THE WORLD BANK

IMPERIAL

Group Design Project 2024

BIG DATA ANALYSIS ON ROAD ATTRIBUTES

Academic Supervisors: Mohammed Quddus, Felix Feng, Shaofan Sheng, Hanlin Tian, Leah Camarcat

Imperial College London

Industrial Supervisors: Said Dahdah, Dipan Bose, Kazuyuki Neki, Diego Canales

World Bank

Group 4: Guillaume Amann, Frederic Dai, Rui Huang, Claudio Mena Cavieres, Tom Ralton, Ai Yu

Imperial College London

Table of Contents

Abstract

The World Bank encourages awareness of traffic and road safety incidents in its future projects. The Road Safety Screening and Appraisal Tool (RSSAT) helps to integrate these concerns into project investment studies, by proposing a quantitative approach to road safety assessment. With a comprehensive list of road attributes, the World Bank analysts can identify investment opportunities in safety enhancement.

Deep Learning would allow the efficient processing of large quantities of such data. The objective is to develop AI-based algorithms, leveraging purposefully trained Convolutional Neural Networks (CNN) for fast road attribute extraction from publicly available big data sources. This approach is particularly relevant in LMICs (Low and Middle-Income Countries) as it eliminates the need for costly data purchases or road video recordings.

Automating road feature detection would help the World Bank evaluate more projects, thereby increasing opportunities for expansion in LMICs.

1. Introduction

The World Bank's revised Environmental and Social Framework (ESF) now incorporates measures to address traffic and road safety issues. These standards require that all projects financed by the World Bank proactively avoid or mitigate potential road safety risks and their impacts. To facilitate the implementation of these measures, the World Bank has introduced the Road Safety Screening and Appraisal Tool (RSSAT). This guide provides detailed guidance to assess current road safety conditions divided in requirement categories. This study will focus on identifying relevant road variables such as: segment characteristics (physical median, pedestrian facilities, …), dominant roadside objects (safety barrier, posts, trees, …), intersection features, and pedestrian crossings.

In Low- and Middle-Income Countries (LMICs), road traffic accidents are a leading cause of death and serious injury. These countries often lack comprehensive road safety data and infrastructure, making the assessment and improvement of road safety a daunting task. Given the new standards from the World Bank, LMICs urgently need effective methods to identify and manage road safety risks.

Traditional road safety assessment methods are not only time-consuming and costly but are also difficult to implement in the absence of adequate data. Particularly in LMICs, collecting comprehensive road attribute data requires significant time and resources, hindering the evaluation of the safety impact of road projects. To meet the new standards set by the World Bank, these countries need to find more efficient and economical ways to conduct road safety assessments.

The objective is to develop AI-based algorithms using purposefully trained CNNs to quickly extract road attributes from publicly available big data sources. The methodology used for the project consists of performing a data collection from specifically selected regions and countries via open-source datasets, such as: Open Image Dataset v7, Google Street View, CeyMo[1] and Mapillary. The second phase consists of filtering and annotating the gathered pictures with the attributes that the RSSAT needs. The use of one extensive model is limited by the versatility of objects' shape to detect.

References

[1] Liu, X., Deng, Z., Lu, H., and Cao, L., 2017. Benchmark for road marking detection: Dataset specification and performance baseline. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6. IEEE.

This is where the approach defers from previous strategies: a modular multi-layer model (MLM) method was proposed which divided the object identification task and assigned a batch of attributes to purposefully trained models. This MLM is composed of four layers, one for each RSSAT requirement, and its modular structure lays the groundwork for potential future improvements.

2. Database Creation

The tailored requirements set by the World Bank has forced the development of new methods for comprehensive road feature detection. The absence of valid and representative data often prohibits a full understanding of road safety risks. Therefore, there is a need for specific road attribute dedicated datasets to feed the algorithm.

2.1. Data Collection

The data collection process has led to two distinct types of datasets. One type includes data downloaded from open-source databases and are already annotated – in this study, these are all sourced from Open Image Dataset v7 (OIDv7). The other type are mostly web scrapped via Google Street View (GSV) or require manual labelling (carried out on a free labelling software, namely Roboflow). These two datasets form the database which consists of both rural and urban locations, as the road infrastructure in these contexts varies a lot (e.g. both bike lanes and crossings are more present in urban areas rather than rural areas). Notwithstanding the above-mentioned categories, the database is composed of images from both roads and motorways.

Overall, the database gathers around 30,000 pictures from various origins:

- Open-source databases (OIDv7, CeyMo Benchmark Road Marking dataset)
- Web scrapped data from APIs (Google Street View, Mapillary)
- Manually collected data from Google Maps

The data collection process has filtered the countries of origin to specifically target Low- and Middle-Income Countries (see Figure 1):

- OIDv7: 80% LMICs sourced pictures
- CeyMo: 100% Sri Lanka
- GSV: 100% LMICs
- Mapillary: 100% LMICs
- Google Maps: 100% LMICs

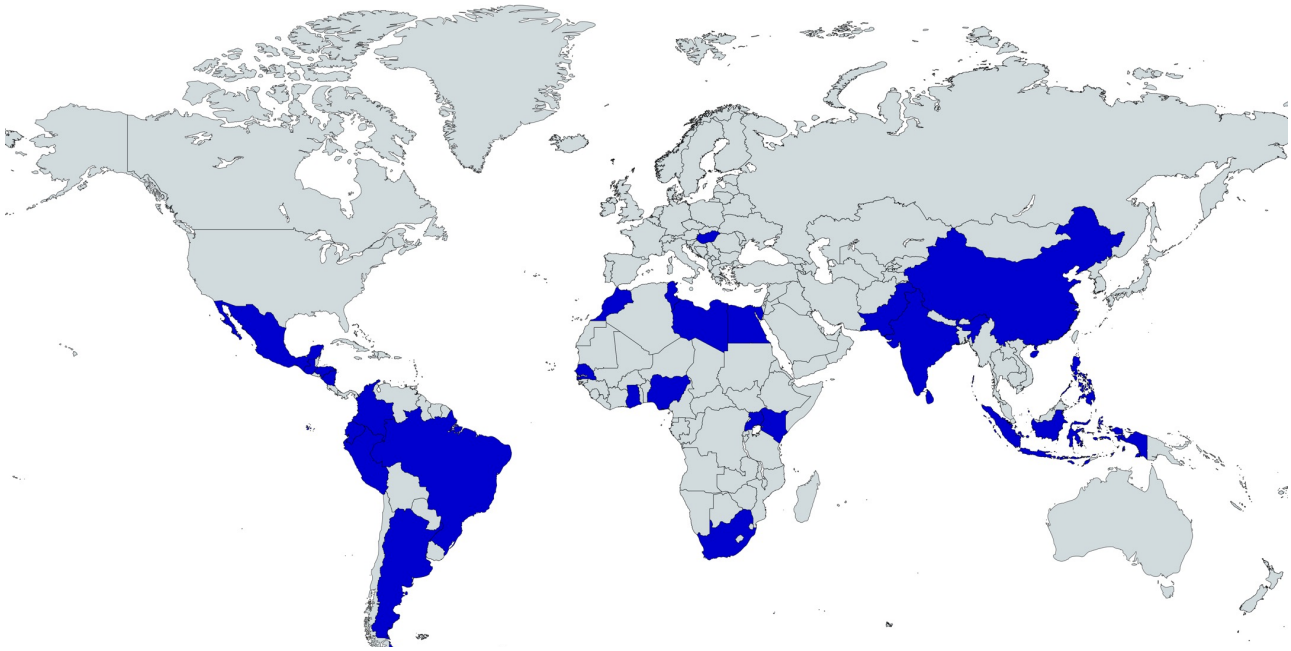


Fig. 1. Map showing the origin of the dataset across Low- and Middle-Income countries

2.2. Data Cleaning and Integration

Amidst almost 30,000 pictures collected, some had to be removed because they were duplicates, and some because they would provide nothing in terms of training the algorithm (no feature to detect) or because they depicted too extreme context (sideway picture of the road, picture in the middle of a railroad, …).

Finally, it was necessary to ensure consistency in terms of coordinate systems and data formats (for labelling in .txt file) and structure (creation of training, testing and validation folders), when merging all the images from different sources to create a unified database. The database is divided into four datasets, one for each model.

3. Multi-Layer Model

Recently, deep learning methods have shown great success in computer vision, like lane detection algorithms based on a CNN[2]. However, one common issue is that one model alone struggles to identify varying different types of shapes, meaning it is hard to extract both lane marking, traffic signs and safety barriers using only one model.

A Multi-Layer Model (MLM) was trained on Ground-level (GL), Aerial view (A), and detailed Lane & Road Marking (LRM) datasets. By training each of the GL object identification, the satellite view object identification, and the LRM identification models, the MLM comprehensively detects and classifies various road and traffic attributes, providing accurate road information to ease both urban planning, traffic management strategies, and infrastructure development in LMICs. Figure 2 provides an illustration of the MLM pipeline.

The differences between the objectives for each model is that the Python library YOLOv8 performs well when detecting boxes, polygons or lines, but not all at the same time.

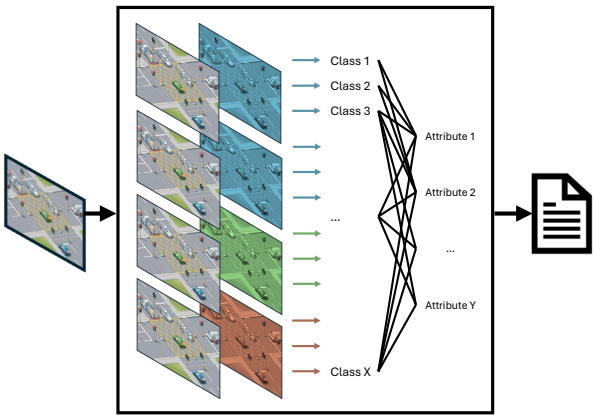


Fig. 2. MLM architecture

3.1. Layer 1: Ground Level

The first layer of the multi-layer model consists of two sub-layers. Each of these two models has been trained on a specific dataset to detect certain classes.

The first model of this GL layer detects four of the road classes: car, tree, traffic light, traffic sign; while the second model detects a further nine road attributes, amongst which include: physical median, central hatching, safety barrier, sidewalk, protected sidewalk, pole, bike lane and pedestrian marked crossings. Model 1 was first to be trained, focusing initially on testing out the capabilities of YOLOv8, using the OIDv7 pre-labelled dataset. Model 2 has been trained and improved three times. The road attributes in these images were manually labelled using Roboflow for labelling images to be trained using a CNN algorithm. However, the lack of precision for detecting bike lanes indicated the need of further training iterations

References

[2] Road markings dataset, 20,000 images: Lee, S., Kim, J. , Shin Yoon, J., Shin, S., Bailo, O., Kim, N., Lee, T.H., Seok Hong, H., Han, S.H., So Kweon, I.: VGPNet: Vanishing point guided network for lane and road marking detection and recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 1947–1955 (2017)

using dedicated datasets to improve this specific class detection performance. The new bike lane dataset allowed the third training iterations of the second model to be able to detect a wider range of bike lanes.

3.2. Layer 2: Aerial Level

The second layer consists of a YOLOv8 model for detecting intersection characteristics, such as roundabouts and grade junctions. All features of this model, including both the dataset and the annotations, were specific to detecting these attributes. The dataset consisted of LMICs satellite images obtained from Google Maps (manually labelled using Roboflow). This model is initiated if and only if the user specifies the need for aerial view detection. The focus shifted once the limitations of the initial GL method became apparent, when it was necessary to detect intersections, prompting the adoption to a new aerial perspective.

3.3. Layer 3: Lane & Road Marking

The third layer consists of a final YOLOv8 model for detecting road markings such as: direction arrows, diamond signals, slow signs, continuous lane marking such as edge lines, yellow markings, bike lanes, pedestrian marked crossings and bus lanes (which is a brand-new attribute, not required yet for the RSSAT).

This model uses data that reverts to a ground-level point of view, allowing the reuse of the same LMICs originated training dataset used for Layer 1.

3.4 Merging the Layers

The Multi-Layer Model can process either images, videos, or both (the videos are discretised as a series of pictures) as an input. To analyse a road, one can then input a series of ground-level pictures, which intersection is disjointed, or ground-level video of the road. Satellite pictures or videos are optional and can be avoided if the classes identified by Layer 3 are not needed. The algorithm processes each picture one after another. Each model is applied one after another on every picture to extract its relative classes. This method allowed for detailed and unique object identification. The results are stored in *grouped_result_detailed* table, where the field headings are: *filename*, *object_id*, *class*, *confidence*, *Index* (a file index), *GPS* (at present this column is left empty, allowing a space for this section to be developed). This structure leaves the possibility for future work to address unique object level treatment.

Nevertheless, road safety attribute assessment requires a different approach. The RSSAT requires specific road features, thus the detailed table is refined to aggregate results per road section. This formatted table summarises findings across distinct categories such as: intersection characteristics, pedestrian crossings, roadside objects, and segment attributes.

Objective evaluation criteria are applied throughout the analysis to interpret feature presence and condition. Criteria include cumulative counts, average confidences, and thresholds derived from the aggregated data to ensure consistent and reliable outcomes.

Interactive elements are integrated throughout the full script to facilitate user inputs, and to ease advancement and outputs savings during script execution.

4. Results

The models respectively took 35, 45, 30, and 50 hours to be trained using the High-Performance Computing (HPC) system at Imperial College London. HPC specifications are: Ubuntu 20.04, 48-core EPYC 7443 CPU, RTX A5000 GPU, 256GB of RAM. *Nota bene*, make sure YOLOv8 performs the calculations on the GPU.

4.1. Layer 1

4.1.1 Model 1

In the first model, 9,000 annotated images were obtained from OIDv7. The use of a pre-labelled dataset provides a time benefit in the progress of the project, eradicating the requirement for a manual labelling stage. This first model was trained for 200 epochs and with a “large” configuration setting of YOLOv8.

Appendix 17 shows the normalised confusion matrix output from the training of the first model. The model was best at predicting cars and traffic lights, with a precision above 72% for both classes. The confusion matrix is a fundamental tool for evaluating the performance of a classification model, providing a comprehensive view of its predictions. Key metrics derived from this matrix include accuracy, precision, sensitivity, and specificity. These metrics are summarised for all models in Appendix 22.

While accuracy measures the overall correctness of the model by calculating the ratio of true positive and true negative predictions to the total number of predictions, precision focuses on the quality of positive predictions, indicating the proportion of true positive predictions out of all positive predictions made by the model. Sensitivity (or recall) assesses the model's ability to identify all relevant instances by calculating the proportion of true positives out of all actual positives. Specificity, on the other hand, evaluates the model's ability to correctly identify negative instances, being the proportion of true negatives out of all actual negatives. Together, these metrics offer a detailed understanding of the model's strengths and weaknesses, aiding in its refinement and improvement.

To go further, Appendix 9, 10, and 5 respectively provide the Precision-confidence, Recall-confidence, and F1-confidence curves. A Precision-confidence curve shows how precision varies with the confidence threshold; as the threshold increases, precision typically increases because the model makes fewer, but more accurate, positive predictions. A Recall-confidence curve depicts how recall changes with the confidence threshold; higher thresholds usually lower recall as fewer positive instances are identified. The F1-confidence curve plots the F1 score against the confidence threshold, indicating the point where the balance between precision and recall is optimal. These curves help in choosing the best threshold to meet specific performance needs, balancing the trade-offs between precision, recall, and the combined F1 score. Regarding Model 1 results, the Precision-confidence, Recall-confidence and the F1-confidence are presenting adequate trends and shapes. Importantly, a high F1 value is seen throughout, indicating high precision. Figures 3 and 4 provide an explicit representation of the physical predictions of the first model. The first image shows a batch of labelled images to indicate what the model should predict correctly, while the second picture depicts the predictions from the model.

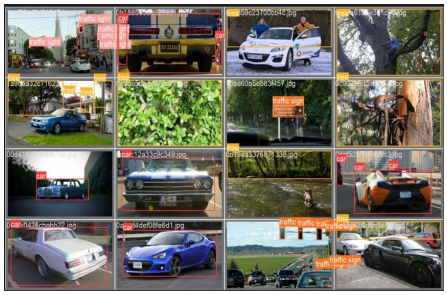


Fig. 3. Model 1 - annotations

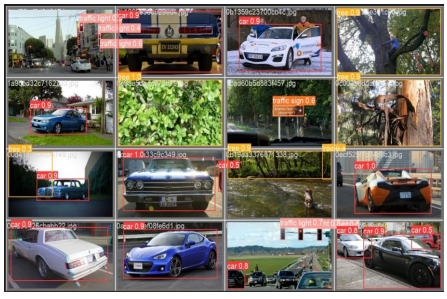


Fig. 4. Model 1 - predictions

Finally, the training performance results seen in Appendix 1 highlight model's training performance. There, the first "train/box_loss", "train/cls_loss", and "train/dfl_loss" graphs indicate that as the number of processed epochs increase, the error of the model decreased. The "val/box_loss", "val/dfl_loss", and "val/cls_loss" respectively display how well the algorithm can locate the centre of an object, and how well the predicted bounding box covers an object, These also show that the error of detecting objects with various scales and aspect ratios decrease, and that it is stable to indicate the correct classification of the attribute. Additionally, the "metrics/…" graphs show the accuracy of the detected objects along with the ability of the model to identify all instances of objects in the images.

Fortunately, all training and validation curves are indeed of a decreasing trend while all other metrics curves are increasing. This results can be found for the other models in Appendix 2, 3, and 4.

4.1.2. Model 2

This second model has been trained three times using YOLOv8 with "large" configuration and for 200 epochs. This because the first iteration yielded poor results related to bike lane detection (only 17% true positive).

The initial dataset consisted in +11,000 images from 17 LMICs downloaded from Mapillary. To improve this, another +1,000 new bike lane images from 7 LMICs (mainly in Asia and Latin America) were added to the initial database and were trained into the second iteration.

Even though the second iteration improved the bike lane detection, mostly every other attribute decreased in accuracy. This prompted the run of a third iteration. This time the training parameters were changed, which influenced the randomisation of the training batches. Specifically, this involved altering the value of the seed. A seed is the initial value used by a random number generator to start the generation of a sequence of pseudo-random numbers. This value ensures that the sequence of numbers can be replicated exactly if the same seed is used again, providing reproducibility and consistency in experiments and simulations. Seeds are used to control random processes, and by utilising this feature, the reliability of the experimental results was maintained.

In Appendix 21, a comparison between the three iterations can be seen. The chosen model was the third iteration, as it is the best performing model. The improvement is clear against the second iteration as the proportion of true positives for bike lanes increased from 0.57 to 0.70, proving the usefulness of the change in the parameters.

It is important to note that the "pro_bikelane" attribute was dismissed, as there were not enough images (that can highlight the high variety between regions) gathered, since there is generally a lack of bike lane infrastructure in LMICs.

Appendix 18 shows the normalised confusion matrix of the third iteration.

Note that "val/seg_loss" curve increases after a big drop, however, this metrics is not crucial because this project put the emphasis on detection rather than precisely cropping each item. Figure 5 and 6 show an input picture without annotations, and an image with the predicted labelling from the model 2. The model is able here to detect poles, sidewalk, and crossing.

Fig. 5. and Fig. 6. Comparison between input picture and detection of the Layer 1_2

4.2. Layer 2

The dataset used in the training of the model consisted of +1,000 unique satellite images obtained from Google Maps in 21 LMICs, manually labelled using Roboflow, which were flipped horizontally and vertically, and the brightness was adjusted in a duplicate set of images, to increase the dataset to nearly 4,000 unique labelled images. Model 3 was trained using YOLOv8 algorithm with the "large" training configuration, again for 200 epochs.

In Appendix 19, the normalised confusion matrix highlights that the model detects 53% of true positive grade junctions and 77% of true positives for roundabouts. In Appendix 7, the F1-confidence curve reaches a peak of 0.54 at 29% of confidence. The Precision-confidence curve in Appendix 13, shows the difference in accuracy of the detected objects, while in Appendix 14, the Recall-confidence curve shows the ability of the model to identify all instances of objects in the images.

Note that while all other metrics and training curves are of satisfying shapes and trends, the "val/box_loss" and "val/dfl_loss" graphs are increasing over the number of epochs. This phenomenon might be due to either overfitting or a lack of diversity in the contrast setting of the training pictures.

Figures 7 and 8 show an input picture without annotations, and an image with the predicted labelling from model 3.

Fig. 7. and Fig. 8. Comparison between input picture and detection of the Layer 2

4.3. Layer 3

The third layer was designed to detect mostly road lines and signalling. For this, the CeyMo dataset has been re-employed, again consisting of more than +2,000 unique pictures. Again, model 4 was trained using YOLOv8 algorithm in a "large" configuration, for 200 epochs.

The confusion matrix for this model is shown in Appendix 12, and it is evident from this that the model has produced highly accurate results. The obvious bold diagonal in this matrix indicates that each attribute is mostly predicted correctly. On top of that, all training and validation curves are indeed of a sharply decreasing trend while all other metrics curves are rapidly increasing. These results can be found in Appendix 4. Finally, Appendix 8, 15, and 16 finish to underline the fourth model's relevancy and precision.

Figures 9 and 10 show an input picture without annotations, and an image with the predicted labelling from model 4.

Fig. 9. and Fig. 10. Comparison between input picture and detection of the Layer 3

5. Requirements

5.1. Technical Requirements

The requirements listed below are intended to ensure optimal performance for both training a YOLOv8 model and running detection tasks. Training a YOLOv8 model is computationally intensive and requires robust hardware and software specifications while running detection tasks typically requires fewer resources than training, unless deployed in real-time.

For software, YOLOv8 supports multiple operating systems, including Windows (≥ 10, 64-bit), Ubuntu (≥ 18.04), and macOS (≥ 10.15). Python (≥ 3.7) is necessary as the primary programming language, with essential

libraries and frameworks such as: PyTorch (≥ 1.7.0), CUDA Toolkit (≥ 10.2), compatible cuDNN, OpenCV (≥ 4.0), NumPy, SciPy, Matplotlib, Pillow, and optionally TensorRT for optimized inference on NVIDIA GPUs.

On the hardware front, YOLOv8 necessitates a robust configuration. The CPU should ideally be an Intel i5 (8th Gen or higher) or AMD Ryzen 5 (3000 series or higher), featuring a minimum of 4 cores and 8 threads. For accelerated performance, an NVIDIA GPU with CUDA capability 6.0 or higher is recommended, with the NVIDIA RTX 2060 or higher being optimal (≥ VRAM 8GB). To support these components, a minimum of 16 GB RAM is recommended.

Training machine learning models requires powerful GPUs for intensive computations and substantial CPU power and RAM for managing large datasets and complex calculations. In contrast, detection can run on less powerful GPUs and requires less CPU and RAM, though it still benefits from GPU acceleration and sufficient system resources.

To summarise the technical requirements, adhering to these specifications ensures that YOLOv8 operates efficiently on a personal system, leveraging both software capabilities and hardware performance for optimal results in object detection and related tasks.

Also note that a recommended 100 GB SSD might be needed to reckon with the training database and models' weights storage.

5.2. Input Requirements

The MLM relies on YOLOv8 which can process both pictures or videos. When initiating the models, the MLM algorithm asks for the need (or not) for aerial view detection. If this is needed, only Layer 1 and 3 will be initiated and road information can be input as either a series of pictures (above Standard Definition (SD) 640x360 pixels (360p), preferably High Definition (HD 1280x720 pixels (720p)) with null intersection, or a single shot video (same resolution requirements, frame rate of 30 fps).

In the case one wants to add satellite images as an input, the MLM will initiate all layers to treat the two types of input (ground and aerial level). The resolution requirements for the aerial point of view are the same as for ground level, either it is a series of pictures or a video. However, the scale is third parameter to consider here. When web scrapping the data via Google Maps, a zoom level 15 or higher will roughly correspond to a scale of 1:20,000. This setting allows for viewing small towns to road level precision, which is exactly what Layer 2 is designed for.

6. Limitations and Future Work

Despite the promising results achieved by the Multi-Layer Model for road and traffic attribute detection, some limitations can be addressed to enhance its effectiveness and generalisability.

Firstly, the MLM's performance is closely tied to the quality and diversity of the datasets. Additionally, the current model struggles with detecting and classifying objects when presented with cluttered or occluded scenes, which are common in real-world scenarios. These limitations underscores the need for more sophisticated approaches that can handle complex and overlapping annotations more effectively.

Future work will focus on addressing these limitations through key areas of improvement. Enhancing the annotation quality and consistency of the specific methodology used across datasets is paramount, potentially with semi-automated or crowdsourced annotation methods to ensure a more uniform and comprehensive training dataset. Expanding the dataset to include more varied and challenging scenarios will also be crucial for improving the model's robustness. Finally, on the model front,

incorporating techniques such as data augmentation, transfer learning, or synthetic data generation can help to mitigate the limitations posed by diverse data sources and enhance the model's adaptability to different environments.

6.1. Limitations

Although YOLOv8 has demonstrated strong performance in various object detection tasks, as shown previously, some limitations must be considered for optimal deployment in complex scenarios. Firstly, YOLOv8 tends to struggle in accurately detecting small objects within large images, because its grid division strategy leads to small objects being easily missed or inaccurately detected. Secondly, YOLOv8's performance decreases with overlapping or densely distributed objects, as it only predicts one object per grid, causing multiple overlapping objects to be recognized as a single object.

Regarding data sources, acquiring data from platforms like Google Maps, Mapillary, or CeyMo presents significant challenges when collating data from many of the low- and middle-income countries. These platforms often have data coverage that is biased towards developed countries and major cities, resulting in sparse data for developing and remote regions. Additionally, certain attributes listed by sources like the World Bank, such as protected bike lanes, are difficult to find in LMICs. These countries often have less developed infrastructure, and the relevant data collection and recording are not as comprehensive.

Despite efforts to collect relevant coordinates, a standardized method to differentiate between urban and rural areas is lacking. Ultimately, while on the whole, the database compiles miscellaneous weather conditions, limited variations of each location were gathered. As a result, data for the same location across different conditions is insufficient, thereby restricting the understanding of how environmental changes impact transportation and infrastructure.

6.2 Future Research Directions

It is noted that the current algorithms used are existing algorithms sourced from open online libraries and adapting them to integrate the new data sources. To achieve the desired outcomes, set by the World Bank, the focus was directed to creating a comprehensive list of attributes and improving the accuracy of the object detection algorithm. With regard to the originality of the algorithm, the multi-layer model highlights that leveraging pre-existing algorithms fits above and beyond the necessary requirements for this object detection model.

Currently, the annotation process relies on manual operations. However, to improve efficiency and accuracy, the plan would be to implement auto-labelling technology. This transition will not only significantly reduce the manual workload but also greatly enhance data processing speed and consistency. Auto-labelling will leverage advanced algorithms and machine learning techniques to ensure that annotation tasks are completed efficiently and accurately, thereby improving the overall workflow efficiency and quality. To address the identified limitations and enhance the data collection and analysis, the proposal is to establish partnerships with local governments, NGOs, and research institutions in developing countries to access more localized data and collaborating with international organizations for broader data collection.

Additionally, diversifying the data collection methods by incorporating crowd-sourcing, satellite imagery, and mobile data collection apps, and promoting the use of open data platforms will ensure a wider range of data types and sources, thereby improving the

robustness and comprehensiveness of our dataset. For example, similar to the road markings attribute, a point of note for the continuation of the research in future would be to shift the focus onto ensuring that the multi-layer model would be able to detect bike lanes in model 2 with less obvious features and with somewhat faded bike lane markings. Again, this is particularly relevant for road attributes in low-and middle-income countries.

Google Maps APIs offer the possibility to obtain satellite images given specific location coordinates. Simultaneously, Google Street View assists in accessing 360° panoramic views and images of the very same locations. For future improvements, one might want to focus on depth estimation[3] of the different objects detected on ground-level images in order to determine their precise GPS coordinates. Obtaining images from an API based on the coordinates is a straightforward task, thus allowing a seamless match of aerial and ground-level data. Another way to improve this MLM could be to rely solely on satellite image datasets, which typically include files indicating coordinates. With this GPS data, new developers can integrate GPS treatment in merging part of the MLM which has purposefully been design for GPS data integration.

6.3. Novelty of the Research

Mapillary computer vision technology is potentially the leader in object detection for street images. As of 2023, they declare a list of 65 classes of objects they can detect. Considering this list, it can be noted that some key attributes that the RSSAT model is able to detect are not present in Mapillary, such as: central hatching, physical median and safety barrier for median type; the difference between a sidewalk and a protected sidewalk which is only indicated as “pedestrian area”; and it does not have an aerial view, meaning that it cannot detect the roundabouts neither grade junctions, though from street view it is able to detect “bridges” which might be the most similar attribute to grade junctions.

As reported in [15], the YOLOv8 algorithm has the best mean average precision of traffic signs detection compared to previous YOLO versions using the same dataset. This shows that this research is utilizing state of the art algorithm.

Another of the innovative aspects of this research is the source of the images in the datasets, coming from low- and medium-income countries (following the World Bank classification), as other street object detection studies have focused on only developed countries or regions such as in Europe and Japan [4], or more precisely in Europe [5] (Germany [6], and Greece [7]); or in both developed and undeveloped countries, such as [8] and [10] on a global view, [9] in India, Japan and Czech, or China and United States [11]. Only a few focus specifically on developing countries like Colombia [12] or Turkey [13]. When focusing on research that only uses YOLO algorithm, [16] summarize the state of the art for road traffic sign object detection and classifies them based on their dataset geography.

Out of the 161 studies published between 2016 and 2022, 67 (41.6%) focus on only one specific developed country, 57 (35.4%) focus on only one low- or medium-income country, 35 (21.7%) focus somewhere in the world without explicitly mentioning the location, and only 1 (0.6%) was based around the whole world.

References

[3] Koch, T., Liebel, L., Fraundorfer, F. and Körner, M., 2018. Evaluation of CNN-based Single-Image Depth Estimation Methods. arXiv preprint arXiv:1805.01328.

Lastly, this study is more comprehensive in terms of the types of road marking that can be detected, as Layer 3 of this research is also able to detect diamond signs and slow signals.

7. Conclusion

The objective of this project was to leverage pre-existing AI algorithms to detect road features in low- & medium-income countries. The modular approach of the Multi-Layer Model, which integrates specifically trained Convolutional Neural Networks, effectively mitigates the weaknesses of some models by utilising the strengths of others.

In order to increase model expansion opportunities for the World Bank, the modularity of the Multi-Layer Model approach allows the final algorithm to be enriched by adding more relevant layers or analysis tools. One might be interested in generating a Digital Twin of the road after thoroughly tracking unique objects' GPS coordinates via depth estimation methods.

Furthermore, some attributes are still missing due to a lack of quantifiability. For example, one attribute that may seem trivial is the number of lanes. In some countries, because of the lack of road lane marking, people will drive side by side despite a lack of a safe distance between cars. Likewise, pedestrians often present a challenge when it comes to labelling. Detecting only humans is not enough to train the model for this class as people are often driving cars, riding bikes, etc, causing confusing in the detection of pedestrians.

No matter what methods are employed to assess such attributes, there remain some interpretation issues regarding their proper definition, which would be a paramount first stage to develop a detection algorithm further.

8. Acknowledgements

The authors would like to thank Professor Mohammed Quddus, Dr. Felix Feng, and Shaofan Sheng of Imperial College London for the assistance and support they were able to provide throughout the course of this project upon which this document is based.

Further thanks to Messrs. Said Dahdah, Dipan Bose, Kazuyuki Neki, and Diego Canales from the World Bank for this learning opportunity, along with the informative and constructive feedback that was provided throughout the duration of the project.

8. References

[1] Liu, X., Deng, Z., Lu, H., and Cao, L., 2017. Benchmark for road marking detection: Dataset specification and performance baseline. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6. IEEE.

[2] Lee, S., Kim, J., Shin Yoon, J., Shin, S., Bailo, O., Kim, N., Lee, T.H., Seok Hong, H., Han, S.H., and So Kweon, I., 2017. VGPNet: Vanishing point guided network for lane and road marking detection and recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1947–1955.

[3] Koch, T., Liebel, L., Fraundorfer, F. and Körner, M., 2018. Evaluation of CNN-based Single-Image Depth Estimation Methods. arXiv preprint arXiv:1805.01328.

[4] Yang, Z., Zhao, C., Maeda, H. and Sekimoto, Y. (2022). Development of a Large-Scale Roadside Facility Detection Model Based on the Mapillary Dataset. *Sensors*, 22(24), p.9992. doi:https://doi.org/10.3390/s22249992.

[5] Braun, M., Krebs, S., Flohr, F. and Gavrila, D.M. (2019). The EuroCity Persons Dataset: A Novel Benchmark for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, [online] 41(8), pp.1844–1861. doi:https://doi.org/10.1109/TPAMI.2019.2897684.

[6] Voth, R.S. (2024). *Real-time detection of German street signs*. [online] reposit.haw-hamburg.de. Available at: https://reposit.haw-hamburg.de/handle/20.500.12738/14527 [Accessed 21 Jun. 2024].

[7] Moschos, S., Charitidis, P., Doropoulos, S., Avramis, A. and Vologiannidis, S. (2023). StreetScouting dataset: A Street-Level Image dataset for finetuning and applying custom object detectors for urban feature detection. *Data in Brief*, p.109042. doi:https://doi.org/10.1016/j.dib.2023.109042

[8] Neuhold, G., Ollmann, T., Rota Bulo, S. and Kontschieder, P. (2017). *The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes*. [online] openaccess.thecvf.com. Available at: https://openaccess.thecvf.com/content_iccv_2017/html/Neuhold_The_Mapillary_Vistas_ICCV_2017_paper.html [Accessed 21 Jun. 2024].

[9] Arya, D., Maeda, H., Ghosh, S.K., Toshniwal, D., Mraz, A., Kashiyama, T. and Sekimoto, Y. (2021). Deep learning-based road damage detection and classification for multiple countries. Automation in Construction, 132, p.103935. doi:https://doi.org/10.1016/j.autcon.2021.103935.

[10] Chen, Y., Chen, L. and Li, Y. (2021). Recognition algorithm of street landscape in cold cities with high difference features based on improved neural network. *Ecological Informatics*, 66, p.101395. doi:https://doi.org/10.1016/j.ecoinf.2021.101395.

[11] Luo, Z., Gao, L., Xiang, H. and Li, J. (2023). Road object detection for HD map: Full-element survey, analysis and perspectives. ISPRS Journal of Photogrammetry and Remote Sensing, 197, pp.122–144. doi:https://doi.org/10.1016/j.isprsjprs.2023.01.009.

[12] Escallón Páez, F. (2023). BEDPI-DL: Built Environment objects Detection on Panoramic Images of Bogota with Deep Learning models. *repositorio.uniandes.edu.co*. [online] Available at: https://repositorio.uniandes.edu.co/entities/publication/0393c193-1eec-4197-aec7-4d1a75c1d104 [Accessed 21 Jun. 2024].

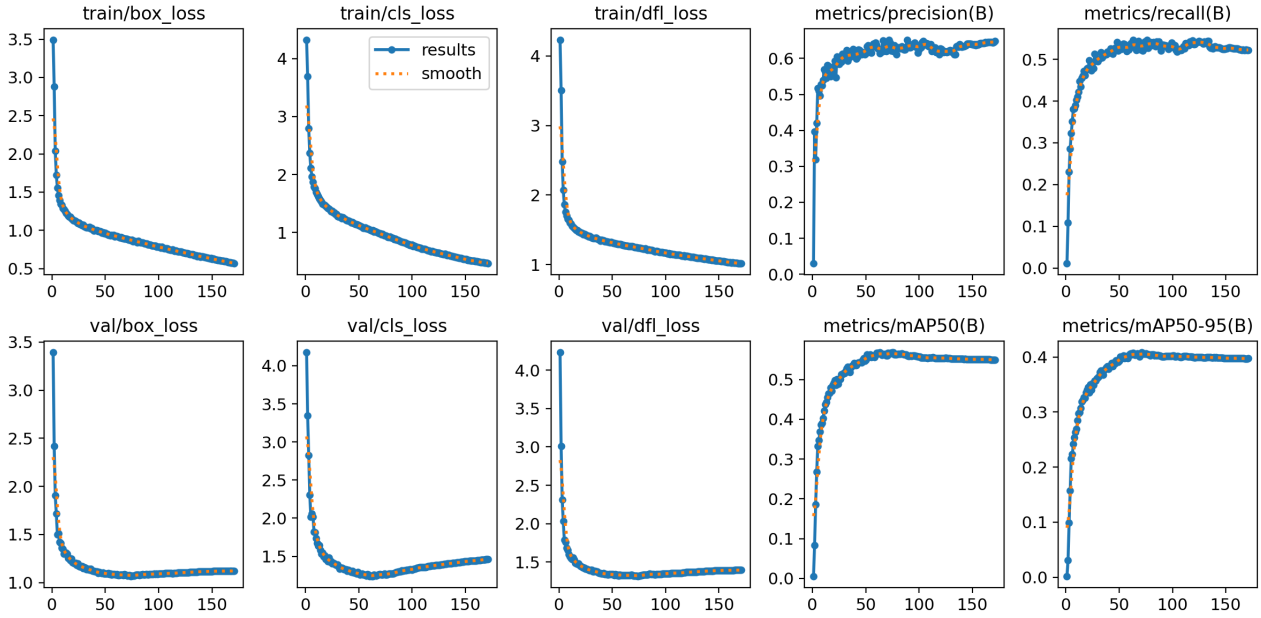
[13] Güney, E. and Bayılmış, C. (2021). An Implementation of Traffic Signs and Road Objects Detection Using Faster R-CNN. *SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES*, [online] 5(2). doi:https://doi.org/10.35377/saucis.05.02.1073355

[14] Chen, T., Dai, J., Dong, B., Zhang, T., Xu, W. and Wang, Z. (2024). Road Marking Defect Detection Based on CFG_SI_YOLO Network. Digital signal processing, pp.104614–104614. doi:https://doi.org/10.1016/j.dsp.2024.104614

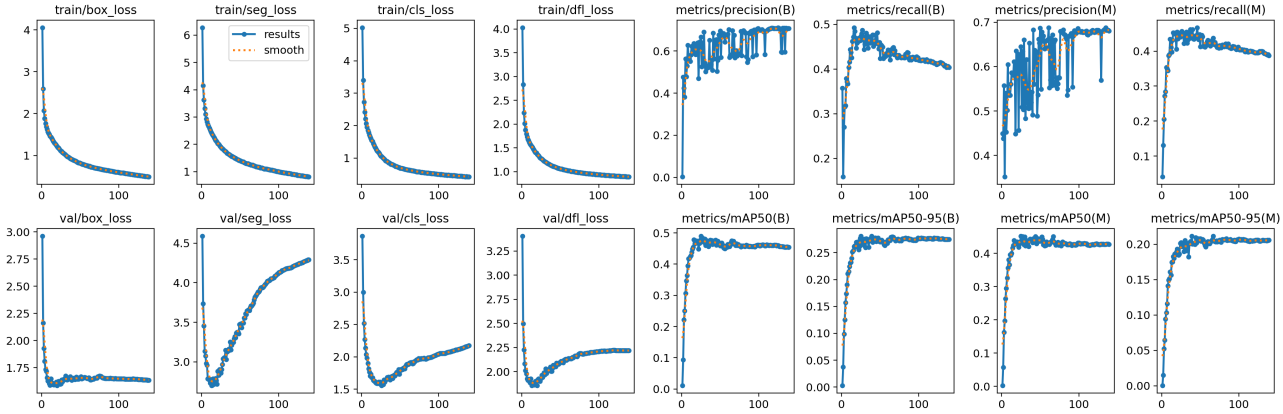
[15] Patel, P., Vipul Vekariya, Shah, J. and Vala, B. (2024). Detection of traffic sign based on YOLOv8. AIP conference proceedings. doi:https://doi.org/10.1063/5.0212701.

[16] Flores-Calero, M., Astudillo, C.A., Guevara, D., Maza, J., Lita, B.S., Defaz, B., Ante, J.S., Zabala-Blanco, D. and Armingol Moreno, J.M. (2024). Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review. Mathematics, [online] 12(2), p.297. doi:https://doi.org/10.3390/math12020297.

Appendices

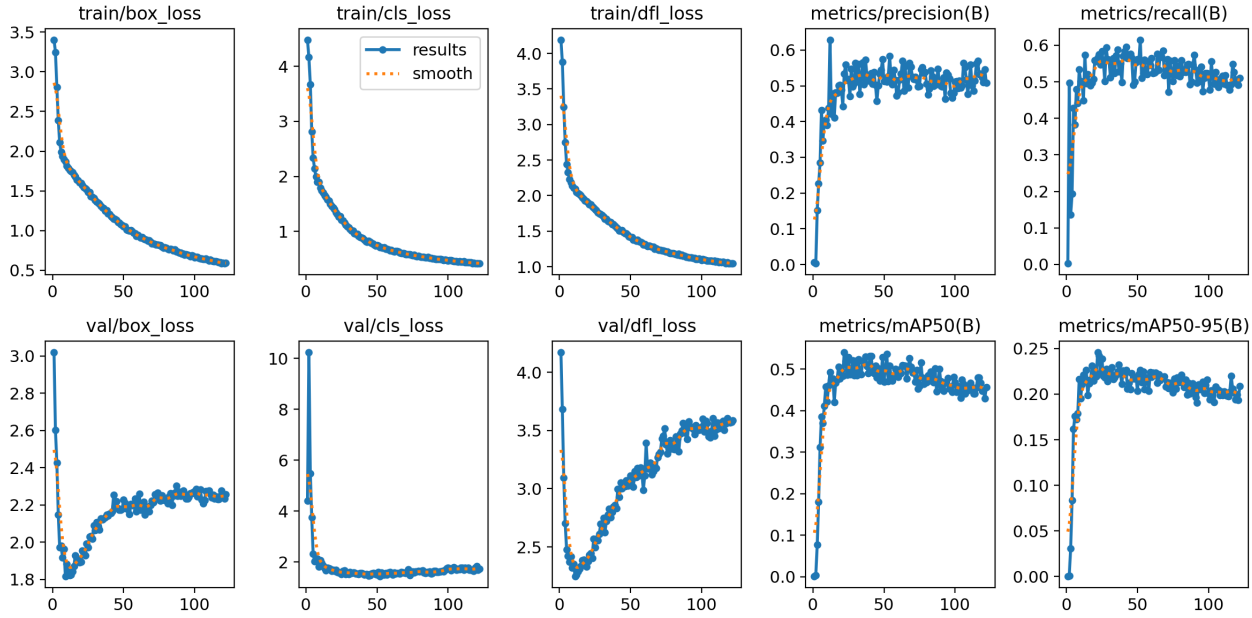


Appendix 1. Performance metrics for Layer 1_1

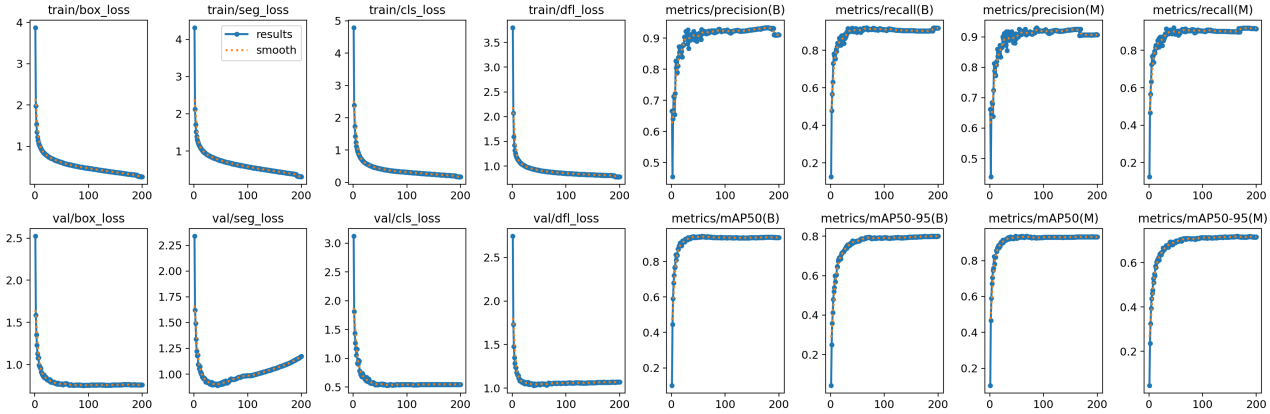


Appendix 2. Performance metrics for Layer 1_2

Appendices

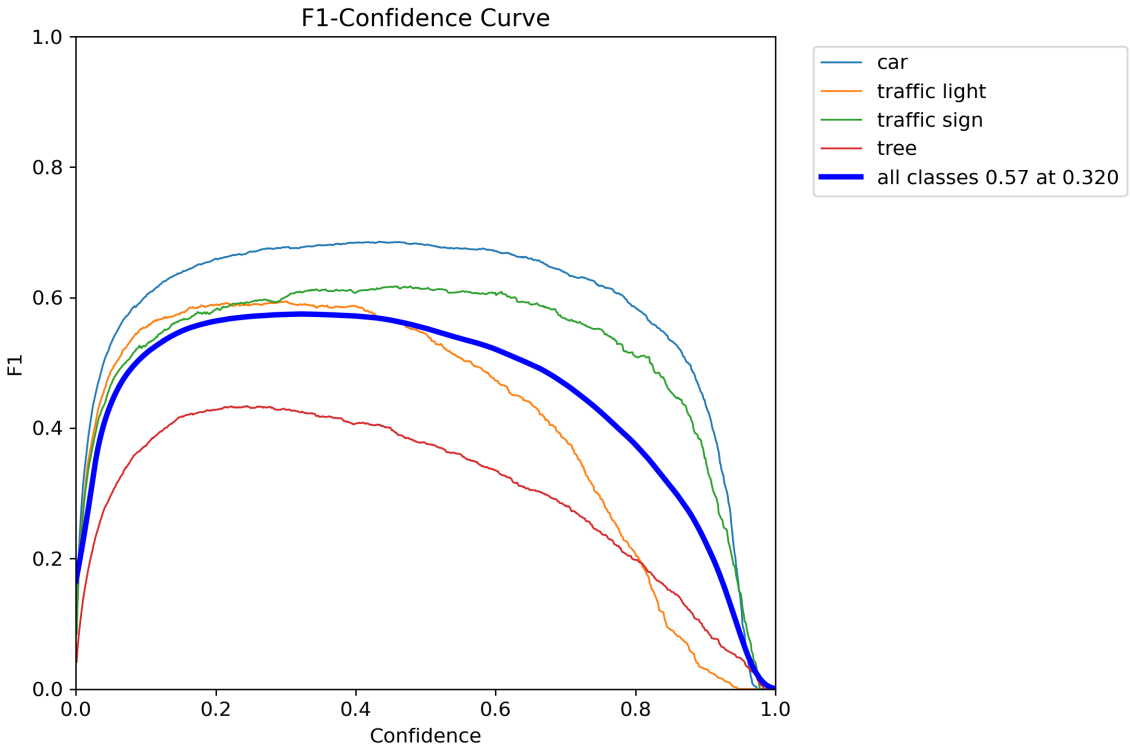


Appendix 3. Performance metrics for Layer 2

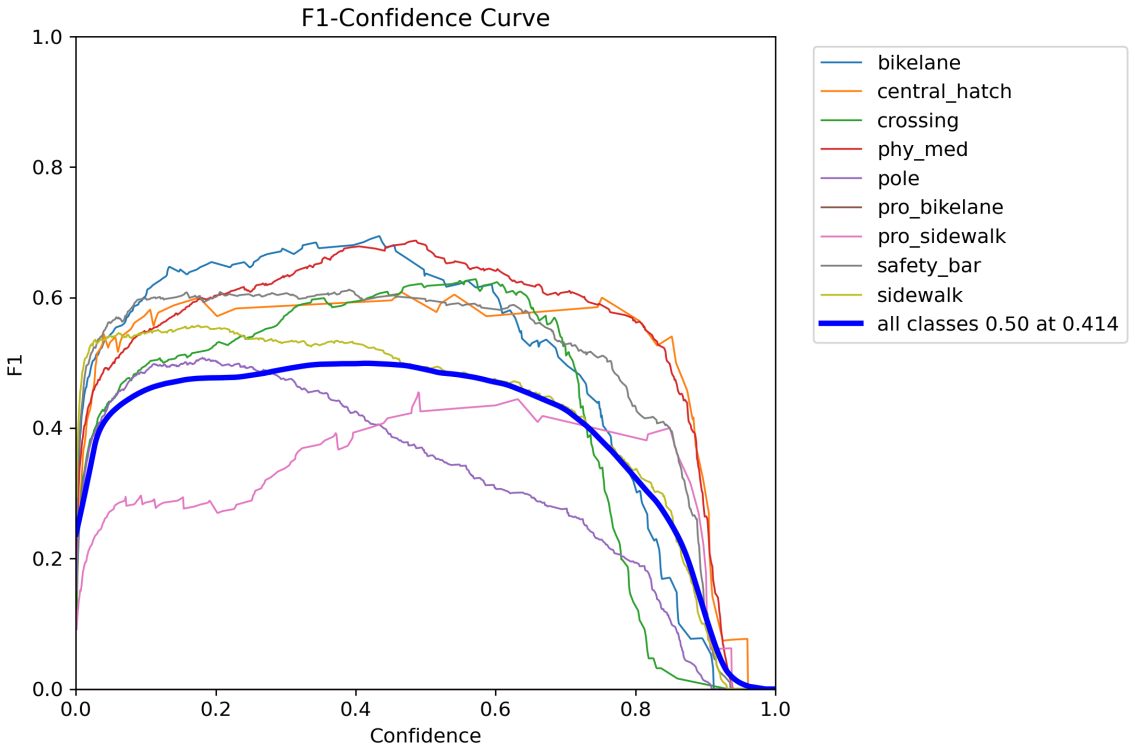


Appendix 4. Performance metrics for Layer 3

Appendices

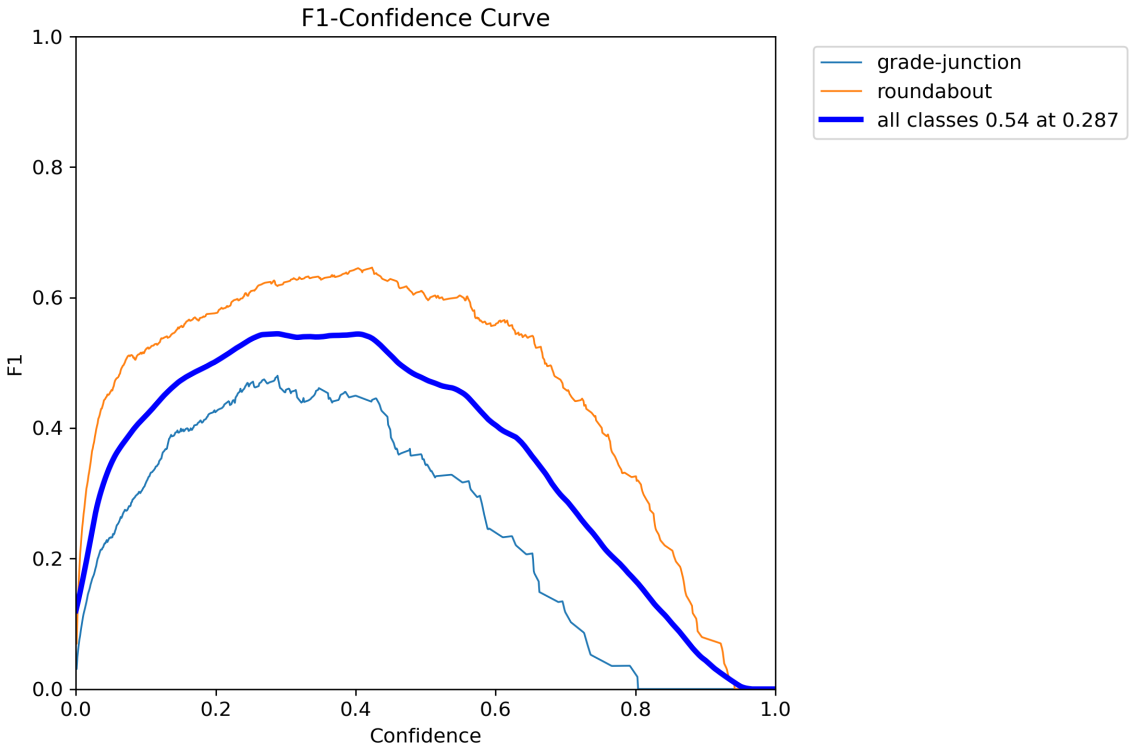


Appendix 5. F1 Confidence curve for Layer 1_1

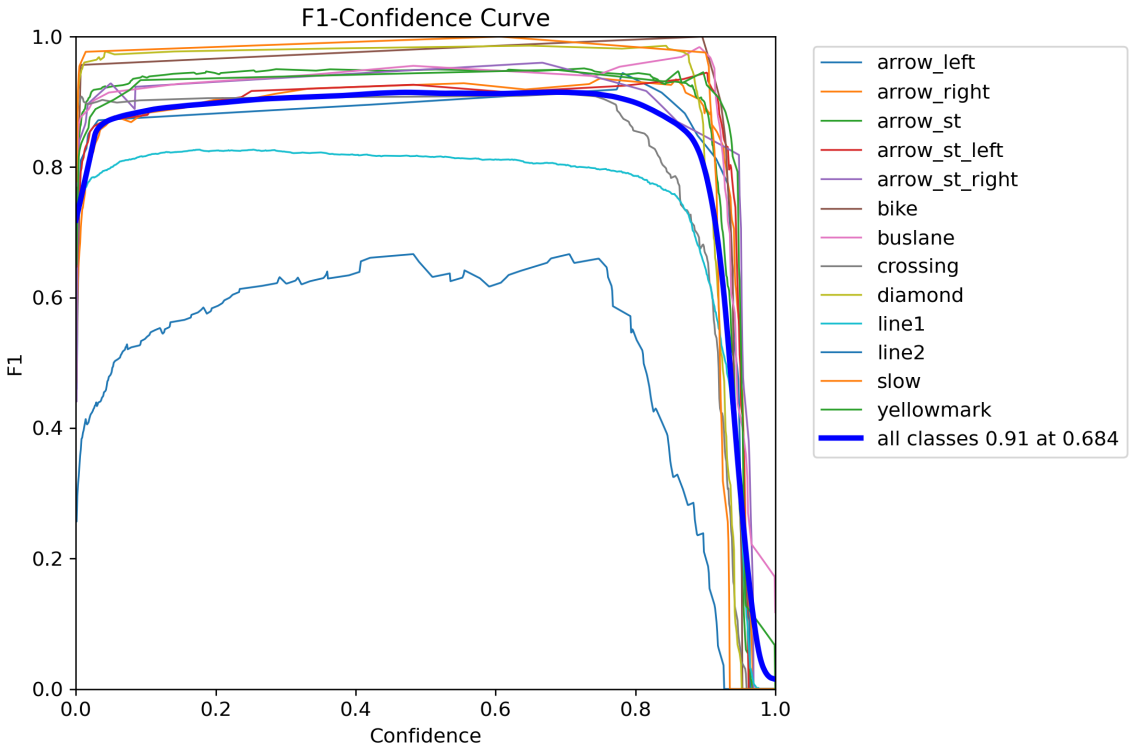


Appendix 6. F1 Confidence curve for Layer 1_2

Appendices

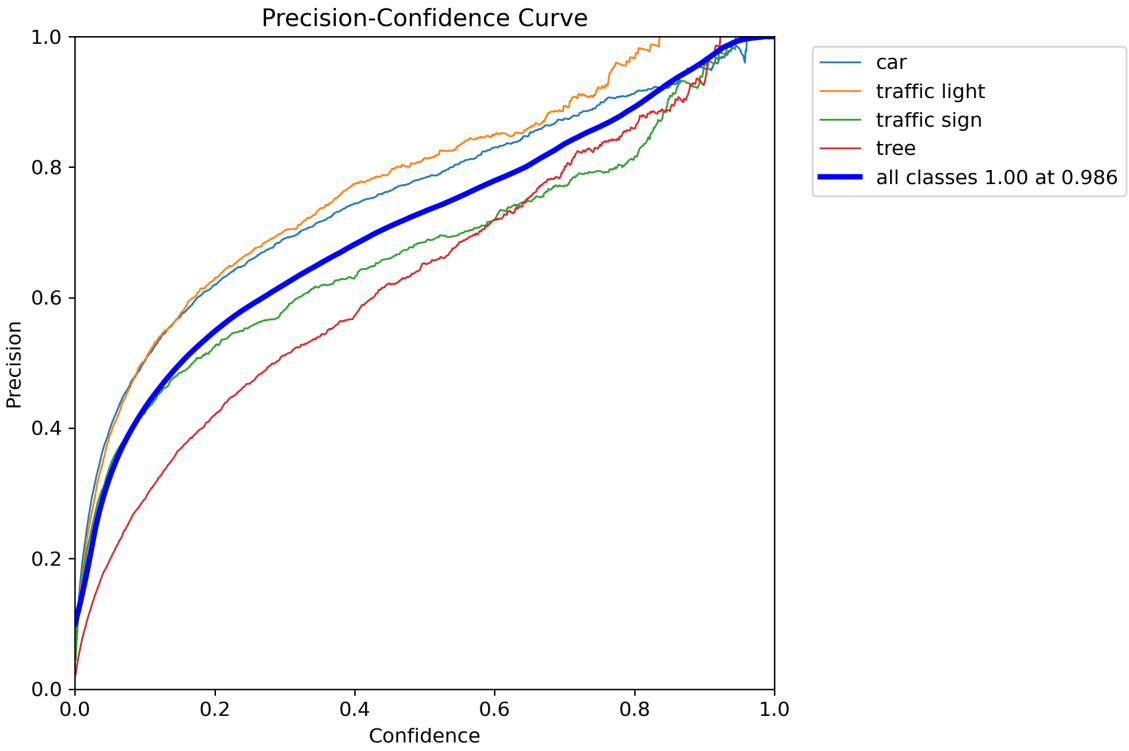


Appendix 7. F1 Confidence curve for Layer 2

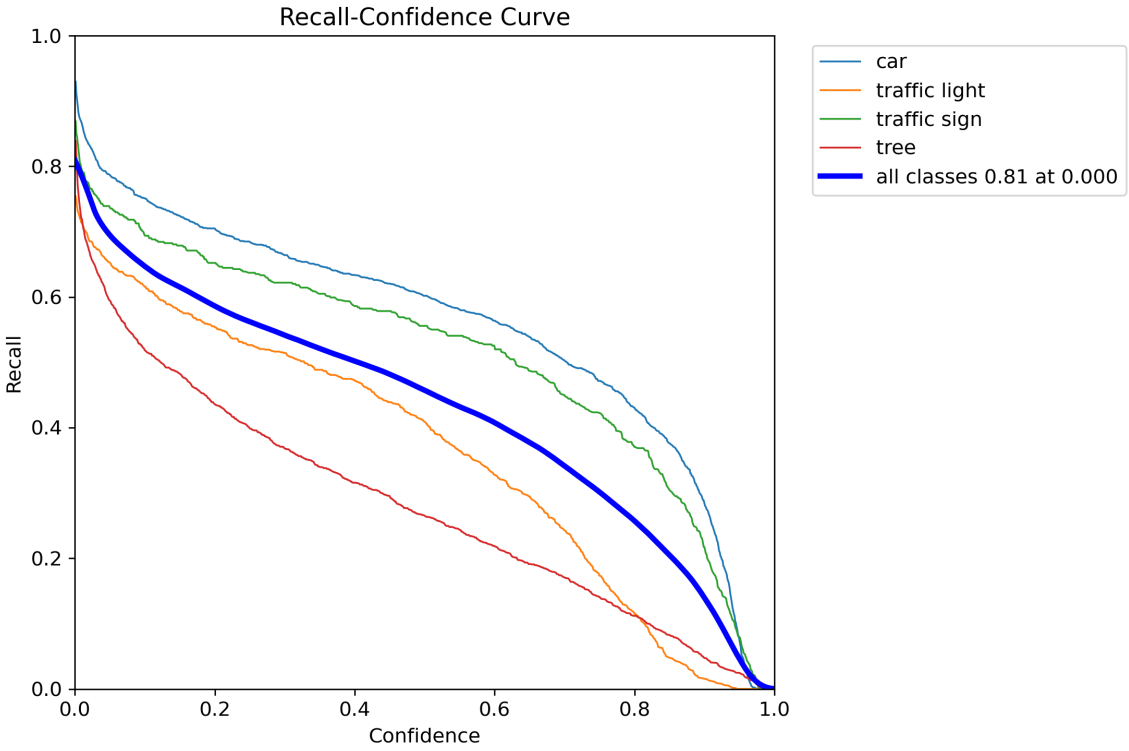


Appendix 8. F1 Confidence curve for Layer 3

Appendices

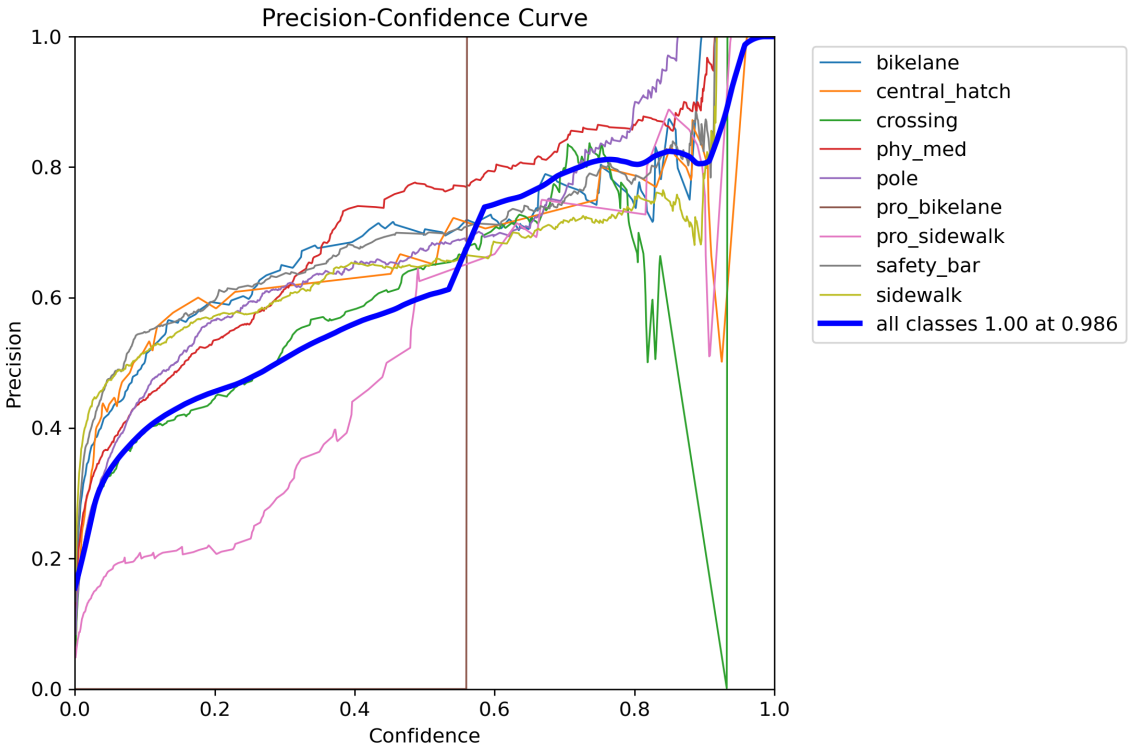


Appendix 9. Precision – Confidence curve of Layer 1_1

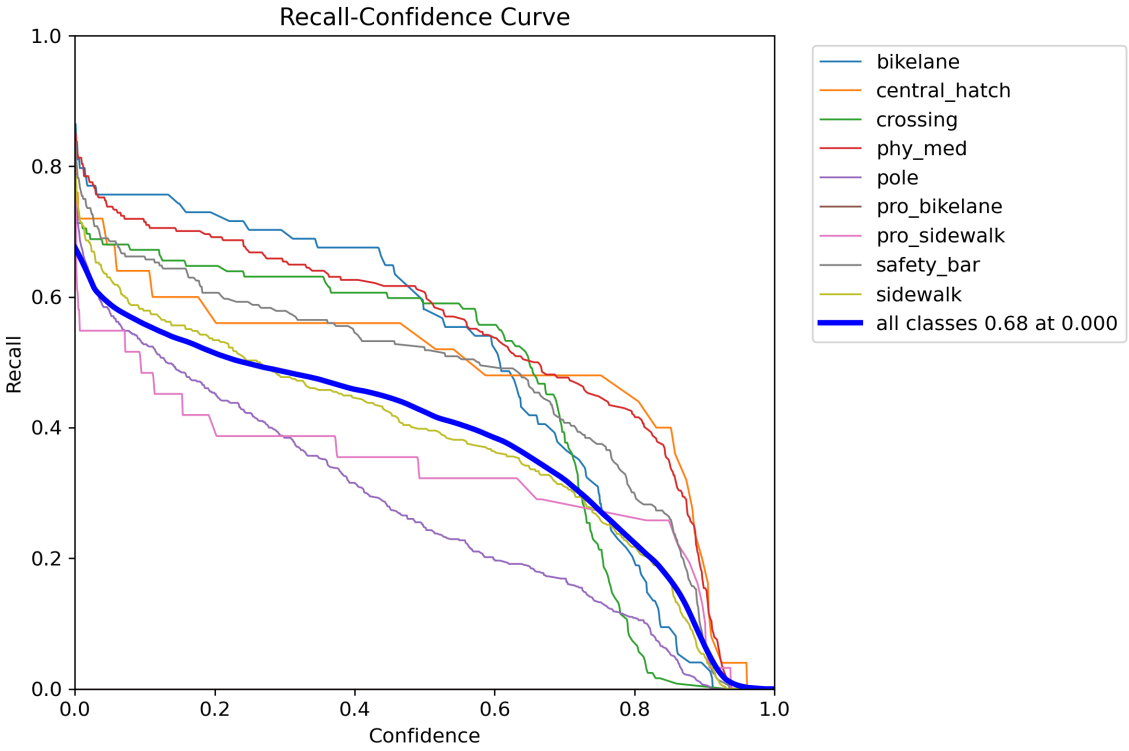


Appendix 10. Recall – Confidence curve of Layer 1_1

Appendices

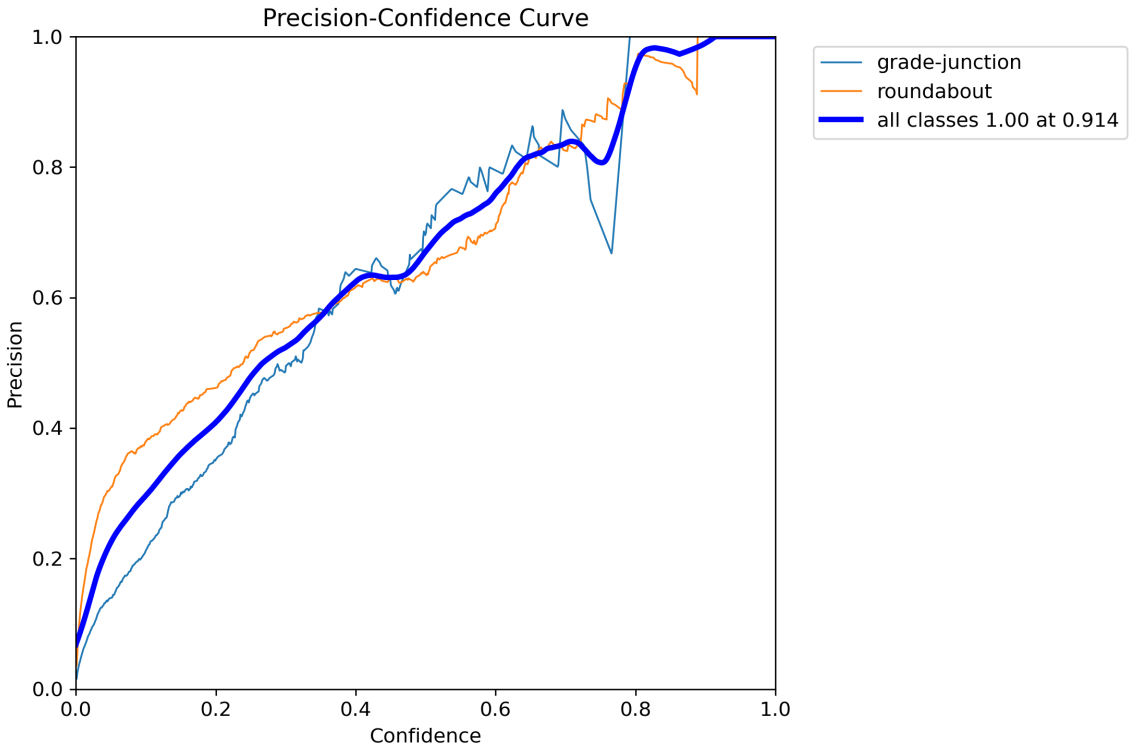


Appendix 11. Precision – Confidence curve of Layer 1_2

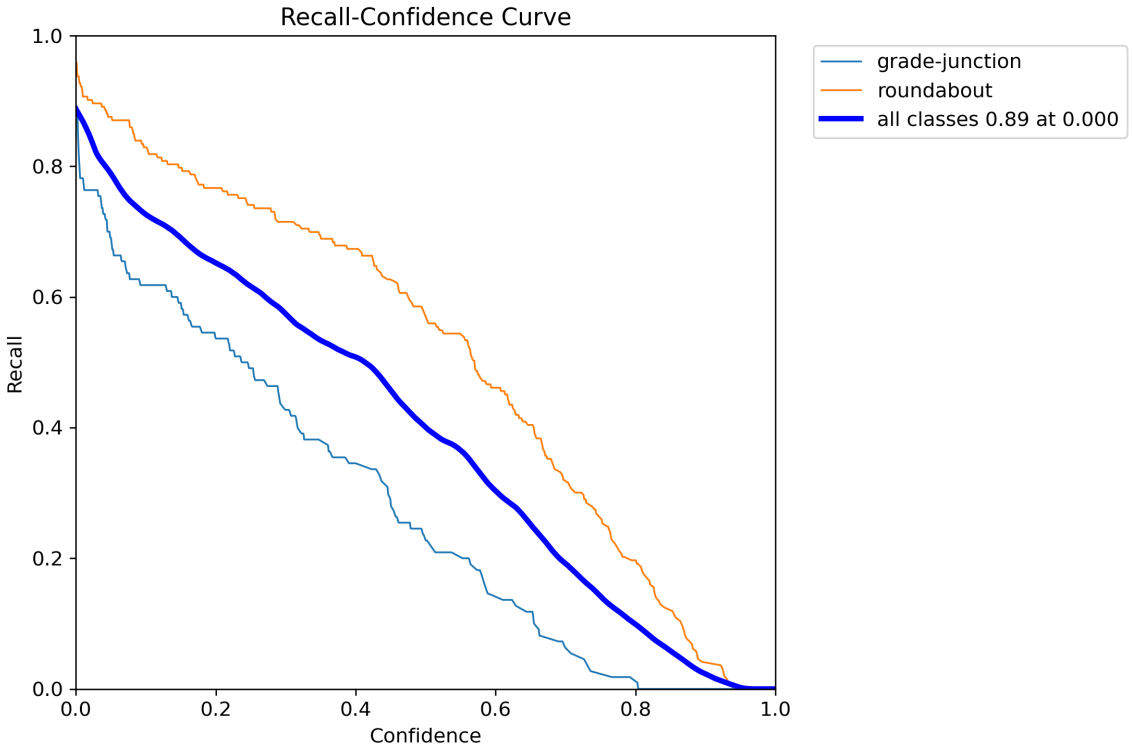


Appendix 12. Recall – Confidence curve of Layer 1_2

Appendices

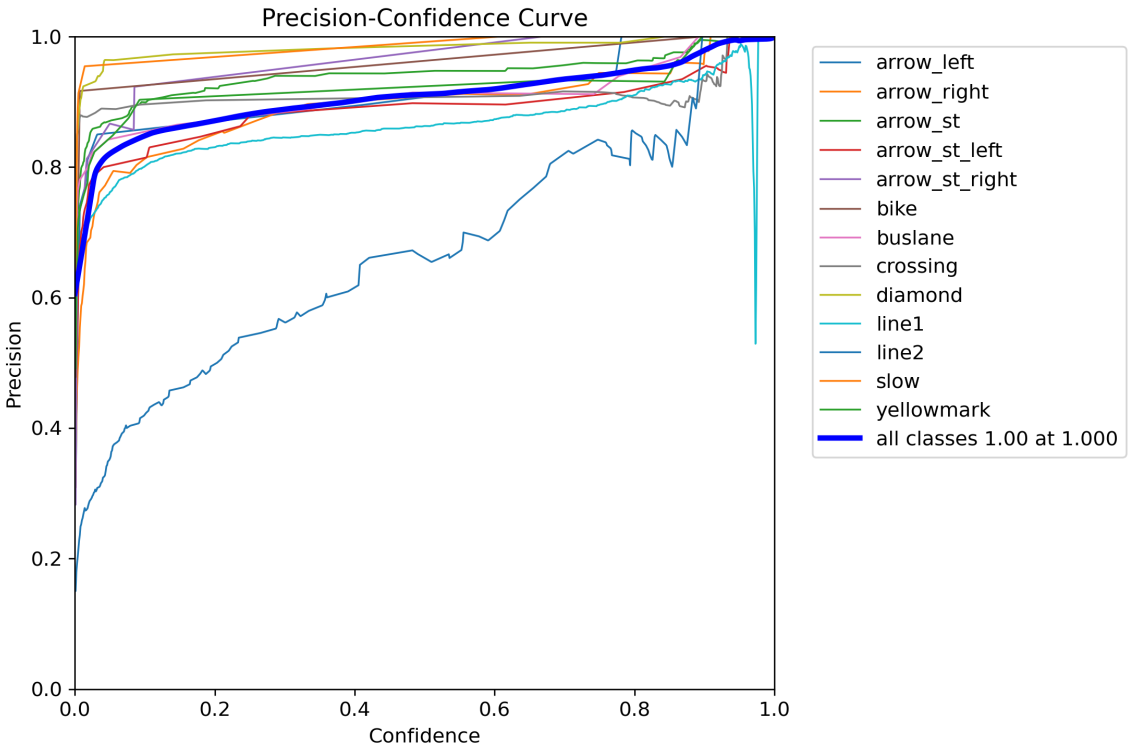


Appendix 13. Precision – Confidence curve of Layer 2

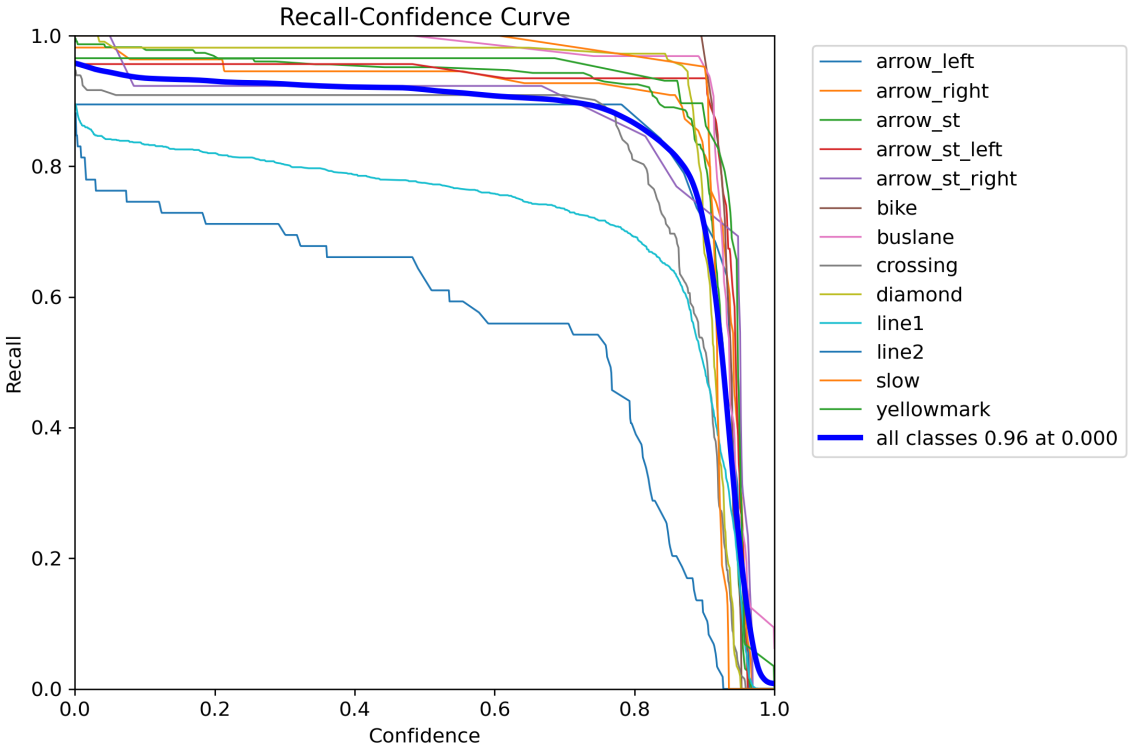


Appendix 14. Recall – Confidence curve of Layer 2

Appendices

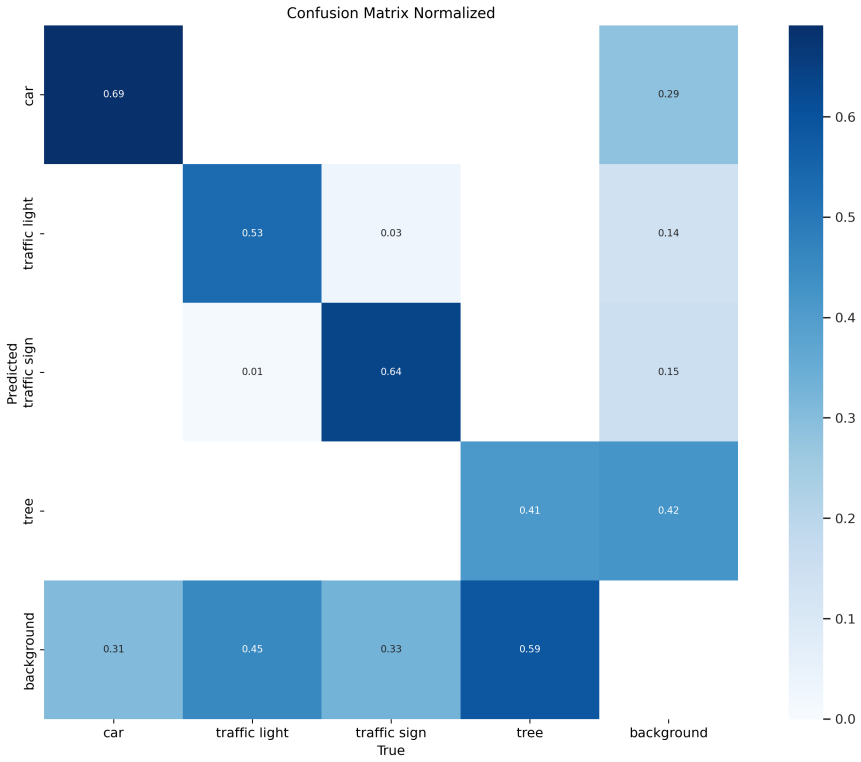


Appendix 15. Precision – Confidence curve of Layer 3

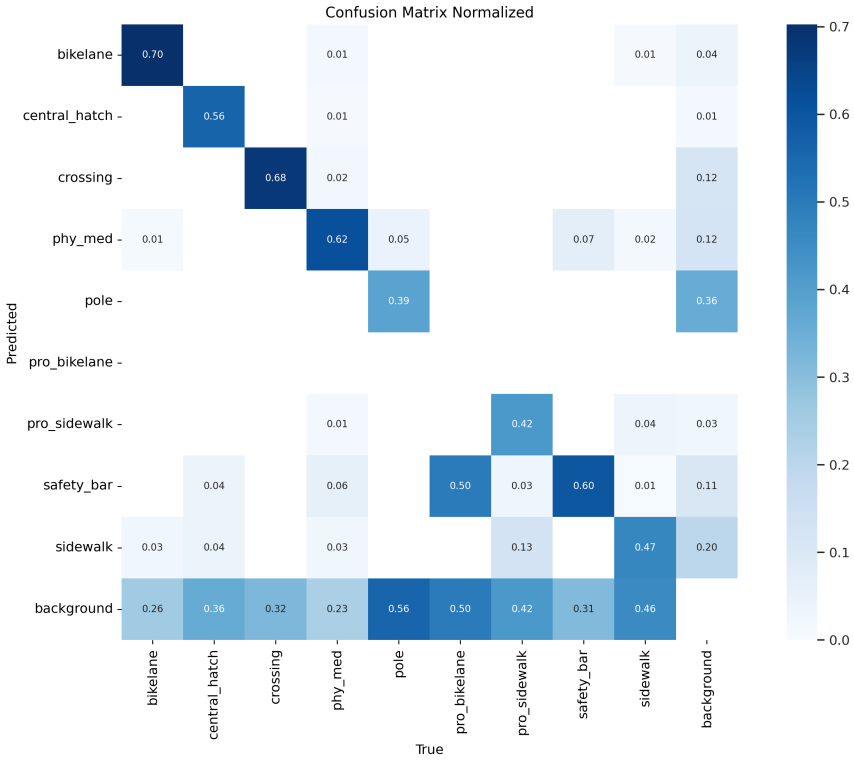


Appendix 16. Recall – Confidence curve of Layer 3

Appendices

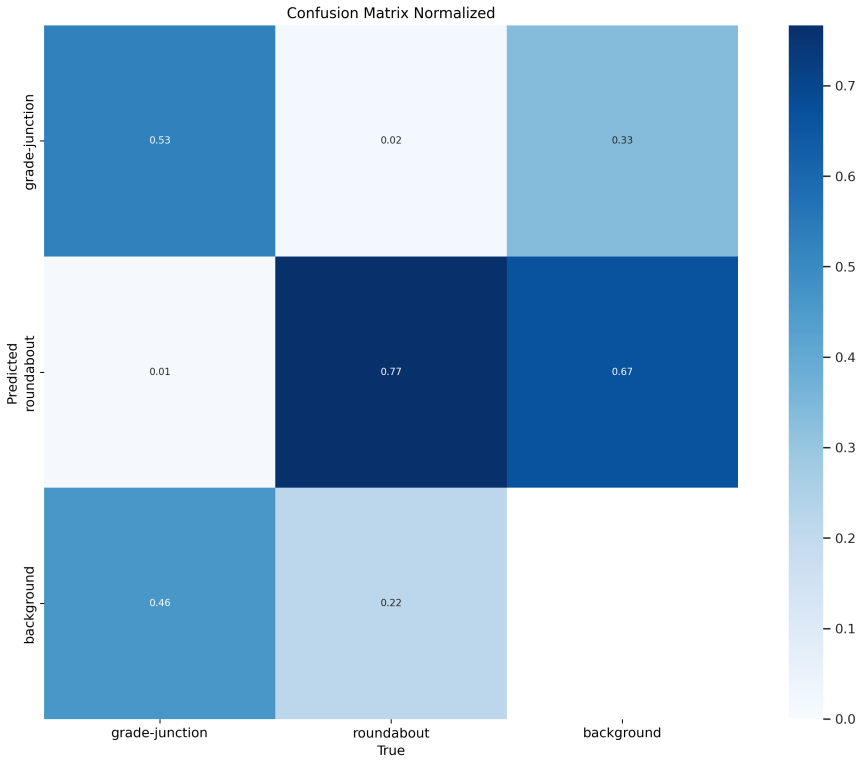


Appendix 17. Normalised confusion matrix of Layer 1_1

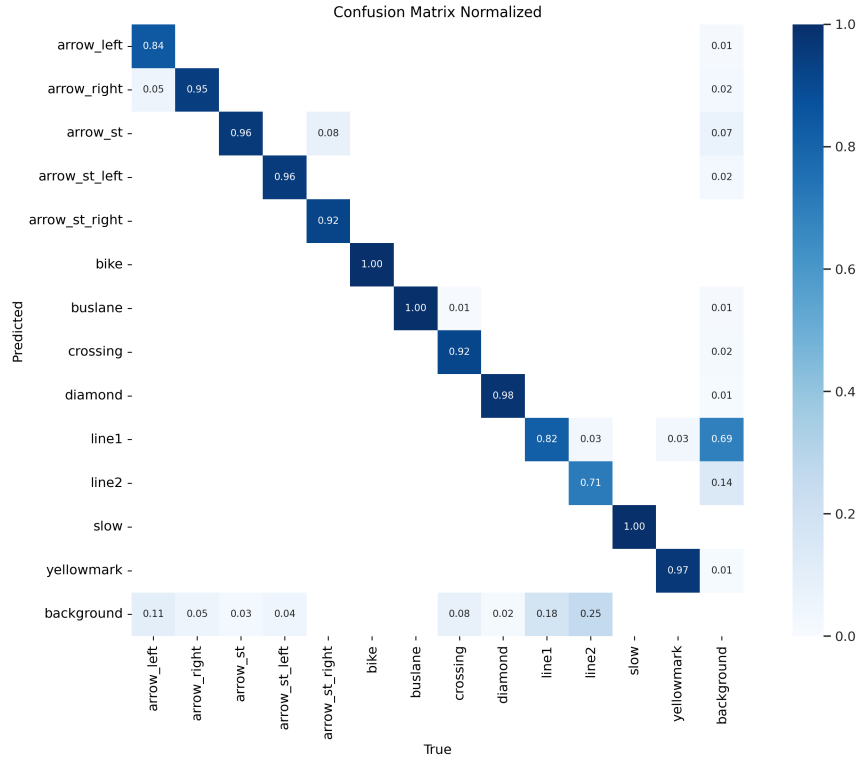


Appendix 18. Normalised confusion matrix of Layer 1_2

Appendices



Appendix 19. Normalised confusion matrix of Layer 2



Appendix 20. Normalised confusion matrix of Layer 3

Appendices

Class	Iteration 1	Iteration 2	Iteration 3	Change in TP% Between 2nd & 3rd
Bike lane	0.17	0.57	0.70	+0.13
Central hatch	0.60	0.36	0.56	+0.20
Crossing	0.70	0.54	0.68	+0.12
Physical medium	0.62	0.47	0.62	+0.15
Pole	0.42	0.23	0.39	+0.16
Protected sidewalk	0.61	0.23	0.42	+0.19
Sidewalk	0.50	0.30	0.47	+0.17

Appendix 21. True positive (TP) proportion of each class for every iteration of model 2

Appendices

		Sensitivity	Specificity	NPV	Precision	Occurrences
Layer 1 Model 1	car	0.6911	0.9182	0.9055	0.7237	1444
	traffic light	0.5333	0.9603	0.9123	0.7267	1007
	traffic sign	0.6389	0.9616	0.9656	0.6123	529
	tree	0.4125	0.8705	0.7799	0.5712	1799
Layer 1 Model 2	bikelane	0.7027	0.9869	0.9889	0.6667	74
	central_hatch	0.5600	0.9961	0.9946	0.6364	25
	crossing	0.6803	0.9684	0.9796	0.5764	122
	phy_med	0.6168	0.9434	0.9549	0.5593	214
	pole	0.3879	0.8941	0.7877	0.5906	580
	pro_bikelane	0.0000	0.9985	0.9990	0.0000	2
	pro_sidewalk	0.4194	0.9862	0.9911	0.3171	31
	safety_bar	0.5972	0.9646	0.9532	0.6649	216
	sidewalk	0.4718	0.9400	0.8952	0.6208	354
Layer 2 Model 3	grade-junction	0.5273	0.8746	0.8323	0.6105	110
	roundabout	0.7668	0.6745	0.7606	0.6820	193
Layer 3 Model 4	arrow_left	0.8333	0.9986	0.9986	0.8333	18
	arrow_right	0.9455	0.9968	0.9986	0.8814	55
	arrow_st	0.9605	0.9914	0.9954	0.9280	228
	arrow_st_left	0.9565	0.9972	0.9991	0.8800	46
	arrow_st_right	0.9231	1.0000	0.9995	1.0000	13
	bike	1.0000	1.0000	1.0000	1.0000	11
	buslane	1.0000	0.9982	1.0000	0.8889	32
	crossing	0.9167	0.9981	0.9947	0.9680	132
	diamond	0.9817	0.9990	0.9990	0.9817	109
	line1	0.8177	0.8257	0.7885	0.8506	1212
	line2	0.7119	0.9837	0.9920	0.5455	59
	slow	1.0000	1.0000	1.0000	1.0000	21
	yellowmark	0.9655	0.9991	0.9995	0.9333	29

Appendix 22. Detection metrics for model benchmarking