
Accessibility	216	81	54	61	0.223	82	52	59	0.218
Public lighting	178	79	60	64	0.260	80	60	62	0.255

The algorithm produced an average accuracy rate (mAP) of 78%, 79%, 61%, and 64% for the detection of power grid poles, crosswalks, accessibility, and public lighting, respectively. When comparing these results with the performance on the test dataset, as shown in Table 2, we observe a generally satisfactory performance for most categories. However, it is worth noting that the classification of the accessibility category achieved a relatively low average accuracy of 61% on both validation and test datasets. This can be attributed to the inherent difficulty of identifying this class in the images, as it may share visual similarities with other objects, particularly sidewalks. Similarly, the public lighting category also exhibited a relatively low average accuracy of approximately 64%. This can be attributed to the limited visual representation of the class compared to other objects and, in some cases, the size and resolution of the objects in the images can be challenging for accurate detection, especially when they are positioned far from the camera.

On the other hand, the categories of "crosswalks" and "power grid poles" exhibited relatively high mAP scores, around 79% and 78%, respectively, on both validation and test datasets. This can be attributed to the distinct visual features and clear typification of these categories in the images. Crosswalks are often visible and well-signposted, contributing to better delineation and identification. Likewise, power grid poles, with their geometric shapes and distinct appearance, stand out prominently against the background, facilitating their accurate detection and leading to better performance of the model in these categories.

Overall, the model's performance on the validation dataset closely matches its performance on the test dataset, indicating that the model's generalisation ability is robust. The differences in accuracy for specific categories highlight the inherent challenges in object detection tasks and the impact of visual characteristics on model performance. Understanding these differences can guide further improvements in the model and help address potential challenges in future applications.

The experiments were carried out in the cloud, in the Google Colab environment, taking advantage of the available computing resources to increase reproducibility and optimize memory and hardware requirements. One advantage of using such environments is the ability to perform processing with low demand on local computational resources. When analyzing the processing time to detect each set of instances it was found that, on average, the model requires only 0.32ms per inference. This inference time is satisfactory for real-time operations, indicating the model's ability to quickly respond to object detection tasks.

In the second step, the classification confusion rate between the proposed objects was measured, with the help of the confusion matrix shown in Table 4. It is possible to observe that the precision of some classes, such as "accessibility" and "public lighting" is relatively low, around 60% and 68%, corresponding to 130 and 121 correctly classified classes, respectively. The low rating for these classes may be associated with the difficulty of their classification and the confusion created by the background. The best classification rate was obtained in two classes, "power grid poles" and "crosswalks", around 75%, corresponding to 387 power grid poles and 211 pedestrian crosswalks were correctly classified. This means that these classes have comparably good detection rates. Table 4 also presents the rates created by the background in different classes under study. For example, the classes of accessibility and public lighting show a high rate of false positives, which may be associated with long distances from the object to the camera, weak power, and low resolution.

Table 4: Confusion matrix generated from the experiment.

Class	Power grid pole	Crosswalk	Accessibility	Public lighting	Background FP
Power grid pole	0.75 (387)	0	0	0	0.41 (212)
Crosswalk	0	0.75 (211)	0.01 (2)	0	0.13 (37)
Accessibility	0	0.01 (3)	0.60 (130)	0	0.18 (39)
Public lighting	0	0	0	0.68 (121)	0.28 (50)
Background FN	0.25 (129)	0.24 (67)	0.39 (84)	0.32 (57)	-

Figure 5 shows the evolution of five quality descriptors. Thus, after 100 training epochs, the loss function of the training and validation sets was computed, including the loss of the object detection and classification frame. From left to right: the regression box loss (train/box_loss), object loss (train/obj_loss), classification loss (train/cls_loss), and the performance metrics of accuracy and recall. The first row corresponds to the training set and the second, to the validation.

Detection box loss indicates how well the algorithm can locate the centre of an object, or how much the predicted bounding box covers a detected object. Object loss is a measure of the probability that an object exists in a proposed region of interest. A high value means it is more likely that the image window will contain an object. The smaller the value of the loss function, the greater the accuracy. Classification loss gives an idea of how well the algorithm can correctly predict the class of a given object. The smaller the loss value, the more accurate the classification.

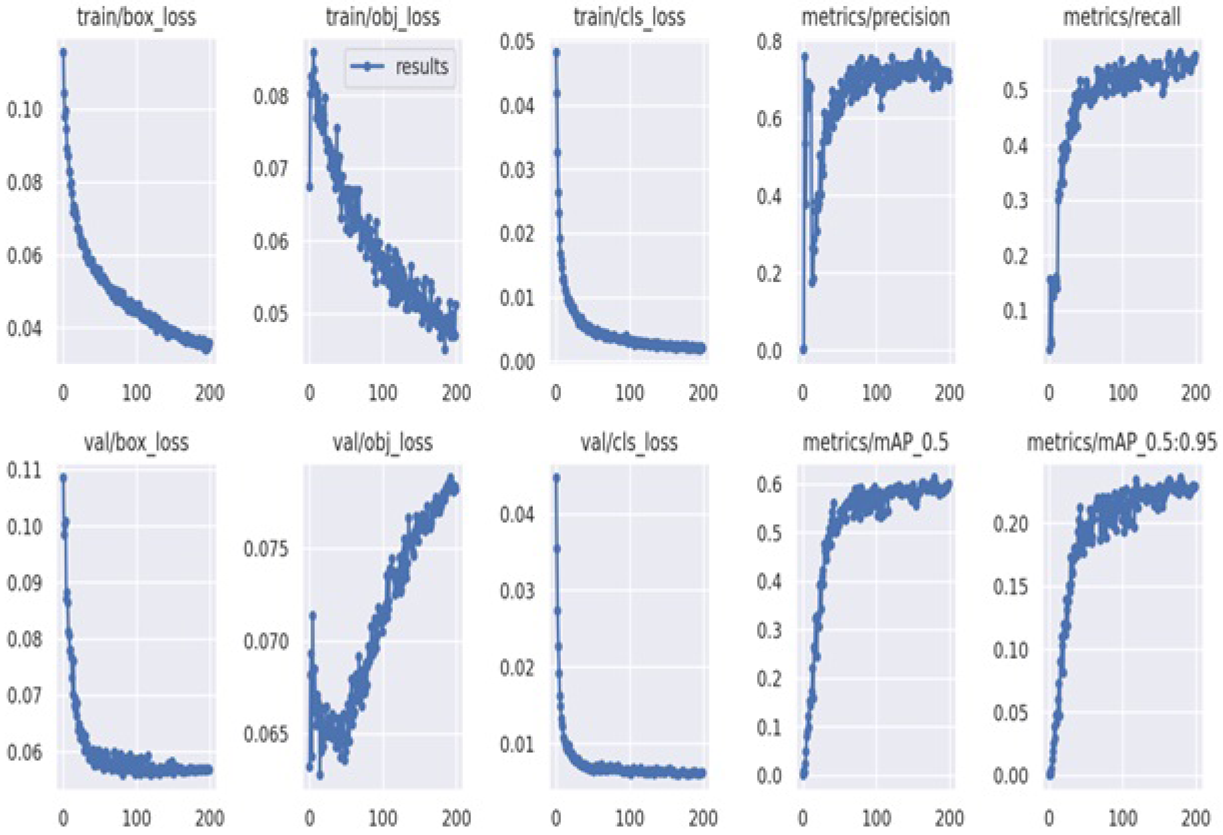


Figure 5: Performance comparison of three types of loss for the training and validation set: (i) box loss; (ii) loss of objectivity; (iii) loss of classification; (iv) accuracy; (v) recall and; (vi) mean accuracy (mAP) over training epochs.

In the experiment, the value of the loss function had a downward trend during the training process. This means that the Stochastic Gradient Descent function tried to optimise the network and the network weights and parameters were constantly updated. Before 50 training epochs, the loss function value dropped drastically and when it reached 50 training epochs, the loss function value gradually decreased. Even after 50 epochs, the average retrieval rate and accuracy improved rapidly. When the model reached 50 epochs, the loss curves of the training and validation sets showed no downward trend and other indices were also stabilised.

4.1 Discussion

The experimental results showed that it is possible to map elements of urban infrastructure using public images obtained from the collaborative platform Mapillary, associated with OpenStreetMap. These elements include power grid poles, public lighting, pedestrian crosswalks, and accessibility features, which are identified through the use of deep learning models like those present in the YOLO family. After training the YOLOv5 model, it became feasible to detect the proposed objects in new images that were not used in either the training or validation sets. The tests demonstrated that the algorithm is capable of identifying power grid poles (mAP=78%) and pedestrian crosswalks (mAP=79%) with a higher degree of certainty. However, the degree of certainty is lower for public lighting (mAP=64%) and accessibility features (mAP=61%).

Based on the confusion matrix, the model demonstrates a higher classification capacity for the classes "power grid poles" and "crosswalks," achieving approximately 75% accuracy for these classes, corresponding to 387 classes of power grid poles and 211 of pedestrian crosswalks. However, the performance is comparatively lower for the "accessibility" and "public lighting" classes, with accuracy values around 60% (130 classes) and 68% (121 classes), respectively. A noteworthy observation is that the model's performance tends to decrease with object size - detecting small or very large objects in the image can impact accuracy. Smaller objects can be challenging to identify and classify, while larger objects may lose important details or get cut off. Additionally, reduced model performance is influenced by image resolution, as low-resolution images lack the necessary detail for the accurate localisation and classification of objects. These factors contributed to difficulties in correctly recognising the "accessibility" class, often confused with "pavement," especially when lacking explicit signage like tactile flooring for the visually impaired or when access ramps on main roads do not exhibit significant slopes. The absence of these structural elements that aid class differentiation contributed to the model's lower classification rate. The algorithm also faced challenges in recognising the "public lighting" class, mainly due to its visually intricate nature, often associated with small object size.

The challenges identified through the confusion matrix offer valuable insights into the limitations of the object detection model. Acknowledging these difficulties can guide future enhancements to the algorithm and lead to the selection of more suitable models and techniques for each specific class of objects. Despite the observed challenges, the results remain promising and relevant, especially considering the complex and diverse nature of the object detection images in the studied scenario. This underscores the model's potential and emphasises the importance of continuous improvement in its classification and identification capabilities. Understanding these limitations is crucial for ongoing progress in this field, with the goal of achieving even higher levels of accuracy and efficiency in the future. By addressing these challenges proactively, we will be better equipped to handle the diversity of objects and scenarios encountered in real-world applications, making the object detection model a more robust and reliable tool for various uses.

The low detection rate of objects in certain classes is highlighted by Redmon et al. (2016) as they discuss the limitations of the YOLO model. The YOLO model imposes spatial constraints on bounding box predictions, with each grid cell predicting only two boxes and being associated with only one class. This constraint restricts the number of nearby objects that the model can detect, particularly smaller objects. Consequently, during model training to optimise detection performance, errors are treated equally for both large and small bounding boxes. However, a small error in a small box has a significantly larger impact on the area overlap metric (IoU) than a small error in a large box. As a result, this directly affects the accurate localisation of detected objects. The current state-of-the-art demonstrates that by reducing the threshold value of IoU (e.g., to IoU=0.8), the accuracy reaches around 66.67%, with only 4 out of 6 classes correctly detected.

4.2. Conclusions

Technological development and the application of photogrammetry are of paramount importance in urban management, particularly in situations where there is limited information to continuously monitor the operational elements of urban infrastructure. The current study introduced an alternative approach utilising deep learning techniques and publicly available images from the Mapillary collaborative platform. The experiment demonstrated that the YOLOv5 model enables precise detection, identification, and rapid, cost-effective classification of object classes in urban environments, thereby enhancing decision-making processes which are traditionally resource-intensive and laborious. So, the model proves to be adaptable for real-time object detection across various scenarios and outperforms two-stage models that require substantial computational capabilities due to their multi-stage nature.

The study demonstrates the algorithm's exceptional localisation capability but also highlights challenges in classifying certain object classes. The conducted experiment reveals that the model achieves higher predictability in detecting power poles (mAP=78%) and pedestrian crossings (mAP=79%). However, its predictability decreases when detecting public lighting (mAP=64%) and accessibility (mAP=61%) classes. This low rate of recognition accuracy and the elevated false positive rate can be attributed to factors such as the object's distance from the camera, its size, and the relatively low image resolution. These elements collectively impact the algorithm's ability to accurately identify and classify objects in these particular classes.

For future work, it is recommended to replicate the experiment with a larger sample size and maintain a consistent proportion of instances across different classes, which was not feasible in this study. This approach aims to address the issues of low recognition rates and high false positive rates. Additionally, employing visible and infrared images, or fusing both types of images, could offer a potential solution to mitigate these challenges. Evaluating the possibility of enhancing results through the combination of different models, such as Faster R-CNN and YOLO, may also be fruitful in reducing background interference. Furthermore, leveraging transfer learning to initialise the model parameters could prove beneficial in improving the algorithm's performance for object detection tasks. So, future research can make significant strides in advancing the model's accuracy and efficacy for urban infrastructure analysis.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ACKNOWLEDGMENT

We extend our thanks to the National Council for Scientific and Technological Development (CNPq) and The World Academy of Sciences (TWAS) for the scholarship [Process: 167085/2018-2]. We would also like to thank the Postgraduate Program in Geodetic Sciences at the Federal University of Paraná for their support.

AUTHOR´S CONTRIBUTION

Author 1: designed and implemented the algorithm, data collection, conducted experimental study and wrote the paper; Author 2: study conception, formal analysis, development of research methods, critical review and final revision; Author 3: substantial contributions to conception, analysis, and critical review; Author 4: substantial contributions to conception, analysis, and critical review; Author 5: development of research methods, critical review; Author 6: study conception, formal analysis, development of research methods, critical review and final revision.

REFERENCES

Amisse, C., Jijón-Palma, M. E., & Centeno, J. A. S. (2021). Fine-tuning deep learning models for pedestrian detection. *Boletim de Ciências Geodésicas*, *27*.

Bai, Q., Li, S., Yang, J., Song, Q., Li, Z., & Zhang, X. (2020). Object detection recognition and robot grasping based on machine learning: A survey. *IEEE access*, *8*, 181855-181879.

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Cao, L., Shang, Y., Zhao, J., & Li, Z. (2019). Comparison of grayscale image colorization methods in different color spaces. In *Advances in Graphic Communication, Printing and Packaging: Proceedings of 2018 9th China Academic Conference on Printing and Packaging* (pp. 290-300). Springer Singapore.

Chen, Z., Cao, L., & Wang, Q. (2022). Yolov5-based vehicle detection method for high-resolution UAV images. *Mobile Information Systems*, *2022*.

Cheng, R. (2020). A survey: Comparison between Convolutional Neural Network and YOLO in image identification. In *Journal of Physics: Conference Series* (Vol. 1453, No. 1, p. 012139). IOP Publishing.

de Andrade Peixoto, E. B., & Centeno, J. A. S. (2020). Mobile terrestrial lidar data to detect traffic sign and light pole. *Brazilian Journal of Development*, *6*(6), 39506-39518.

Dildar, M., Akram, S., Irfan, M., Khan, H. U., Ramzan, M., Mahmood, A. R., ... & Mahnashi, M. H. (2021). Skin cancer detection: a review using deep learning techniques. *International journal of environmental research and public health*, *18*(10), 5479.

Elallid, B. B., Benamar, N., Hafid, A. S., Rachidi, T., & Mrani, N. (2022). A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving. *Journal of King Saud University-Computer and Information Sciences*, *34*(9), 7366-7390.

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2147-2154).

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

Hassan, M., Hussain, F., Khan, S. D., Ullah, M., Yamin, M., & Ullah, H. (2023). Crowd counting using deep learning based head detection. *Electronic Imaging*, *35*, 293-1.

He, D., Li, K., Chen, Y., Miao, J., Li, X., Shan, S., & Ren, R. (2021). Obstacle detection in dangerous railway track areas by a convolutional neural network. *Measurement Science and Technology*, *32*(10), 105401.

He, H., Chen, Q., Xie, G., Yang, B., Li, S., Zhou, B., & Gu, Y. (2022, October). A Lightweight Deep Learning Model for Real-time Detection and Recognition of Traffic Signs Images Based on YOLOv5. In *2022 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* (pp. 206-212). IEEE.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

Jiang, F. & Hao, L., (2018, November). A new facial detection model based on the Faster R-CNN. In *IOP Conference Series: Materials Science and Engineering* (Vol. 439, No. 3, p. 032117). IOP Publishing.

Jiang, H., & Learned-Miller, E. (2017, May). Face detection with the faster R-CNN. In *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)* (pp. 650-657). IEEE.

Jocher, G., Stoken, A., Borovec, J., Changyu, L., Hogan, A., Chaurasia, A., ... & Yu, L. (2021). ultralytics/yolov5: v4. 0-nn. SiLU () activations, Weights & Biases logging, PyTorch Hub integration. *Zenodo*.

Jung, H. K., & Choi, G. S. (2022). Improved yolov5: Efficient object detection using drone images under various conditions. *Applied Sciences*, *12*(14), 7255.

Krišto, M., Ivasic-Kos, M., & Pobar, M. (2020). Thermal object detection in difficult weather conditions using YOLO. *IEEE access*, *8*, 125459.

Lawal, O. M., Huamin, Z., & Fan, Z. (2021, November). Ablation studies on YOLOFruit detection algorithm for fruit harvesting robot using deep learning. In *IOP Conference Series: Earth and Environmental Science* (Vol. 922, No. 1, p. 012001). IOP Publishing.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., ... & Wei, X. (2022). YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*.

Li, S., Li, Y., Li, Y., Li, M., & Xu, X. (2021). Yolo-firi: Improved yolov5 for infrared image object detection. *IEEE access*, *9*, 141861-141875.

Li, W. (2021, March). Analysis of object detection performance based on Faster R-CNN. In *Journal of Physics: Conference Series* (Vol. 1827, No. 1, p. 012085). IOP Publishing.

Li, Y., Wang, W., Li, X., Xie, L., Wang, Y., Guo, R., ... & Tang, S. (2019). Pole-like street furniture segmentation and classification in mobile LiDAR data by integrating multiple shape-descriptor constraints. *Remote Sensing*, *11*(24), 2920.

Liang, M., & Hu, X. (2015). Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3367-3375).

Linaza, M. T., Posada, J., Bund, J., Eisert, P., Quartulli, M., Döllner, J., ... & Lucat, L. (2021). Data-driven artificial intelligence applications for sustainable precision agriculture. *Agronomy*, *11*(6), 1227.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International journal of computer vision*, *128*, 261-318.

Liu, W., Quijano, K., & Crawford, M. M. (2022). YOLOv5-Tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 8085-8094.

Lu, Y., Sun, C., Li, X., & Cheng, L. (2022, May). Defect detection of integrated circuit based on yolov5. In *2022 IEEE 2nd International Conference on Computer Communication and Artificial Intelligence (CCAI)* (pp. 165-170). IEEE.

Muchuchuti, S., & Viriri, S. (2023). Retinal Disease Detection Using Deep Learning Techniques: A Comprehensive Review. *Journal of Imaging*, *9*(4), 84.

Ning, H., Yin, R., Ullah, A., & Shi, F. (2021). A survey on hybrid human-artificial intelligence for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, *23*(7), 6011-6026.

Oguine, K. J., Oguine, O. C., & Bisallah, H. I. (2022, November). YOLO v3: Visual and Real-Time Object Detection Model for Smart Surveillance Systems (3s). In *2022 5th Information Technology for Education and Development (ITED)* (pp. 1-8). IEEE.

Piotr Skalski (2019). Make Sense. url: https://github.com/SkalskiP/make-sense/ (visited on 24/04/2023).

Piotr Skalski, Make Sense, 2019. https://github.com/SkalskiP/make-sense/.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2023). Real-Time Flying Object Detection with YOLOv8. *arXiv preprint arXiv:2305.09972*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

Sarker, S., Jamal, L., Ahmed, S. F., & Irtisam, N. (2021). Robotics and artificial intelligence in healthcare during COVID-19 pandemic: A systematic review. *Robotics and autonomous systems*, *146*, 103902.

Seto, T., & Nishimura, Y. (2022). Analysis of the spatiotemporal accumulation process of mapillary data and its relationship with osm road data: A case study in japan. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *48*, 403-410.

Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, A review. *Cognitive Robotics*.

Sumit, S. S., Watada, J., Roy, A., & Rambli, D. R. A. (2020, April). In object detection deep learning methods, YOLO shows supremum to Mask R-CNN. In *Journal of Physics: Conference Series* (Vol. 1529, No. 4, p. 042086). IOP Publishing.

Szegedy, C., Reed, S., Erhan, D., Anguelov, D., & Ioffe, S. (2014). Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*. URL link: https://www.mapillary.com/dataset/vistas - accessed on April 24, 2022.

Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7464-7475).

Wu, H., Gao, W., & Xu, X. (2020). Corrections to "Solder Joint Recognition Using Mask R-CNN Method" [Mar 20 525-530]. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, *10*(6), 1069-1069.

Yang, C., Mengxing, H., Jinjin, Y., Siling, F., & Yuanyuan, W. (2022, August). Rotated Object Detection of High-Resolution Remote Sensing Image Based on Yolov5. In *2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)* (pp. 697-702). IEEE.

Yu, X., Lin, M., Lu, J., & Ou, L. (2022). Oriented object detection in aerial images based on area ratio of parallelogram. *Journal of Applied Remote Sensing*, *16*(3), 034510-034510.