
2023

ALAN NUNES CAETANO

# CYCLIST ROUTE ASSESSMENT USING MACHINE LEARNING

Faculdade de Design, Tecnologia e Comunicação
Universidade Europeia

**2023**

**ALAN NUNES CAETANO**

# CYCLIST ROUTE ASSESSMENT USING MACHINE LEARNING

Dissertação apresentada ao IADE - Faculdade de Design, Tecnologia e Comunicação da Universidade Europeia, para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Computação Criativa e Inteligência Artificial realizada sob a orientação científica do Doutor Edirlei Soares de Lima, Professor Auxiliar da Universidade Europeia – IADE e do Doutor Jacinto Estima, Professor Auxiliar do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

First, I wish to thank my supervisors Prof. Edirlei Lima and Prof. Jacinto Estima for their guidance, support and for showing a big interest in the work I developed. I also appreciate all the love, patience and encouragement from my friends and colleagues.

**palavras-chave** avaliação de segurança do ciclista; visão computacional; deep learning.

**resumo** Aumentar o número de deslocamentos de bicicleta pode trazer inúmeros benefícios para indivíduos e comunidades. No entanto, vários fatores, incluindo a disponibilidade de ciclovias, características do tráfego e qualidade do pavimento, podem encorajar ou desencorajar o uso de bicicletas. Para promover o ciclismo e entender como os ciclistas interagem com o ambiente urbano, é crucial avaliar a qualidade das rotas dos ciclistas. Esta tese propõe uma ferramenta de avaliação automática que usa aprendizado de máquina para detectar características do segmento de rota e calcula uma pontuação que representa o nível de segurança e conforto para os ciclistas. Os modelos são treinados no YOLOv5 para classificar os tipos de pavimento, detectar defeitos no pavimento e detectar a presença de ciclovias. Dois datasets foram construídos e anotados para as tarefas de classificação do tipo de pavimento e detecção de infraestrutura ciclável. Foi aplicado um questionário aos ciclistas para comparar as percepções reais com a avaliação automática. Os resultados mostraram um bom alinhamento com as percepções reais, validando a abordagem, mas também demonstraram a necessidade de adicionar novas características e melhorar a performance dos modelos antes de ser adequado para uso real.

**keywords** cyclist safety assessment; computer vision; deep learning.

**abstract** Increasing the number of bike commutes can provide numerous benefits for individuals and communities. However, several factors including the availability of cycle paths, traffic characteristics, and pavement quality, can either encourage or discourage the use of bicycles. To promote cycling and understand how cyclists interact with the urban environment, it is crucial to assess the quality of cyclist routes. This thesis proposes an automatic assessment tool that uses machine learning to detect features of the route segment and calculates a score representing the level of safety and comfort for cyclists. The models are trained on YOLOv5 to classify pavement types, detect pavement defects and detect the presence of cycle paths. Two datasets were built and annotated for the pavement type classification and cycle infrastructure detection tasks. A questionnaire was applied to cyclists to compare the real perceptions with the automatic assessment. The results showed a good alignment with the real perceptions, validating the approach, but also demonstrated the need of adding new features and improving the models' performance before being adequate for real use.

# Contents

# List of Tables

# List of Figures

# 1 Introduction

Choosing a bicycle or other active transportation methods for commuting instead of a car brings many benefits for individuals and the community (Rabl & De Nazelle, 2012; Pucher, Dill, & Handy, 2010). One of the biggest impacts on the individual that uses a bicycle for the daily commute is the health improvement (Rabl & De Nazelle, 2012), with lower rates of obesity, cardiovascular issues and morbidity (Pucher et al., 2010; Bassett, Pucher, Buehler, Thompson, & Crouter, 2008). In Bassett et al. (2008) the authors examined the relationship between obesity and the used commute method, showing that countries with higher use of active transportation methods had fewer obesity rates and countries with higher automobile dependence had highest obesity rates. Even for the perception that a cyclist is more exposed to air pollution, for example, there is evidence that physical activity has a higher positive impact than the effect of air pollution (Cavill, Kahlmeier, Rutter, Racioppi, & Oja, 2008).

In addition to that, the benefits to the community are not negligible. With fewer cars in traffic, there is a reduction in air pollution, congestion, noise pollution and greenhouse gas emissions (Rabl & De Nazelle, 2012). Increasing the number of drivers switching to bicycle commuting also means a reduction in the rates of bicycling injuries due to more visibility in traffic (Pucher et al., 2010; Cavill et al., 2008).

Although the majority of these benefits for individuals, communities and the environment are common sense to the population, actual transportation infrastructures in cities are mostly made for motorized traffic, making it more convenient to travel with personal automobiles. Because of this, it is important to understand what is more influential in the route choice when using active methods of transportation (Stinson & Bhat, 2003).

The literature presents several factors that can influence route choice and the likeability to use the bicycle as a transportation mode. Safety perception is one of the main factors and relates to the presence of infrastructure that separates the cyclist from the motorized traffic. When there is no separation, other factors may also influence, such as road width, presence of motorized vehicles parking, motorized vehicle weight, speed and general traffic safety (Arellana, Saltarín, Larrañaga, González, & Henao, 2020; Winters & Teschke, 2010).

Another important factor is the comfort of the route and the level of effort to commute.

Looking at types of pavement, paved surfaces are the most preferred over unpaved paths, being them off-street or not, and rural roads are the least preferable (Winters & Teschke, 2010). Adding to that, routes with detours, slopes, and other obstacles that extend the ride tend to increase the commuter effort and are seen as less desirable routes (Arellana et al., 2020).

In addition to studying which factors impact the decision of commuting by bicycle and the weight of each one of these factors, it is important to know how to map them and evaluate routes. There are several studies exploring different approaches to use these findings to calculate indexes, such as the bikeability index, to provide users with easy to understand information.

The majority of studies do not reflect automated approaches. Manton, Rau, Fahy, Sheahan, and Clifford (2016) uses mental mapping combined with a stated-preference survey to compare the levels of safety perception pointed on the routes with the answers about infrastructure preference and safety questions and the data present in a transport infrastructure inventory of Galway City (Ireland). Winters and Teschke (2010) made a population-based survey in the Metro Vancouver region of Canada using pictures of routes asking about the preferred route types and the ones they actually use. A different method of not automated risk perception was made in Catania (Italy) by Cafiso, Pappalardo, and Stamatiadis (2021), it combines the object risk, captured using monitoring equipment (GPS, cameras and GNSS-video system) data analyzed by experts, with the perceived risk, from a web-survey to score the same components. These studies were able to provide very interesting insights about the factors that influence the bikeability perception, although the assessment and identification of the traffic objects on the images were done manually, limiting the application to a large scale. Cafiso et al. (2021) recommends further investigation on how to use artificial intelligence (AI) for the automatic detection of events.

On the other hand, there are many studies exploring techniques on how to use AI to detect route features, such as pavement type, pavement quality and presence of cycling infrastructure. Several studies discuss techniques for creating image-based models to detect cracks in concrete infrastructure using computer vision to automatize inspections (Gupta & Dixit, 2022). This can be applied to detect cracks and other defects on the asphalt as it is demonstrated in L. Zhao, Wu, Luo, and Yuan (2022), where the authors

trained a model based on Faster RCNN. Another example of route feature detection using computer vision is presented in Nolte, Kister, and Maurer (2018), where the author creates a camera-based surface detection model using Deep Convolutional Neural Networks to determine road-tire friction coefficients for motorized vehicles. This could be used to detect types of pavement that influence the comfort of cyclists.

Adapting these computer vision techniques to detect route features related to bicycle commuting and combining available road network datasets could result in a tool to map the safest and most comfortable routes for cyclists, which could be used to encourage the use of bicycles and aid infrastructure improvement decisions.

The main goal of this research is to propose a new approach to automatically assess the safer and more comfortable route segments for cyclists using a parameterized weighted system, computer vision and road network datasets. There is also an objective of using insights from state-of-the-art (SOTA) papers to build useful models. However, there is no intention to reproduce or even outperform the SOTA models, since they typically explore specific techniques and architectures, which was not the focus of this thesis.

This document is organized as follows.

**Chapter 2** contextualizes the main technical concepts that were used on the development of this work.

**Chapter 3** is a literature review that explores how the authors solve the individual problems involved in this work's approach: the measurement of cycling safety and comfort through the urban environment analysis, the routes features identification using computer vision and the integration of these techniques in a pipeline to assess the routes.

**Chapter 4** describes the steps planned to achieve the research goals of this thesis, as the necessary data and tools to execute these steps.

**Chapter 5** presents a comparison between the perceptions from the questionnaire and the assessment tool results.

**Chapter 6** discusses the contributions of this work and lists the limitations and future work opportunities.

# 2 Background

This research will take advantage of computer vision and deep learning algorithms to support automatic feature detection. This chapter contextualizes the technical concepts used in this thesis.

## 2.1 Computer Vision

Computer vision can be defined as the use of statistical methods, geometry and computational models to extract information from digital images and emulate a human performing visual tasks. There is a wide range of areas where computer vision is applied, some examples are autonomous navigation, military intelligence, industrial inspection, medical image analysis and character recognition (Forsyth & Ponce, 2011; Khan, Laghari, & Awan, 2021). There are several algorithms, techniques and tools to implement computer vision systems, here it will be described the ones that were used in this work.

OpenCV is a library of infrastructure operations, utility functions for image processing and computer vision algorithms implementations. It is open source, written in C and was created by Intel with the objective of assisting programmers and data scientists in building vision applications (Bradski & Kaehler, 2008; Parker, 2010).

## 2.2 Machine learning and Deep learning

Machine learning (ML) is a subfield of AI where the machine learns from a dataset without being explicitly programmed for and is able to take certain decisions and answer questions about the information (Khan et al., 2021; Bradski & Kaehler, 2008). ML algorithms can be classified as supervised, unsupervised and semi-supervised learning. Supervised learning means that a labelled dataset is used to create a model for predicting values, which is used in classification and regression tasks. In unsupervised learning, the machine searches for patterns in unlabeled data. The most common type of unsupervised learning is clustering by similarity. And semi-supervised learning stands for a combination of the two previous techniques using both labelled and unlabelled data (Khan et al., 2021).

Some of the OpenCV vision functions are based on ML algorithms: Mahalanobis, K-means, normal/Naïve Bayes classifier, decision trees, boosting, random trees, face

detector, expectation maximization, K-nearest neighbours, neural networks/multilayer perceptron and support vector machine (Bradski & Kaehler, 2008).

Although simple ML algorithms can be used in computer vision tasks, they have some limitations. The features to extract have to be exactly specified and there is only one level of feature identification. Because of this, the dependence on the data representation is too high. This impacts the application of ML in tasks when it is unclear which features to extract. The solution found is deep learning, which uses a hierarchy of concepts and representations to make it possible for the machine to learn more complex concepts by combining layers of simpler concepts. For example, to identify a face, there will be layers to identify each set of contours representing an ear, a nose, a mouth and an eye, and higher feature layers to identify what combination of ears, eyes, mouth and nose represents a face (Bengio, Goodfellow, & Courville, 2017).

## 2.3 Computer vision with deep learning

There are different approaches to how using the multiple layers, the parameters and the dependency between them to solve different problems. The computer vision tasks used in this research were image classification and object identification, and the most popular approach to implement these tasks with deep learning is to use Convolutional Neural Networks (CNN). CNN has this name because of the convolution operation made over a grid of pixels. A CNN architecture is composed of three types of layers: convolutional layers, pooling layers and fully-connected layers. All the layers that are not input or output layers are called hidden layers. Figure 1 illustrates a simplified CNN architecture and how these layers are stacked. For better understanding, the layers were divided into four areas.

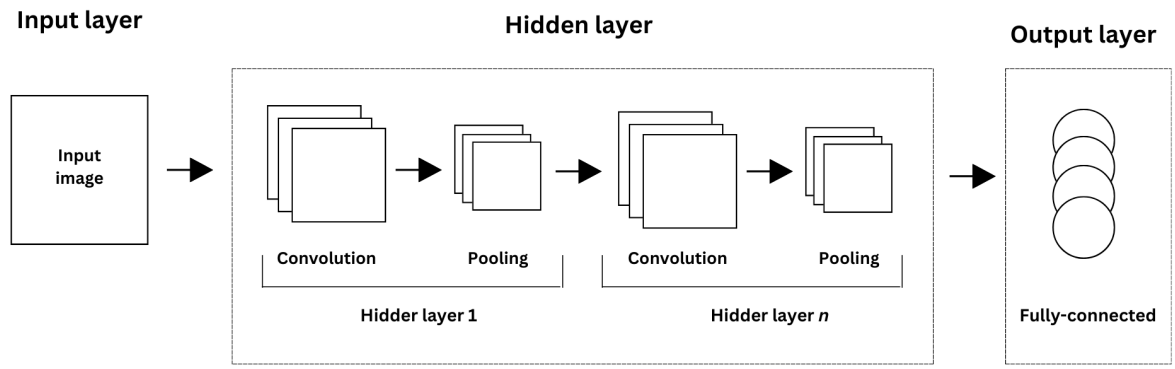


**Figure 1.** Example of a basic CNN architecture.

**Input layer** It is basically the input image read as a 2D matrix. The neurons in this layer just store the pixel values. There is no processing in this layer.

**Convolutional layers** As the name implies, these layers apply convolution operations on the input of the previous layer. This means that the neurons scan the matrix of pixels using a filter. A filter is a matrix of weights that slides in the grid of pixels multiplying each value by the weights and summing the results into an output, creating a feature map. The size of the filters depends on the configuration of the network layer. This size affects the scale and complexity of the features that will be identified in each layer. The stride is the size of the step that the filter makes and can vary, too. Figure 2 exemplifies a convolution operation on an input image 7X7 by a filter 3X3 with stride 2, in the second iteration. This is a key concept in object identification for the reason that the features to be detected are rarely on the expected position, scale, angle and orientation in the image, and the use of filters allows the neural network to identify features in these circumstances.

**Pooling layers** The goal of this functionality is to reduce the spatial dimension of the feature map received from the previous layer. There are a variety of functions that can be applied in the pooling layer, but in most cases, the $MAX$ function is used.

**Fully connected layer** The neurons in these layers are connected to the hidden layers and use the feature map generated to predict classes and the respective probabilities.

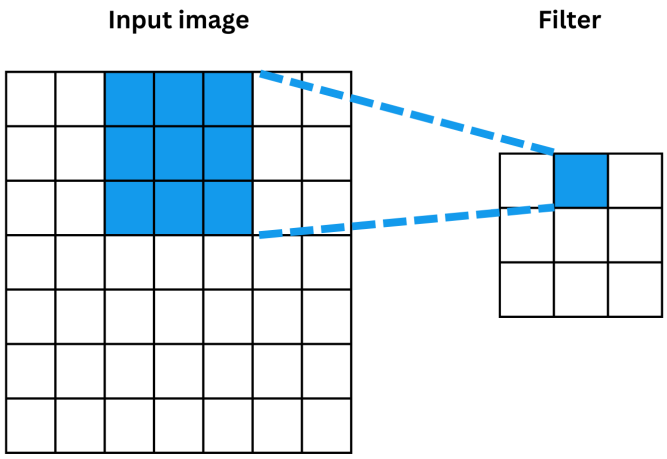


**Figure 2.** A simplified visual representation of the convolution operation.

YOLO (You Only Look Once) is a recent and famous algorithm that uses CNN for image classification and object detection (Jocher, Nishimura, Mineeva, & Vilariño, 2020). The main advantages of using YOLO are the model's small size, strong generalization and fast calculation speed. The weak points are the inaccurate positioning and the lower recall rate compared with other methods (Jiang, Ergu, Liu, Cai, & Ma, 2022). There are five main versions, and although the innovation in the last version (YOLO V5) is questionable compared to YOLO V4, it introduces improvements in performance, size of the model and usability, with the PyTorch framework (Jiang et al., 2022). Jocher et al. (2020) also states that the last version outperforms all the SOTA algorithms in speed and accuracy. YOLOv5 was built to have customization on the model's size to control the trade-off between inference speed and performance. The model size options are YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large) and YOLOv5x (extra large). Larger models mean a bigger network with more parameters and, consequently, a better inference performance, but also demand more processing resources to train, a longer time to infer and more disk space on the device. YOLOv5n, YOLOv5s and YOLOv5m models are recommended for situations when inference speed and disk space are important, like mobile applications and real-time inference (Jocher et al., 2020). YOLOv5 exports the models *best.pt* and *last.pt* after each epoch. *last.pt* is the model generated by the last

epoch. *best.pt* represents the model with the best performance so far when tested against the validation dataset, assuring it is the best model before overfitting.

## 2.4 Assessment of ML models

A model is useful only when it generalizes well, in other words, it can accurately predict unseen data. Two common problems regarding the generalization of the model are overfitting and underfitting, where underfitting means that the model is too simple and does not have the complexity needed to capture the pattern present in the training data, and overfitting stands for an excessive complexity in the model, creating an over-adjustment to the training data that prejudice the prediction on other data.

The generalization of the model can be tested using part of the training data as test data to evaluate the performance. The holdout is a simple and popular method that splits the data into two groups, one to train the model and another one to test its performance. The problem with this method is that if the test data is always the same, it will begin to interfere with the model training and probable cause overfitting. An alternative to avoid this problem is to use the k-fold cross-validation method, in which the dataset is split in $k$ folds and the testing is repeated $k$ times, each time using a different fold for testing and the rest for training. This is an improvement because using different folds of the dataset makes the performance estimate less affected by the test dataset composition (Raschka, 2015). However, not all machine learning algorithms and frameworks have the built-in option to use k-fold cross-validation and the use of this technique demands more computational resources and time, so the possibility and trade offs have to be analyzed in each case.

The confusion matrix is a representation of the prediction results, it summarizes the count of correct and incorrect predictions for each class. The figure 3 shows a confusion matrix for two classes, positive and negative. The prediction results values can be true positives (TP), false negatives (FN), false positives (FP) and true negatives, and are used to calculate the metrics for the model evaluation. The accuracy (ACC) can be described by the equation

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} \tag{1}$$

and represents the percentage of the correct predictions, calculated by dividing the

sum of the positives by the total number of predictions.

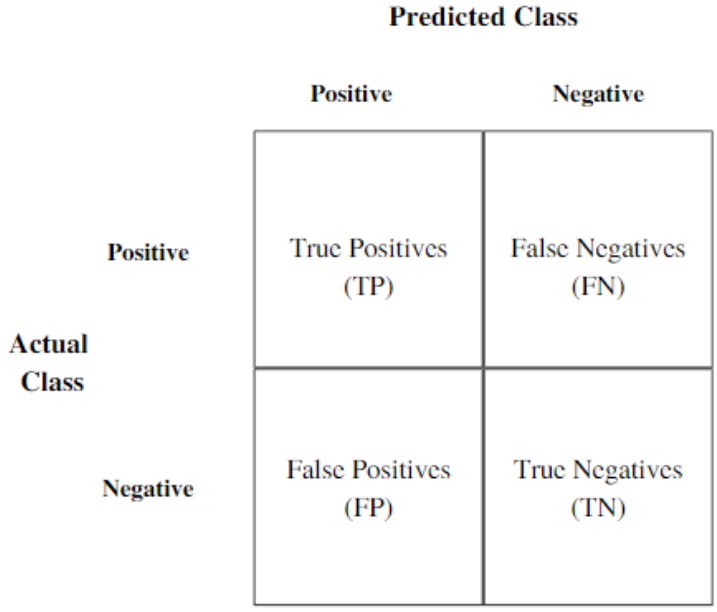


**Figure 3.** Confusion matrix.

In some situations, the problem demands more correct predictions on a specific class. For example, when the model needs to detect true positives with more confidence, a good approach is measuring its precision (PRE), described by the equation

$$PRE = \frac{TP}{TP + FP} \tag{2}$$

On the other hand, if it is more important to detect all the true positives, even at the cost of a lower accuracy, recall (REC) or sensitivity is a more useful metric. Recall can be expressed as

$$REC = \frac{TP}{TP + FN} \tag{3}$$

Additionally, F1 measure or F1-score is a combination of precision and recall and it is effective in identifying when there is an unbalance in one of the classes that do not affect accuracy (Parker, 2010; Raschka, 2015). The following equation describes F1-score:

$$F1 = 2\frac{PRE \times REC}{PRE + REC} \tag{4}$$

Another method to measure the trade-off between precision and recall is the precision-recall curve. It shows the combination of precision and recall for each confidence threshold.

Figure 4 is an example of precision-recall curve. Like the F1-score, it helps to identify problems of class unbalance, because the analysis includes the area under the curve (AUC). A higher AUC means more combinations of high precision with high recall.

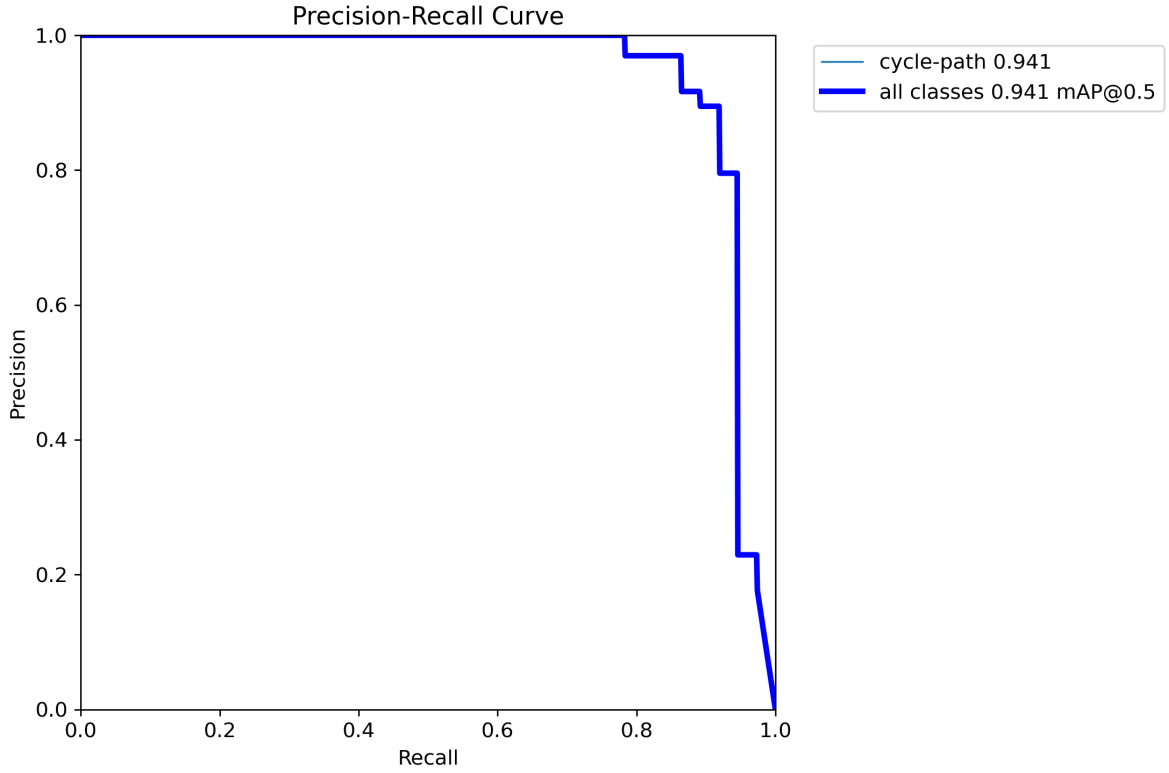


**Figure 4.** Example of precision-recall curve plotted by YOLOv5.

In the object detection task, the output presents the predicted bounding boxes in addition to the predicted classes. It is almost impossible that the models predict the exact bounding boxes coordinates and it is not even needed in real world applications. Padilla, Netto, and Da Silva (2020) explain that the concept of correct prediction depends on the intersection over union (IOU) being inside a threshold. IOU is obtained basically by dividing the overlapping area with the union area, represented by the equation

$$IOU = \frac{area\ of\ overlap}{area\ of\ union} \tag{5}$$

Figure 5 illustrates the concept of intersection and union with bounding boxes. The main idea here is to determine how much of the predicted bounding box is covered by the ground-truth bounding box, and how much the predicted bounding box is covering an incorrect area. The nearer the intersection area is to the union, the most accurate the

prediction is. The prediction of the bounding box is correct if the IOU is greater than the threshold.

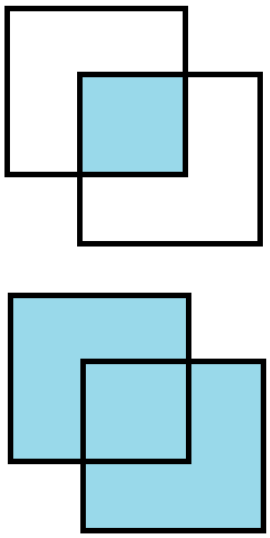

**Figure 5.** Intersection over union.

The performance of an object detection model is interpreted slightly different than a classifier. Precision is a more significant measurement because object detection has the additional concept of predicting the bounding box coordinates. That makes average precision (AP) and mean average precision (mAP) the most common metrics used to measure the performance of object detection models. AP summarizes the values of the precision-recall curve. AP is calculated as

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{6}$$

being a weighted mean of the precision values at each confidence threshold, using the increase in the recall from the *n-1* threshold to the $n$ threshold as the weight (Pedregosa et al., 2011). Once the AP is calculated for each class, the mAP can be obtained by calculating the mean between the AP of all the classes. As mentioned before, in object detection tasks the precision is measured using a threshold for the IOU. This threshold can be a single value, like 50% (mAP50) or 75% (mAP75) or an interval of values, such as 50% to 95% (mAP50-95) (Padilla et al., 2020). YOLOv5 outputs both mAP50 and mAP50-95 in the training records, being mAP50 the most used in papers to report results.

By definition, a good object detector has a high mAP which is obtained with both

high precision and high recall for the most part of the confidence range, or a high AUC as explained before. The perfect mAP is 1.0, meaning all the ground truth annotations were correctly predicted, but since this is improbable with a real world problem, the mAP must be interpreted depending on the characteristics of the problem. For known public datasets, the trained models' performance can be evaluated by comparing them with the SOTA mAP reports.

# 3 Literature Review

This literature review explores previous research in the three parts that compose the cyclist safety and comfort assessment task in this research work: the methodology to identify the factors and weights that define cyclists' perception, the approaches to automatize this task and the computer vision techniques needed to support it.

## 3.1 Calculating the bikeability index

The first step in creating a methodology to assess the cycling risk in the routes for cyclists is to understand the factors that influence the safety perception and the relevance each one represents.

In Manton et al. (2016) the author combines two methodologies to assess the cycling risk in Galway City (Ireland) with the city transport infrastructure inventory. In the survey, participants were asked to draw the route they used regularly and to color according to their perception of safety. After that, they answered a stated-preference survey about elements of the infrastructure, the traffic itself and their experience as cyclists. A generalized linear mixed model was used to predict the color put on the routes and had an accuracy of 67% for all colors and 92% for predicting the green(safe) segments, showing an agreement between the perceived risk and the objective risk. This means that an assessment of a safe route by the presence of infrastructure can result in a route that will be perceived as the safest by the commuter too. This study also identified the cycle lane as the preferred infrastructure to use, with a higher preference when separated from traffic. The most negative factors were traffic with high speed, traffic density and the presence of trucks.

Some of these surveys resulted in frameworks or tools for different types of route safety assessment. One example is the audit tool M-SPACES (Gullón et al., 2015) which used the identified factors and respective weights to measure the association of the urban environment with levels of physical activity in the streets of Madrid (Spain). The weights were defined with interviews and reaching a consensus between experts. The scanning of the city was made manually by field researchers virtually (via Google Street View) and physically. The study validated a good level of agreement between the physical and virtual audit, showing that the gap of time on the Google Street View pictures does not

represent a critical problem in this case, and this could be an indication that this will not be a big obstacle to automate the audit.

## 3.2 Using computing to automate the route assessment

There is a rising number of recent studies exploring how to automate the assessment of urban space for different objectives and types of transportation methods. iWalk (Pisco & Marques-Neto, 2021) was created to measure the quality of the urban infrastructure from the point of view of pedestrians without scanning the space personally. The author compiled several studies to choose the most relevant factors to calculate the walkability index. The solution retrieves the data about the environment analyzed from public geospatial databases to substitute in-person visits and be a scalable tool. The study case used public data from Lisbon (Portugal). According to the authors, the solution was evaluated as efficient, scalable and capable of reaching the goal of calculating the walkability index for the segments.

Although Pisco and Marques-Neto (2021) was able to assess most of the urban environment quality using only geospatial data, it still misses relevant information that generally is not present in public geospatial databases, like the quality of pavement and unmapped obstacles. To cover this, some recent studies complemented the data from geospatial databases with features detected using computer vision. In De Bock and Verstockt (2022) the author proposes a video processing pipeline with a safety scoring mechanism for assessing cycling racecourses. Factors about the format of the route are retrieved from OpenStreetMap(OSM) (the shape of turns, road types, declines and roundabouts). Unmapped details like manholes, signalization poles and cracks on the asphalt are identified using a computer vision model over the frames of a geotagged video. The author validated this automated approach against an analysis made manually by experts, and although the results show the need for some finetuning, the tool demonstrated the potential to be used in real scenarios.

## 3.3 Identifying traffic objects with deep learning

In order to create a pipeline to assess the commuting routes, an object detection model to identify each of the relevant route features is needed. Features such as pavement type, cracks and asphalt defects, and cycle path are key to tackle the issues reported in this

document.

### 3.3.1 Pavement type detection

The study in Nolte et al. (2018) aimed to create a camera-based surface detection model using Deep CNN to determine road-tire friction coefficients and parametrize vehicle control algorithms. To create the training data, a mixed dataset was built using multiple publicly available datasets with pictures of different surfaces representing both wet and dry surfaces. The dataset was balanced by adding pictures of the less representative surfaces from Google Image search. Two architectures were used for classification performance comparison using ResNet50 and InceptionNetV3 applying batch normalization and data augmentation mirroring, scaling and rotating the pictures. ResNet50 demonstrates a slightly higher average classification accuracy in the experiments than InceptionNetV3. The InceptionV3 architecture had a test accuracy of 90% with the basic dataset and started overfitting more easily with the additional images from Google image search, ending with an accuracy of 84%. The ResNet50 architecture reached an average accuracy of 92% trained with the extended dataset. Although the average accuracy of ResNet50 architecture outperforms the results of other referenced studies, there is misclassification of "wet asphalt" and "dirt" as "asphalt" which represents a critical obstacle to assessing the road in real-time, but probably will not be a problem for a static assessment of a city. Although the main application of this study is supporting vehicle control algorithms in real-time, it has shown the possibility of adaptation to identify pavement characteristics related to cycle safety.

### 3.3.2 Crack and asphalts defects detection

A risk factor usually not present in cities' geospatial databases is the presence of cracks, potholes and other pavement defects. In L. Zhao et al. (2022) the author proposes a framework to detect pavement diseases from images named DASNet, a deep CNN based on the Faster RCNN architecture. There are 3 modules in this framework, the first being a deformable convolution module for feature extraction, meaning that the convolution layer has an offset layer to adapt to irregular shapes. The second module is a feature pyramid network using AugFPN. The third module is a sample weighted loss function. When the feature detected has a large curvature the detection bounding box can have an imbalance

between the foreground and background, and the sample weighted loss function can help to solve that. After comparing it with other SOTA detection methods, the author reports that the DASNet approach outperforms them with 41.1% mean Average Precision (mAP).

Another approach is presented in De Bock and Verstockt (2022), which used computer vision to detect 5 types of defects in the racecourses: lateral crack, longitudinal crack, alligator cracks, potholes and white line blurs. The Road Damage Detection challenge dataset (Arya et al., 2020) was used to train a model with the YOLOv5s configuration. The authors reported a mAP of 55% and explained that, despite the fact that this is not an acceptable performance for an object detector, this was mitigated when used in the field because they used an average of detections in multiple frames.

### 3.3.3 Identify cycle paths

Even if cycle lanes are usually mapped in city maps, geospatial databases or crowdsourced databases, it is important to know how to detect them in images because the sources of data can be out-of-date or incomplete. To solve this problem, Saxton (2022) created a deep learning model using the TensorFlow 2 Object Detection API to detect cycle lanes. This study was done in Victoria (Australia) where the cycle lanes are always identified by a bicycle marking on the ground, for this reason, the chosen strategy was to train the model to recognize this symbol. The author built a custom cycle lane dataset using Google Street View images and used it to increment a pre-trained model from TensorFlow 2 Model Garden called "CenterNet HourGlass104 512x512", which is trained with the COCO 17 dataset. The author claims that the detection model had good results, when tested against the validation images it achieved 100% recall and 92% precision, and when used to map an area it was able to identify segments of bicycle lanes not mapped on the official dataset nor the OpenStreetMap.

# 4 Data and methods

This section details the methods, technologies and data used to build and validate the proposed assessment tool.

## 4.1 Technologies

These are the tools that were essential to prepare the datasets, train the deep learning models and build the assessment tool.

### 4.1.1 PyTorch

PyTorch is a machine learning framework written in Python and is applied mainly in computer vision and natural language processing. One of the main advantage of using PyTorch is the capability of integrating with Compute Unified Device Architecture(CUDA) to improve the training and inference performance using GPU memory (Paszke et al., 2019). PyTorch also allows to load trained models and retrieve predictions programatically with some minor differences between object detection and classification modes.

### 4.1.2 Google Street View Static API

Google Street View Static API allows retrieving a panorama image from Google Street View(GSV) with a specific angle and size and without the user interface elements occluding the image. The request to this API can be made via URL or programmatically and the parameters include size, location(in coordinates), heading(horizontal angle) and pitch(vertical angle) (Google, 2023). There are other parameters but they were not explored in this research. Unfortunately for this specific application, the parameter 'heading' refers to angle of the camera based on North (0 degrees) instead of the relative position to the front or back of the car. Adding to that, there is no indication on how to position the camera capturing the street in a certain angle. This created a limitation for this first version of the tool in which it was necessary to search manually for the ideal pitch for each picture retrieved. Probably an integration with a geonavigation API, like Google Directions API[1], would mitigate this limitation allowing to find the rotation

[1]https://mapsplatform.google.com/

degrees to capture the street from an ideal angle.

### 4.1.3 Roboflow

Roboflow is a platform that supports many tasks when building computer vision models (Roboflow, 2023). It allows a human to easily draw the boundaries of the object in images to build an annotated dataset for object detection. Figure 6 illustrates how this annotation works. After building a new dataset or importing an existing dataset, is possible to export it in the most common formats, such as Pascal VOC, COCO json, and the YOLO txt formats, including the YOLOv5 PyTorch format needed for this research. This tool also helps with preprocessing, having functionalities to resize, cut, orient, and manipulate the colours of the images, and with augmentations, like flipping, rotating and colour manipulation. After the dataset is ready to be used, is possible to download the dataset in the desired format or to generate a download command in Python using the Roboflow API, which makes Roboflow also a repository of datasets.

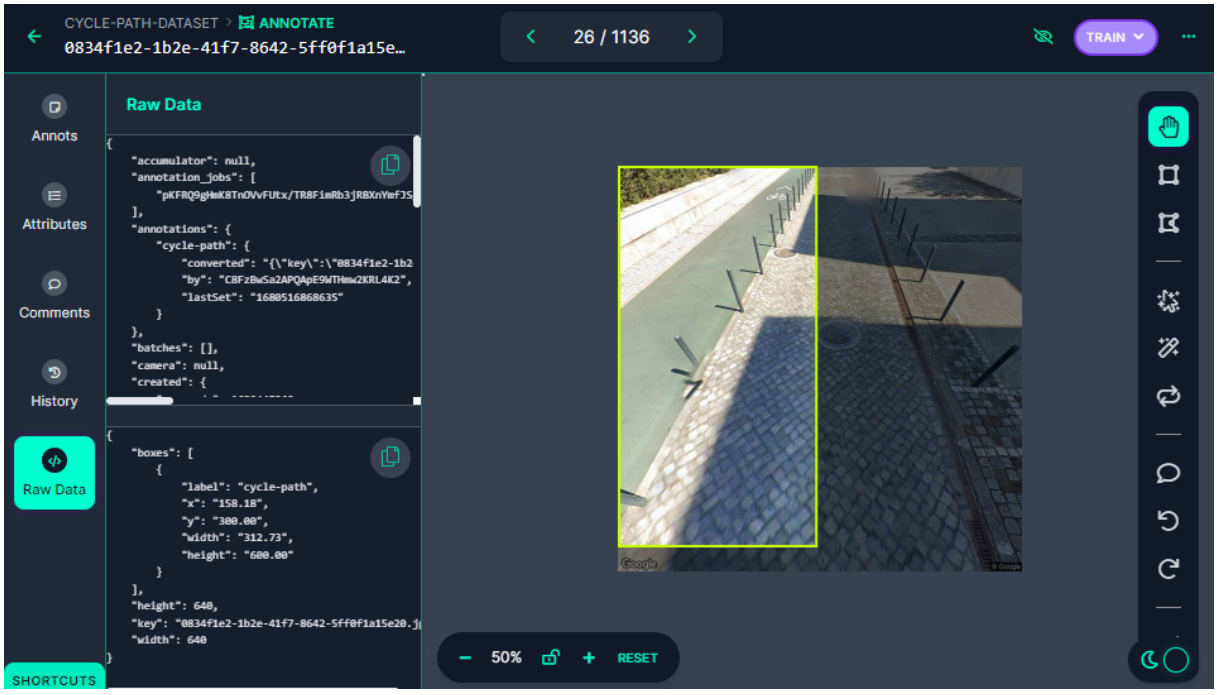


**Figure 6.** Roboflow annotation tool.

## 4.2 Datasets

The data needed for this research were partially adapted from existing datasets and partially created from scratch. The datasets for pavement type classification and cycle infrastructure detection were manually gathered during this work and their creation are detailed in section 4.3. The datasets that were already publicly available are described below.

### 4.2.1 Road network dataset

The public road network dataset available in *Plataforma de dados abertos georreferenciados da Câmara Municipal de Lisboa*[2] holds information about the roads hierarchy, having 5 hierarchic levels according to the road function and traffic flow capacity. Seabra et al. (2011) describes each level of hierarchy, detailed in table 1. In this research, the first three levels were considered major street because they represent roads with higher traffic flow capacity. Furthermore, the roads from the 4th and 5th levels were interpreted as local street for the reason that they have more focus on pedestrian presence and are more integrated with the neighborhood. The less significant roads are not mapped in the road network dataset and by default were defined as local street by the pipeline.

**Table 1.** Roads hierarchy.

| **Hierarchic level** | **Description** |
|---|---|
| 1st | Roads used to travel between counties or long distances inside a county. |
| 2nd | Give access to 1st level roads and handle the largest traffic flows inside the county. |
| 3rd | Internal roads that distribute the traffic to the higher levels roads. |
| 4th | Main streets at the neighborhood level. Higher traffic flow capacity than 5th level, but prioritize the pedestrian flow. |
| 5th | Prioritize the pedestrian flow and give access to the neighborhood local structure. |

[2]https://geodados-cml.hub.arcgis.com/datasets/rede-viaria

### 4.2.2 Pavement defects dataset

The Road Damage Dataset (RDD2022) is a dataset of road images built for the Damage Detection Challenge. This dataset has 47,420 road images from six countries (Japan, India, the Czech Republic, Norway, United States and China) and has more than 55,000 annotations, labelled mainly with four types of road damage: longitudinal cracks, transverse cracks, alligator cracks and potholes (Arya, Maeda, Ghosh, Toshniwal, & Sekimoto, 2022). There annotation format is PASCAL VOC format, which is a popular annotation format based in XML for object detection tasks. The dataset was created to be used in the development of solutions for automatic damage detection. Figure 7 has some examples of annotated images.

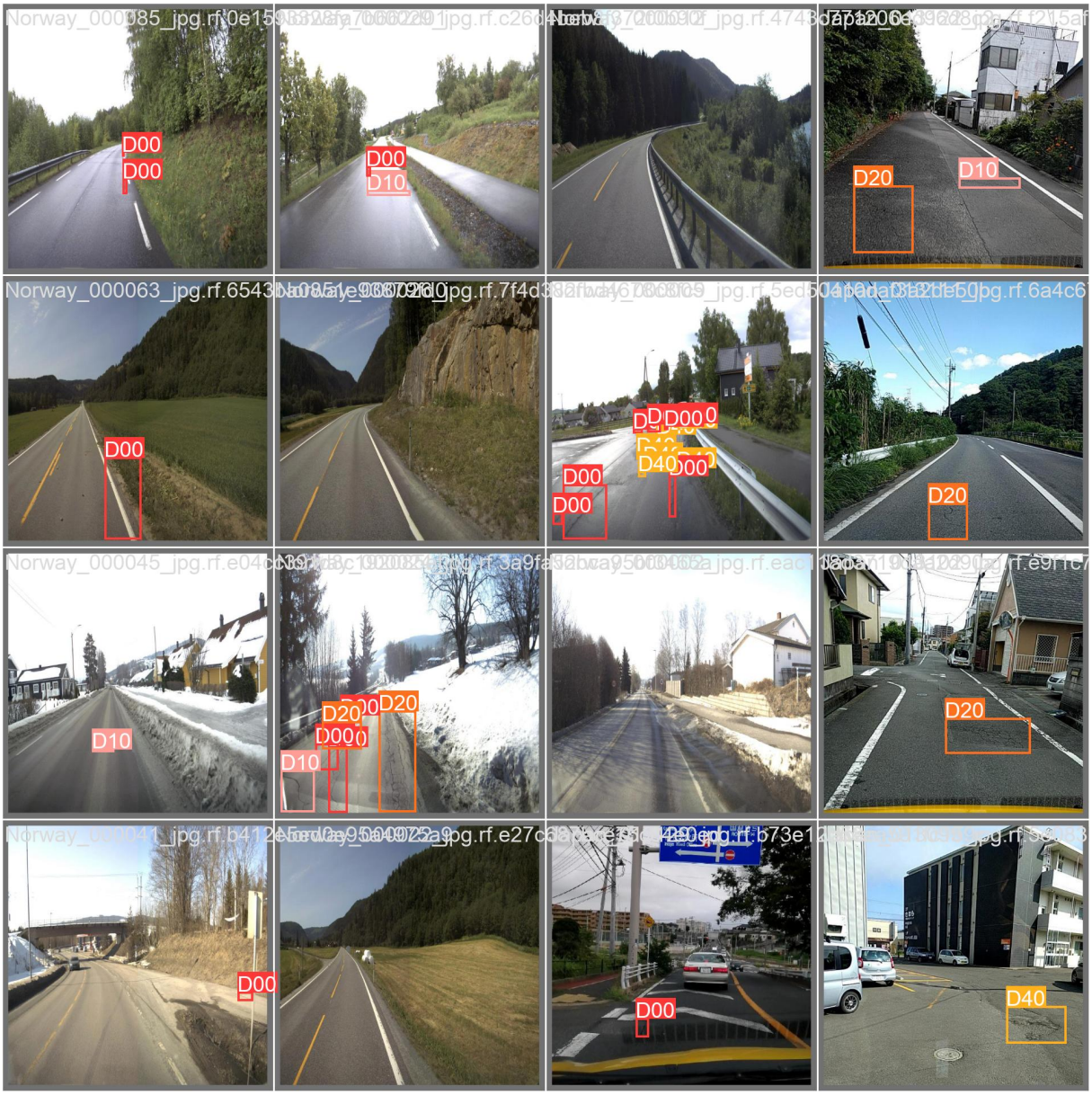


**Figure 7.** Annotated images from the RDD2022 dataset.

## 4.3 Methodology

The assessment tool proposed in this research work is a pipeline implemented in Python that combines different sources of information to calculate a score for a route segment based on traffic features. The chosen features are the presence of cycle lanes or cycle paths, type of pavement, presence of defects on pavement and type of road (major or local). The parameters required to assess a segment are a picture of the road and the coordinates when the picture was taken, to fetch the road network dataset. The trained computer vision models were loaded and used in the pipeline using PyTorch. This section

describes in detail how each feature is extracted and interpreted in the score system. The sequence of steps executed by the pipeline is illustrated in figure 8 and described in the list below.

**Parameters** The image used as a parameter can be retrieved from an online API or dashcam footages, since the angle captures the pavement clearly and a part of the cycle infrastructure. For this experiment, the Google Street View Static API was chosen to retrieve the images for practical purposes, because it allows to control the camera angle and resolution. The coordinates should be the same coordinates used to retrieve the picture.

**Cycle infrastructure model** First, the pipeline detects any cycle path or cycle lane on the image. The cycle infrastructure model results are interpreted straightforward, if there is any detection the score is attributed. Most of the cycle infrastructure in Lisbon is separated from traffic, either by a physical barrier or by being on the sidewalk, hence in this version it is not implemented different scores for being separated or not.

**Pavement type model** After that, the picture is classified into asphalt, cobblestone or unpaved. If the pavement is classified as unpaved or cobblestone by the pavement type model, the score is attributed directly, without searching for defects on pavement. Cobblestone are not rough as unpaved pavements, at the same time that it is not smooth and comfortable as asphalt, thus it was decided to keep a different class for it.

**Pavement defects model** This is model uses object detection to detect defects in the pavement, such as cracks or potholes. Only the pictures classified as asphalt are processed by this step. There is no reason to infer if unpaved pavements have defects, since they are naturally rough and uncomfortable. Although cobblestone pavements can also have defects that put cyclists in danger, a proper dataset with defects in this type of pavement was not found. Even though the pavement defects model can infer multiple classes of defects and more than one defect per picture, the literature does not inform correlations of this type of details with the cyclist perception, thus the score is attributed with any defect found. Besides that, the detailed output was kept internally to be used to analyze the results later.

**Road type retrieval** The last feature is the road type, which can be major or local road. The road network dataset is fetched using the coordinates from the parameters and the road hierarchy is used to identify the type of road. The correlation between the type of road and hierarchy is explained in detail in section 4.2.1.

**Output** Finally, the pipeline uses the feature values and the parameterized weights to calculate a score for the route segment. The output is composed by the score itself and a list of the identified features.

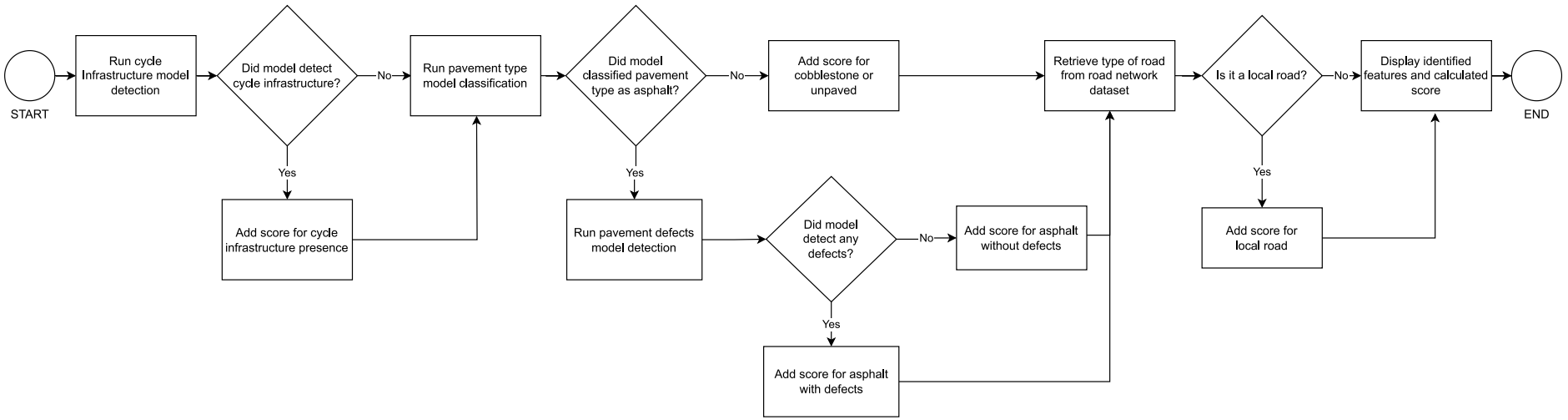


**Figure 8.** Diagram showing the structure of the assessment pipeline.

### 4.3.1 Computer vision models

The method to train the three computer vision models were trained using YOLOv5 to extract the route segment features image is explained in this section. As explained in section 2, YOLOv5 automatically selects the best model based on the performance against the validation dataset. Hence, the models were trained until they reach the overfitting point or stagnation and *best.pt* was considered the best result. For the same reason, the performance results reported are mAP achieved against the test dataset, except for the learning curves.

#### 4.3.1.1 Pavement type model

YOLOv5 were used in classification mode to train a model to identify if a pavement type is asphalt, cobblestone or unpaved. The first experiments were made with a pre-built dataset used for pavement detection for autonomous cars with bumper cams (T. Zhao & Wei, 2022). Although this dataset had the classes needed for this experiment, the resulting model was not capable of producing any generalization to the GSV images, probably because of different resolutions and zoom or pictures being taken in movement with some blur. For this reason, a script was implemented to extract the images in a pattern of angle and image size using the Google Street View Static API.

This implementation was useful to extract images to build the cycle path and pavement type datasets and was used to extract the image from GSV with the same pattern to be assessed by the pipeline. The parameters used are size 640x640 and pitch -50 to achieve a dashcam-like angle, being able to capture the pavement and cycle infrastructure. 1552 pictures of pavement were extracted from the main cities of Portugal, being equally balanced between the three classes. The split train/validation/test was made with the proportion of 75% of the images for the training, 20% for validation and 5% to test the model.

As it is not required to use the entire picture to classify the pavement only a small portion of the pavement texture, was applied a pre-processing to zoom the image 150% and crop in the centre to a fragment of 224x224. This way, most of the resulting images were not occluded samples of the pavement and the training can be more efficient, quicker and accurate than training with 640x640 images with buildings, sidewalks and other elements

around. Figure 9 shows the results of the preprocessing. YOLOv5 has the option to use pre-trained weights on the training and there are default weights available for each model size. It was decided to take advantage of this feature to compensate for the smaller pavement type dataset. The training parameters used for this dataset were 50 epochs, image size 224, batch size 64 (automatically selected for the environment) and three pre-trained weights to compare the results: YOLOv5s, YOLOv5x and ResNet50.

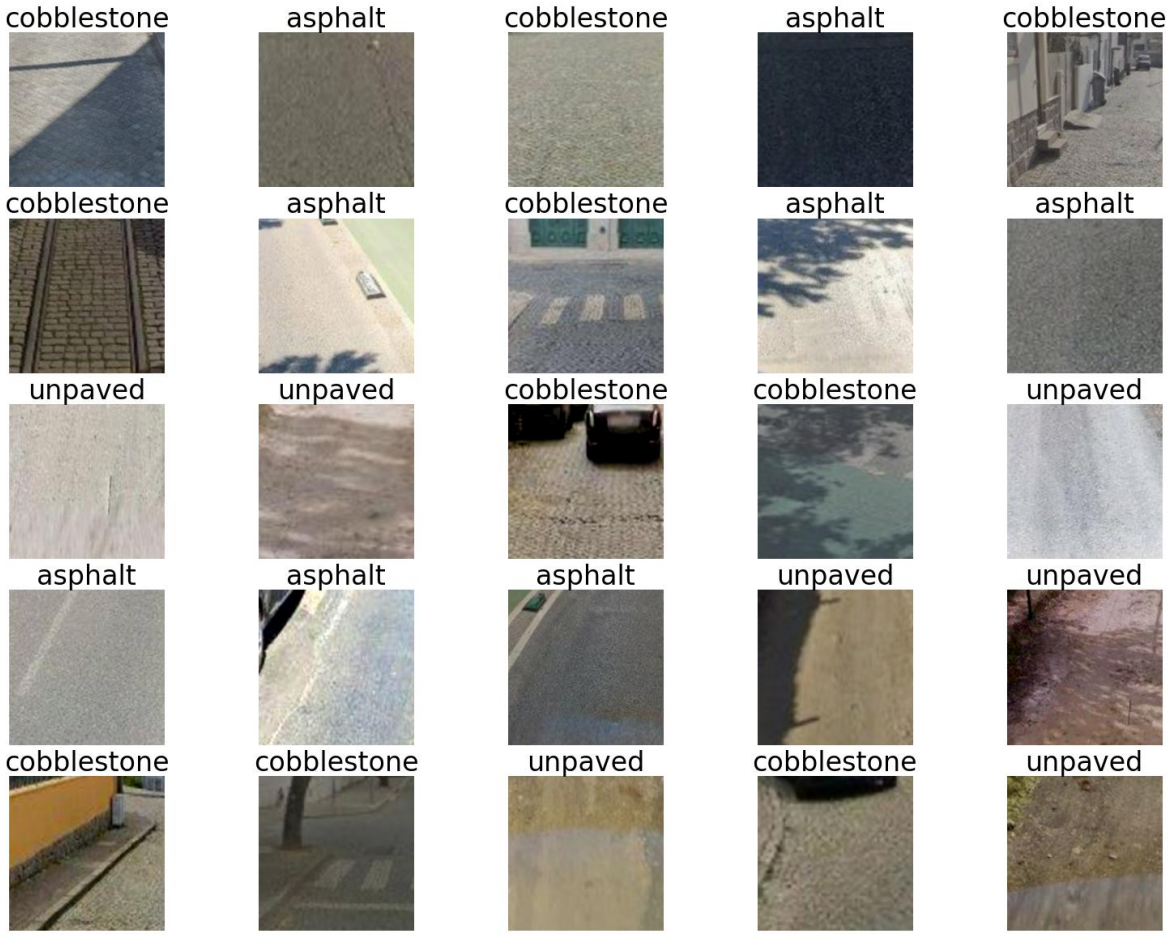


**Figure 9.** Pavement type dataset preprocessed images.

The pavement type model training already reached good results in the first experiments, as can be seen in table 2. The pre-processing strategy probably played the main role in reaching this result, because limiting the image to a part of the pavement area removes other textures on the border of the road, such as vegetation, sidewalk, dirt and constructions, that could influence the model without having a relationship with the pavement type classes. The pre-trained weights helped to reach high accuracy with lesser epochs. The YOLOv5s was the training configuration with better performance,

even though it is the model configuration with the lower complexity. Presumably, the other configurations had been able to reach even higher performance with more epochs, as figure 10 demonstrates that the model was still learning at the end of the 50 epochs. But since the small model had the same level of performance, it will be preferred because it is quicker to infer and demand less disk space. The decision of using pictures from Google Street View not only helped with maintaining the pattern of angle, resolution and saturation but also made the model flexible to different lighting and shadows on the pavement.

**Table 2.** Pavement type classification models performance with 50 epochs.

| Architecture | Accuracy |
|---|---|
| YOLOv5 model S | 0.975 |
| YOLOv5 model X | 0.963 |
| ResNet50 | 0.963 |

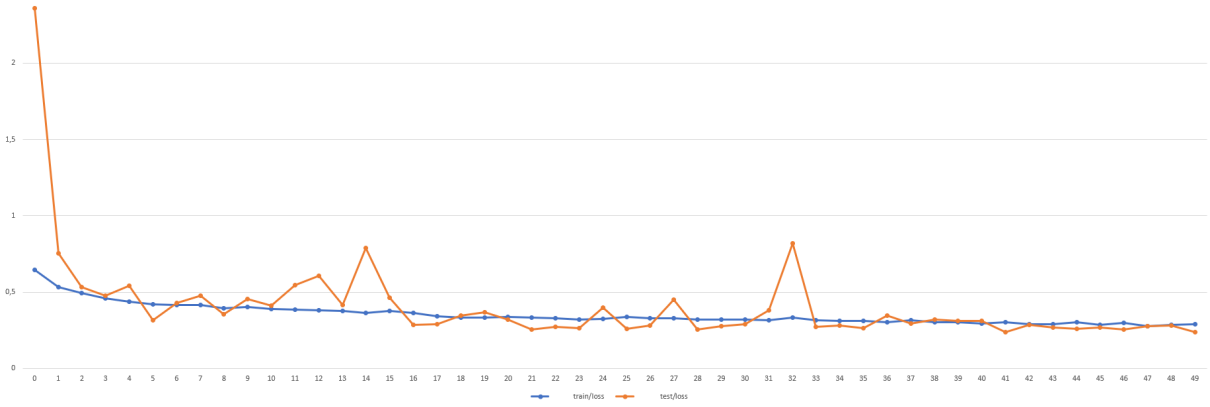


**Figure 10.** Pavement type model learning curve with YOLOv5x configuration during 50 epochs.

After choosing the YOLOv5s as the configuration, a new experiment was run with 1500 epochs to find the best model. Figure 11 shows the training learning curve. Even though there is no sign of overfitting, the test loss curve does not show any lower values in the last hundreds of epochs, indicating that the best performance was reached for this training set. The resulting model performed slightly better, the table 3 details the accuracy per class.

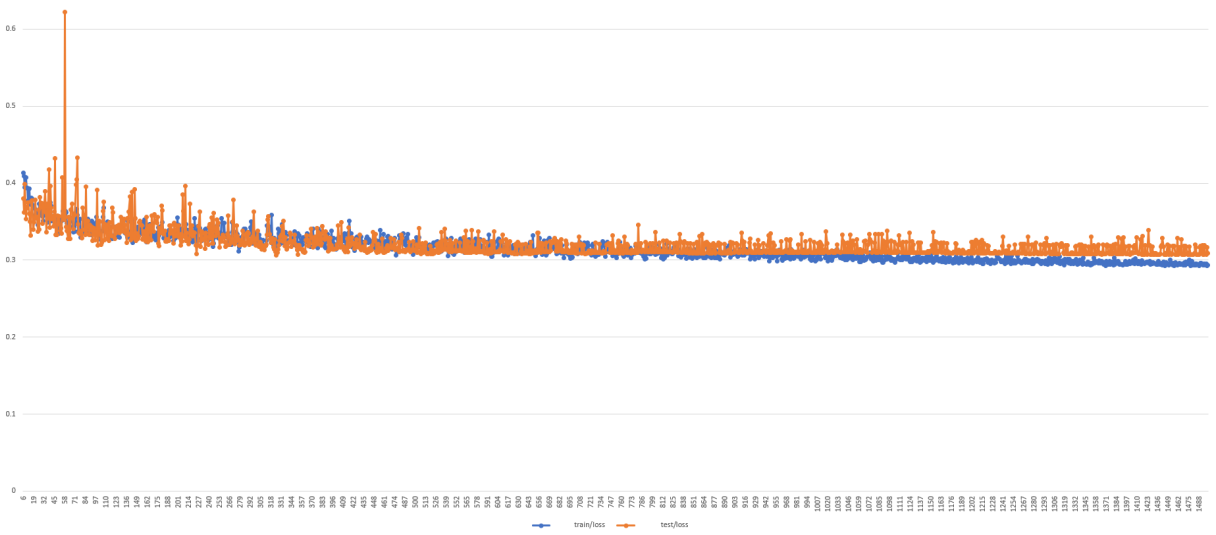


**Figure 11.** Pavement type model learning curve with YOLOv5s configuration during 1500 epochs.

**Table 3.** Performance of the pavement type classification model with YOLOv5s configuration for 1500 epochs evaluated with the test dataset.

| Class | Accuracy |
|---|---|
| All | 0.988 |
| Asphalt | 0.967 |
| Cobblestone | 1 |
| Unpaved | 1 |

#### 4.3.1.2 Cycle infrastructure model

As each location has a pattern of colour or symbol for cycle infrastructure, it is impossible to use cycle lane datasets from other locations, thus, for this research, was required to collect and label the cycle lane images for the target location. The Google Street View Static API also helped in this case, allowing the collection of cycle infrastructure pictures from Lisbon with the same pattern. 1136 images were collected, in which there are images of asphalt and cobblestone roads with and without cycle infrastructure and unpaved roads with and without vegetation around. The size of the images is 640x640. The manual labelling was done using the Roboflow annotation tool. 560 object annotations were made, having in most cases 1 annotation per image.

634 images do not have any object annotated. The split was 89% for training, 8% for validation and 2% for testing. Examples of cycle infrastructure annotations can be seen in figure 12. The YOLOv5s pre-trained weight was also used here due to the limited size of the cycle infrastructure dataset. The chosen image size was 640 and the batch size was automatically defined as 16. The remaining configurations were used as default and the first experiment ran for 100 epochs.

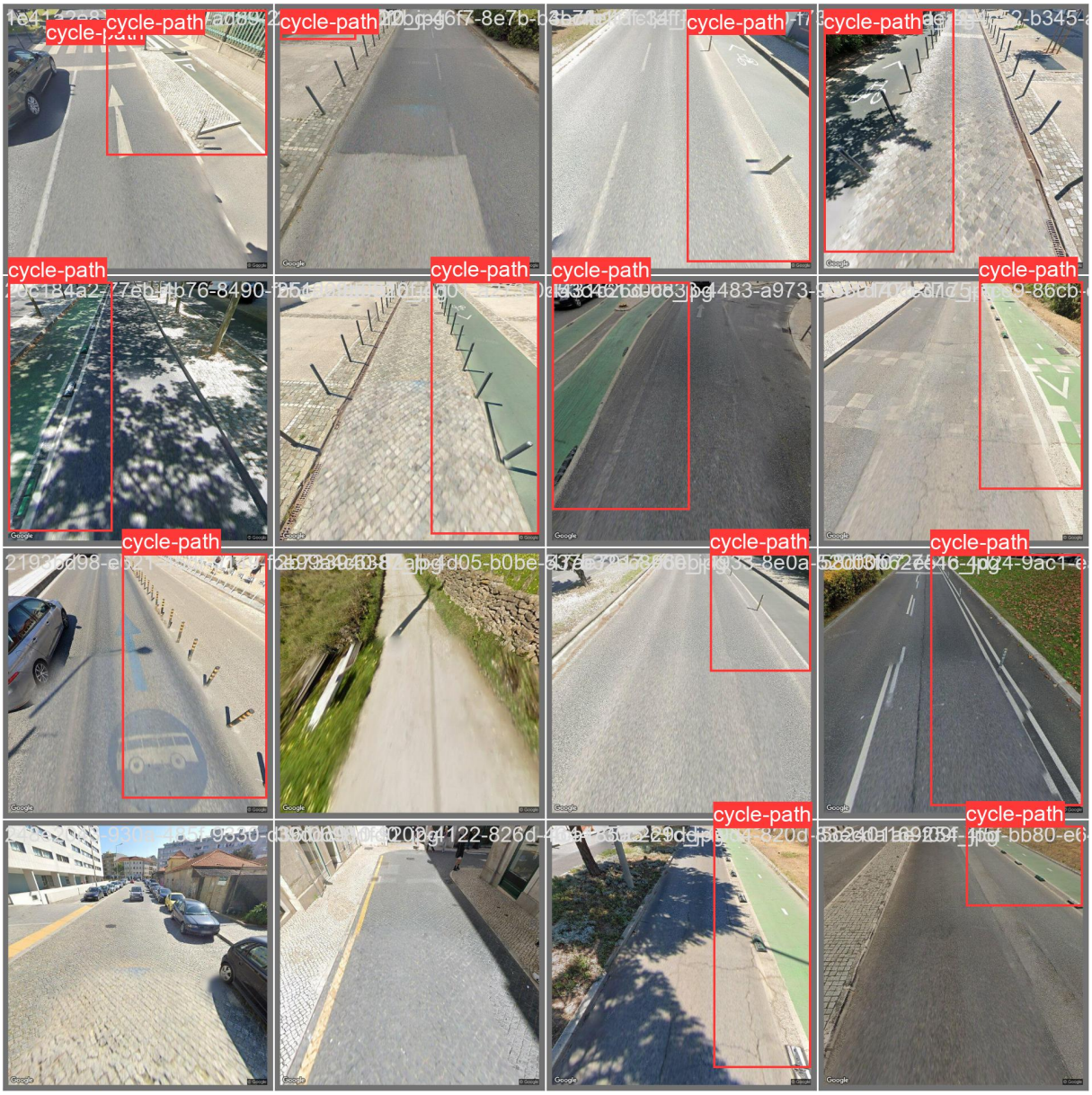


**Figure 12.** Cycle infrastructure dataset annotated images.

The cycle infrastructure detection model also had very good results with the default configurations and the pre-trained model YOLOv5s. In this first experiment, the training was able to reach an mAP of 0.923 at the end of 100 epochs. Besides that, it can be seen in

figure 13 by the *obj_loss* curve that the model is still decreasing the loss value, indicating that the model is underfitting and can be improved by training with more epochs.

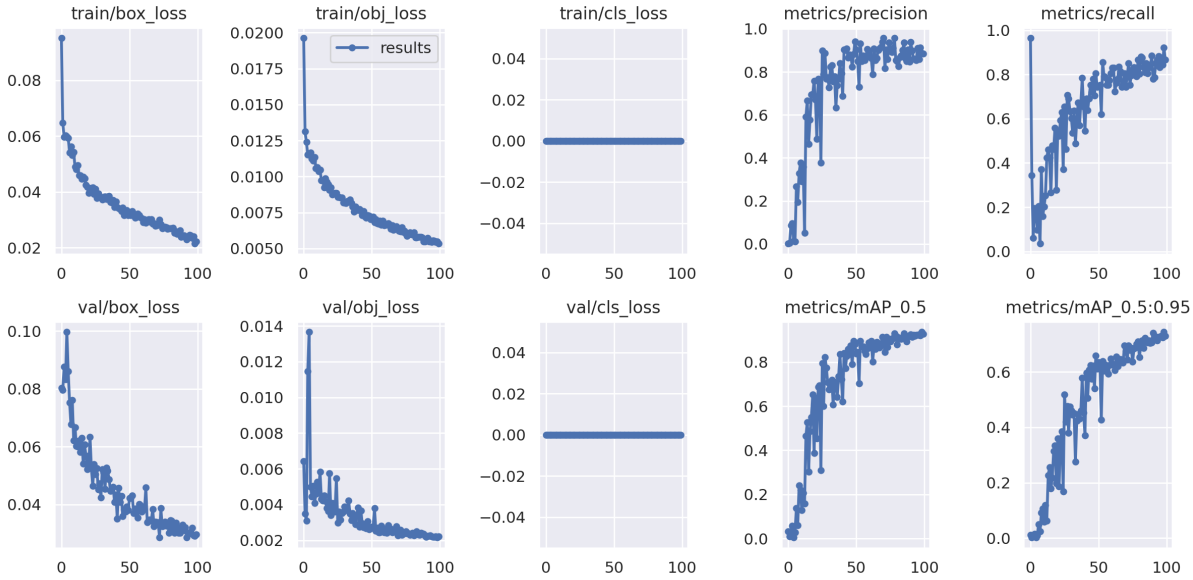


**Figure 13.** Cycle infrastructure detection model training for 100 epochs and pre-trained model YOLOv5s.

Another experiment was run with the same configurations for 400 epochs to find the best model before overfitting. Figure 14 contains the results of the training, in which is possible to notice that 400 epochs were enough to reach the overfitting point. The best model was evaluated with the test dataset and achieved an mAP of 0.941 as shown in the PR curve in figure 15. Despite Lisbon having at least 4 different patterns of cycle infrastructure, the model apparently was able to adapt to the different colours and aspects. Another factor that could have impacted this model's performance was the dataset's small size in terms of annotations. Augmentations of rotation and flip were applied to the training images and this, summed with the use of pre-trained weights, seemed to have mitigated the dataset size limitation.

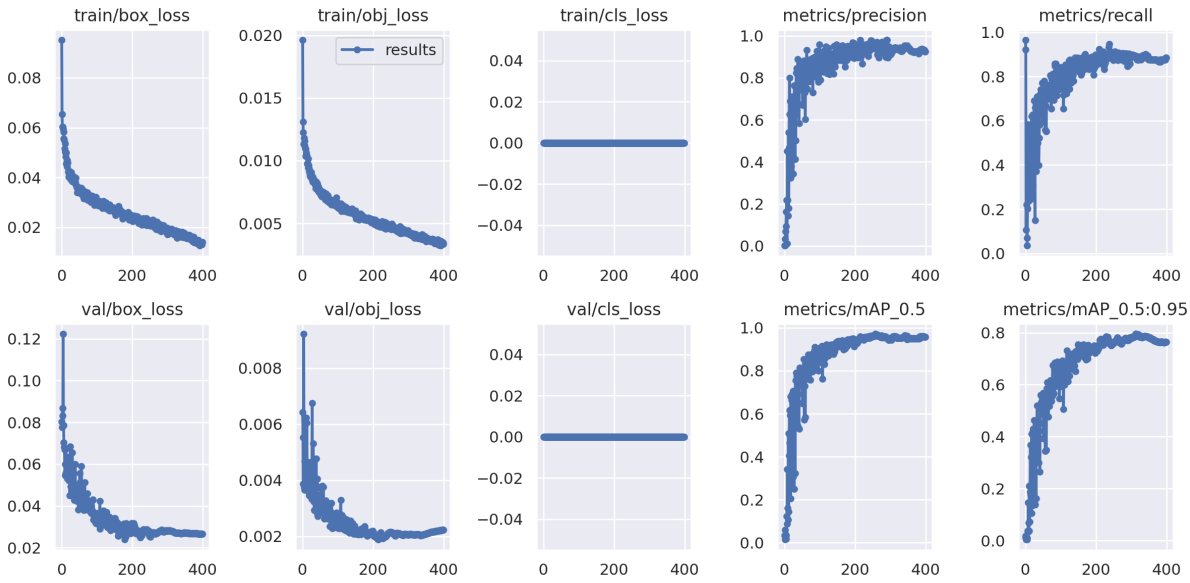


**Figure 14.** Cycle infrastructure detection model training for 400 epochs and pre-trained model YOLOv5s.

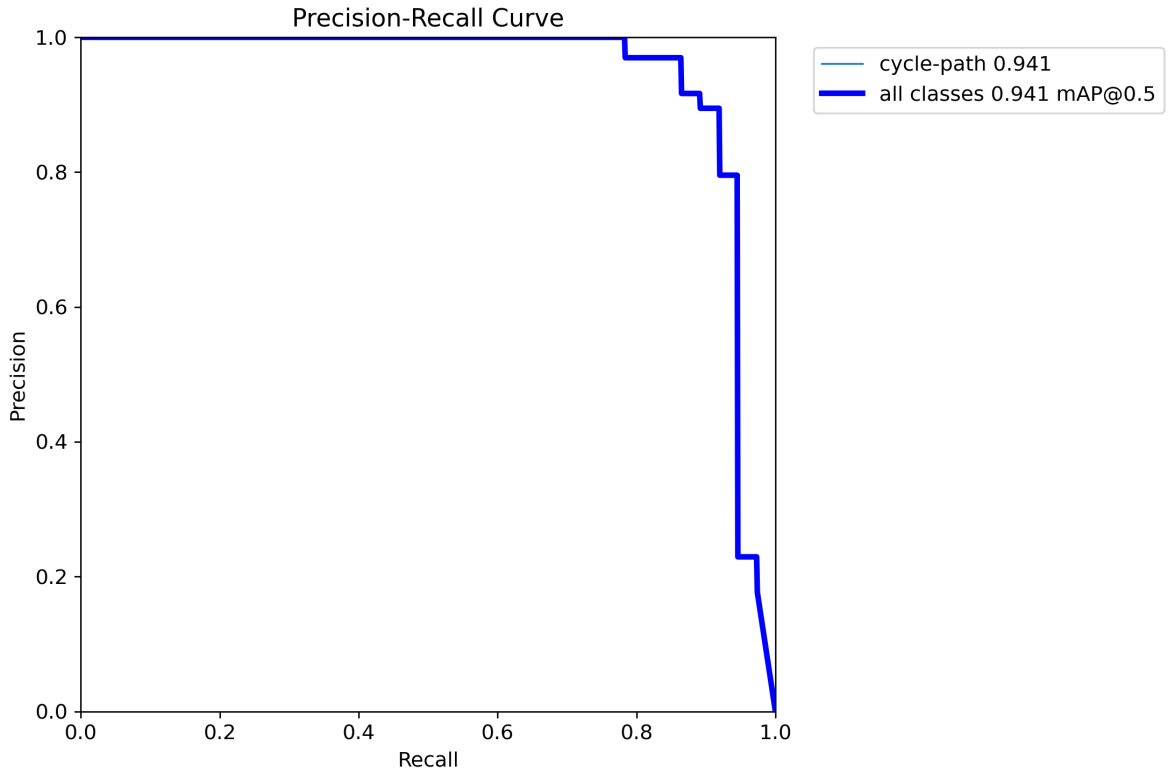


**Figure 15.** Cycle infrastructure detection model PR curve for 400 epochs and pre-trained model YOLOv5s.

#### 4.3.1.3 Pavement defects model

To train the model to detect pavement defects, YOLOv5 in object detection mode was used on the RDD2022 dataset, described in subsection 4.2.2. As YOLOv5 requires its own annotation format called YOLOv5 PyTorch TXT, it was necessary to convert the dataset annotation files. The dataset was uploaded to Roboflow and exported as the target format. This dataset had a considerable size to process during the training, which required a specific training strategy to test and validate the different configurations more quickly. For this reason, the first experiments were ran with the YOLOv5s configuration and fewer epochs. In addition to that, the pre-trained weight YOLOv5s was also used to reduce the training time. The selected image size was set to 640 and the batch size was automatically set to 16. It was run two experiments with and without label smoothing as a regularization technique to delay overfitting.

The training with the default settings for 50 epochs resulted in an mAP of 0.553 and setting the label smoothing parameter to 0.1 the mAP slightly improved to 0.556. The training results in figure 16 suggested that the model can be trained for more epochs before starting overfitting.

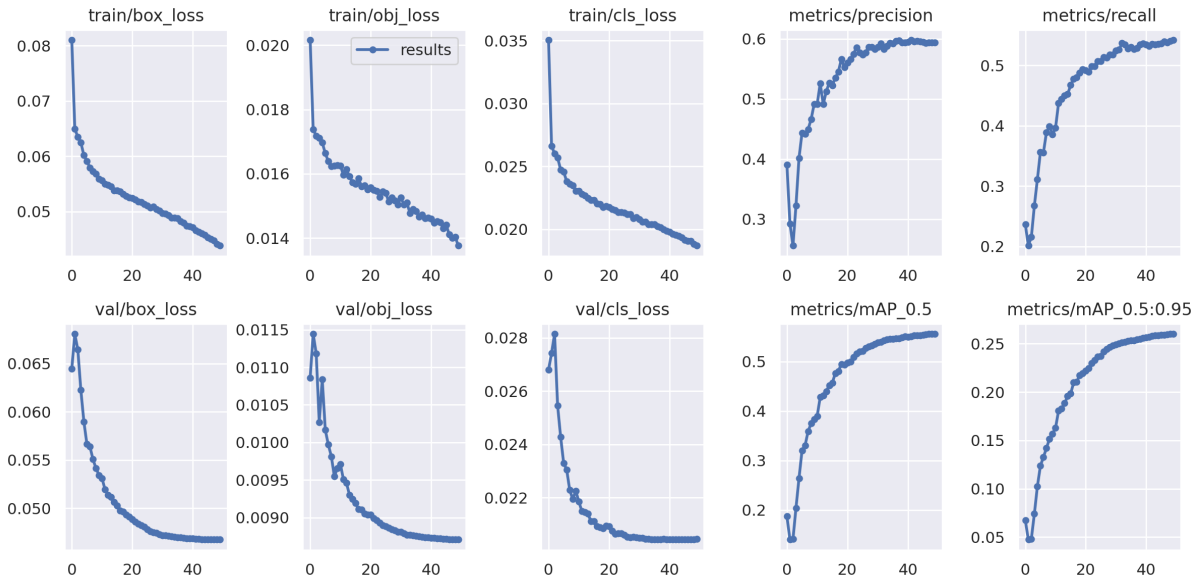


**Figure 16.** Pavement defects detection model training results with label smoothing 0.1 for 50 epochs and pre-trained model YOLOv5s.

After that, another experiment was run with YOLOv5s weights and label smoothing

for 300 epochs to reach the overfitting point. The learning curve in figure 17 confirms that the model was trained until overfitting, having an improvement to a mAP of 0.563. The training did not finish running the 300 epochs because YOLOv5 makes early stops after 100 epochs without performance improvements. Although this performance is already in line with other pavement defects models present in the literature review, there was a possibility that this could still be improved with a more complex model. Since inference time is not a priority in this application, a last experiment was run using the configuration YOLOv5l, which creates more layers in the network and raises the chances to a better performance. The experiment was set with label smoothing equal to 0.1 and YOLOv5l pre-trained weights. Figure 18 demonstrates that the best model was found around 115 epochs, having an overfitting tendency after. As expected, the performance rose to a mAP of 0.594.

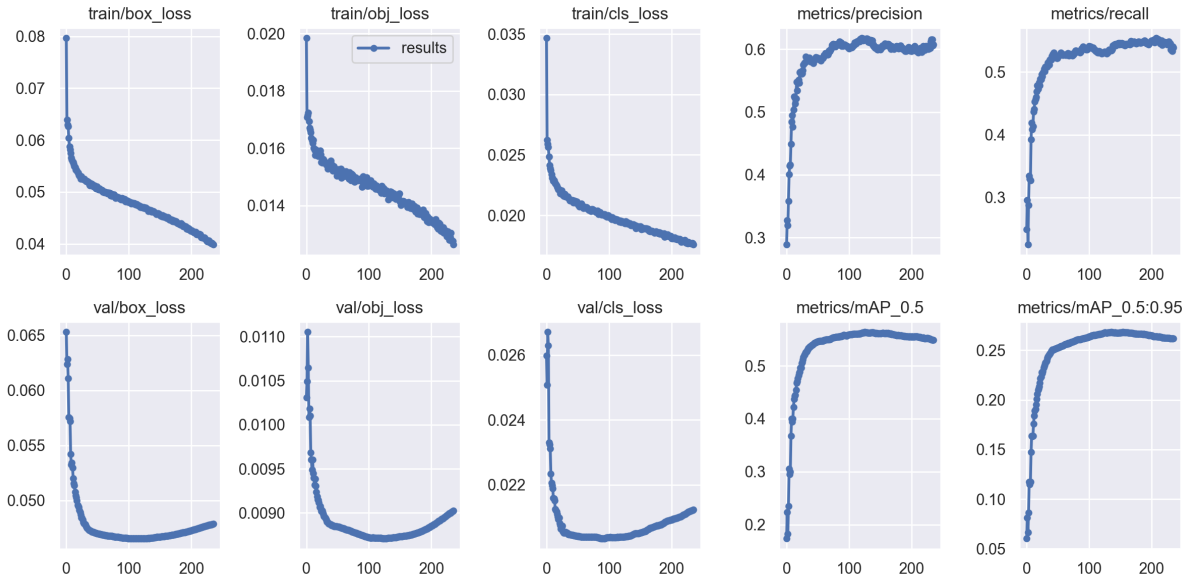


**Figure 17.** Pavement defects detection model training results with label smoothing 0.1 for 300 epochs and pre-trained model YOLOv5s.

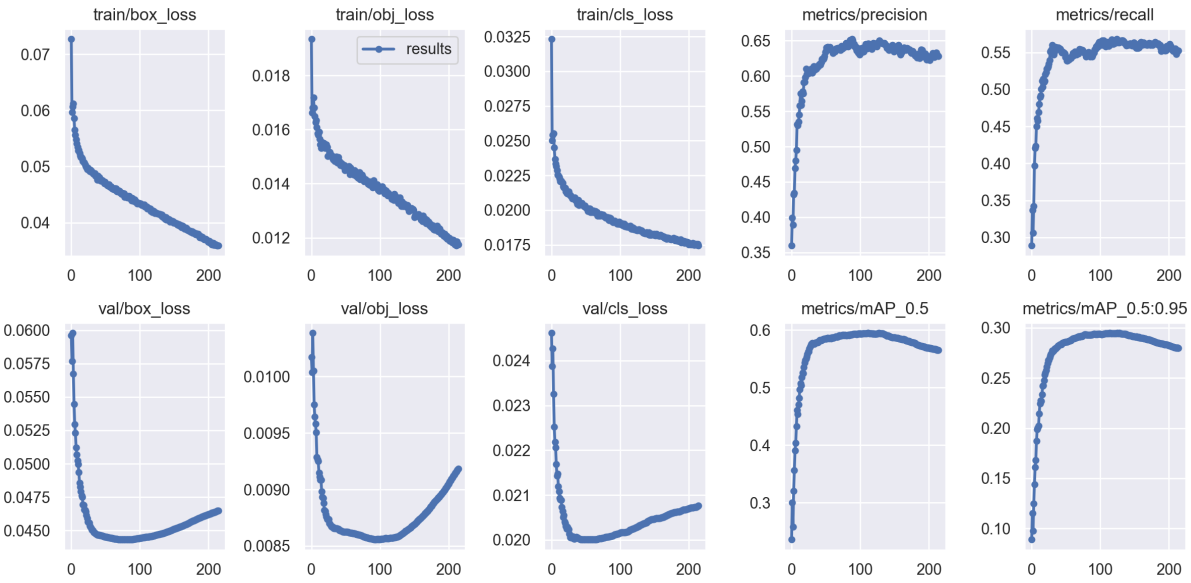


**Figure 18.** Pavement defects detection model training results with label smoothing 0.1 for 400 epochs and pre-trained model YOLOv5l.

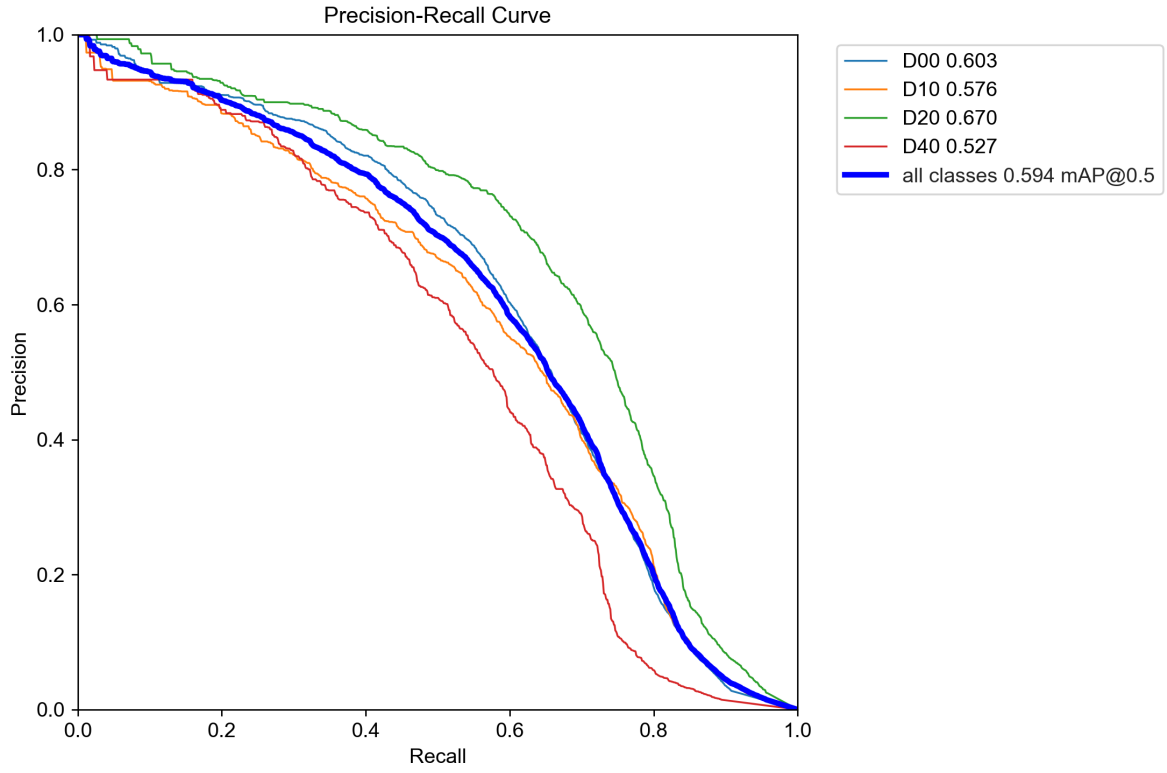


**Figure 19.** Pavement defects detection model PR curve with label smoothing 0.1 for 400 epochs and pre-trained model YOLOv5l.

### 4.3.2 Definition of weights

Although it is not in the scope of this work to define the weights for each feature, it is going to be necessary an initial set of weights that represent the cyclist's perception in order to test the automatic assessment tool. Ideally, should exist a definition of the weight of the safety and comfort perception for each feature present on the pipeline in the target city, but unfortunately, there is no research work like that for Lisbon. Adding to that, a part of the research work on cyclists' perception uses subjective features to measure it and was hardly found papers that had objective values for all the features needed in this assessment tool. Due to this reason, it was decided to use the values present in (Stinson & Bhat, 2003) as a source of parameterization for the reason of having the variables most aligned with our assessment tool.

In (Stinson & Bhat, 2003), the authors use a stated preference survey to analyze the importance of the factors that affect the cyclists' route choice. Two types of factors were used, route-level and link-level factors, being "link" a segment between two intersections. The survey showed a list of preferences, the following are the most useful examples for the context of the assessment tool: the presence of bicycle infrastructure, paved streets over unpaved, smooth pavement over rough pavement, and residential streets over streets with more traffic. The authors use logistic regression to obtain the coefficients that represent the magnitude of the influence of each variable on the cyclists' preferences.

To be able to use these coefficients to calculate a score and rank, it is necessary to translate these weights proportionally to the same scale of the survey (1 - 10) and correlate the features between (Stinson & Bhat, 2003) and our assessment tools. Certainly, "smooth pavement" refers to asphalt without defects, the same way "coarse sand" is the same as unpaved for the assessment tool. It did not specify exactly "rough pavement", but it is a less comfortable pavement between smooth pavement and coarse sand. Ideally, it should be a study to address asphalt with defects and cobblestone with different values, but in this correlation, they both were put inside the concept of "rough pavement". The only bicycle facility type selected to be used is "separate path" because all the cycle infrastructures identified by the assessment tool are separated paths from traffic. Regarding the roadway class, the paper describes three classes: major arterial, minor arterial and residential. Without question, the residential correlates to the local street, but the assessment tool does not differentiate between major arterial or minor arterial, being necessary to join the

two arterials into the category of "major" using an average of the two as the corresponding coefficient. Table 4 shows the correlations made and the coefficient from (Stinson & Bhat, 2003). The negative values represent detractor factors and the positive are factors that motivate the route choice.

**Table 4.** Correlations between the values obtained in (Stinson & Bhat, 2003) and the assessment tool.

| **Reference paper variable** | **Paper coefficient** | **Corresponding feature** |
|---|---|---|
| Roadway class - residential street | 0 | Type of road - Local |
| Roadway class - Major and minor arterial | -1.265 | Type of road - Major |
| Bicycle facility type - separate path | 1.780 | Presence of cycle infrastructure |
| Bicycle facility type - no bicycle facility | 0 | No presence of cycle infrastructure |
| Pavement type - smooth pavement | 0.33 | Pavement type - asphalt without defects |
| Pavement type - rough pavement | 0 | Pavement type - asphalt with defects and cobblestone |
| Pavement type - coarse sand | -0.980 | Pavement type - unpaved |

Once the correlation between the features was made, the next step was to transform these coefficients into the same scale as the assessing tool. The distance between the two extremes of the coefficients was interpreted as the magnitude of the factor's influence on the route choice. Based on the literature review, the perfect route choice (score 10) for the assessment tool have to be asphalt with no defects, in a local street and with the presence of cycle infrastructure. With that definition, it was calculated the proportion of a 10 score for each one of the features based on the coefficients. After that, it was defined the multipliers for each value of the features. For example, the proportion of the feature pavement type to the perfect score of 10 is 2.9, and the extremes "asphalt without defects" and "unpaved" are going to be multiplied by 1 and 0 respectively according to their contribution to route choice. Following the same rule was defined a multiplier of 0.75 for the value "asphalt with defects or cobblestone" because its coefficient is positioned at 75% of the higher value for this feature. Table 5 demonstrates how each coefficient is used to define a proportional part of the score. The final score is the value used to parameterize the assessment tool.

**Table 5.** Coefficients being transformed to the same scale as the assessing tool.

| Feature | Coefficient | Magnitude | Proportion to score | Value multiplier | Final score |
|---|---|---|---|---|---|
| Type of road - Local | 0 | 1.265 | 2.9 | 1 | 2.9 |
| Type of road - Major | -1.265 | | | 0 | 0 |
| Presence of cycle infrastructure | 1.780 | 1.78 | 4.09 | 1 | 4.09 |
| No presence of cycle infrastructure | 0 | | | 0 | 0 |
| Pavement type - asphalt without defects | 0.33 | 1.31 | 3.01 | 1 | 3.01 |
| Pav. type - asphalt w/ defects and cobblestone | 0 | | | 0.75 | 2.26 |
| Pavement type - unpaved | -0.980 | | | 0 | 0 |

### 4.3.3 Experiment

After the evaluation of each model separately and the implementation of the automatic assessment tool, it was made an experiment to evaluate if this tool can predict the cyclists' perceptions. Forty eight route segments were selected for the experiment. A specific questionnaire was created to capture the human perceptions about these selected segments as a score and a list of identified features. Each page has a picture of a route segment and questions about the respondent's impressions. The questionnaire was detailed in section 4.3.4. The questionnaire answers were used to calculate an average of the human score for each segment. For the feature presence questions, could exist different answers for the same questions, depending on the individual perceptions, for example, could exist answers of having and not having defects in the same picture. In this cases, the most answered value was considered.

The trained computer vision models with the best performance were referenced to be used in the assessment tool. Similarly, the weights defined in section 4.3.2 were parameterized in the software. The same pictures shown for the humans were processed by the tool, which exports a similar list of scores and detected feature list. After obtaining both results, a comparison and evaluation were made in section 5. Figure 20 shows the steps of the experiment.

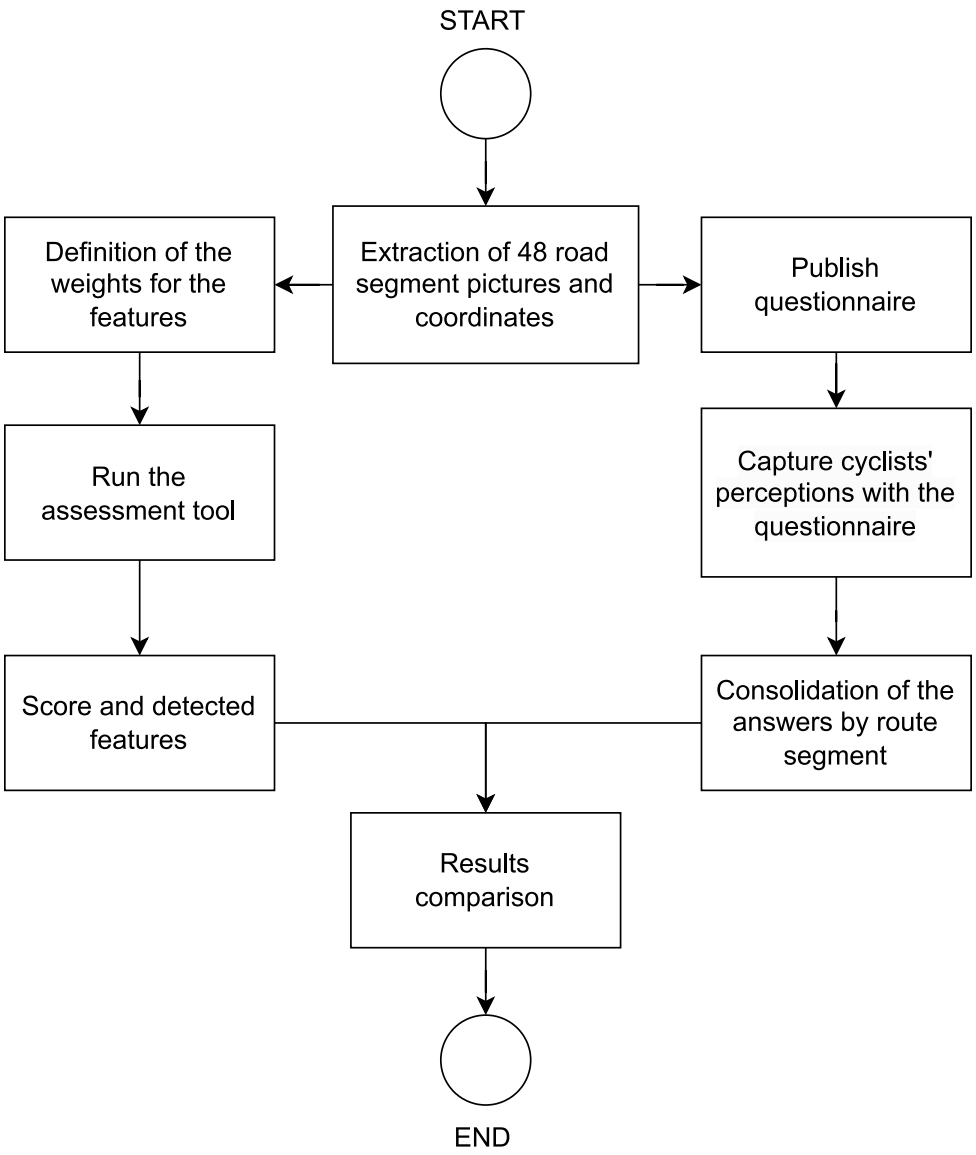


**Figure 20.** Steps to evaluate the assessment tool's performance.

### 4.3.4 Questionnaire

There were two types of questions: close-ended questions asking which traffic features the respondent identifies on the picture and a Likert scale question for rating the level of comfort and safety perceived. For practical purposes, this questionnaire was applied as a web survey and shared via cycling-related groups and communities on the internet. A total of 48 pictures of route segments were selected to be assessed by the cyclists. It was not expected that most of the respondents would answer more than 10 or 15 route segments, so if the route segments were in a predefined order that would make the first route segments to receive way more answers and would invalidate the last ones with few to none answers. For this reason, it was decided to randomise the order of the route

segments. As this functionality is not present on the main online survey tools, a custom questionnaire was implemented in a simple ASP.NET web application to present the route segments in a randomized order and guarantee a balanced number of answers.

The first section of the form, shown in figure 21, has the instructions for the respondents and the picture to be assessed. The second section with the close-ended questions is illustrated in figure 22. These questions were included in the questionnaire to help to analyze the cases in which the tool did not predict well the cyclists' score. By knowing which features were identified by the respondents, it becomes possible to investigate if the models did not work properly or if the pipeline structure needs adjustments. The last section, present in figure 23, has the main question of the questionnaire asking about the perception level of comfort and safety in the route segment picture.


**Figure 21.** Header of the survey page.

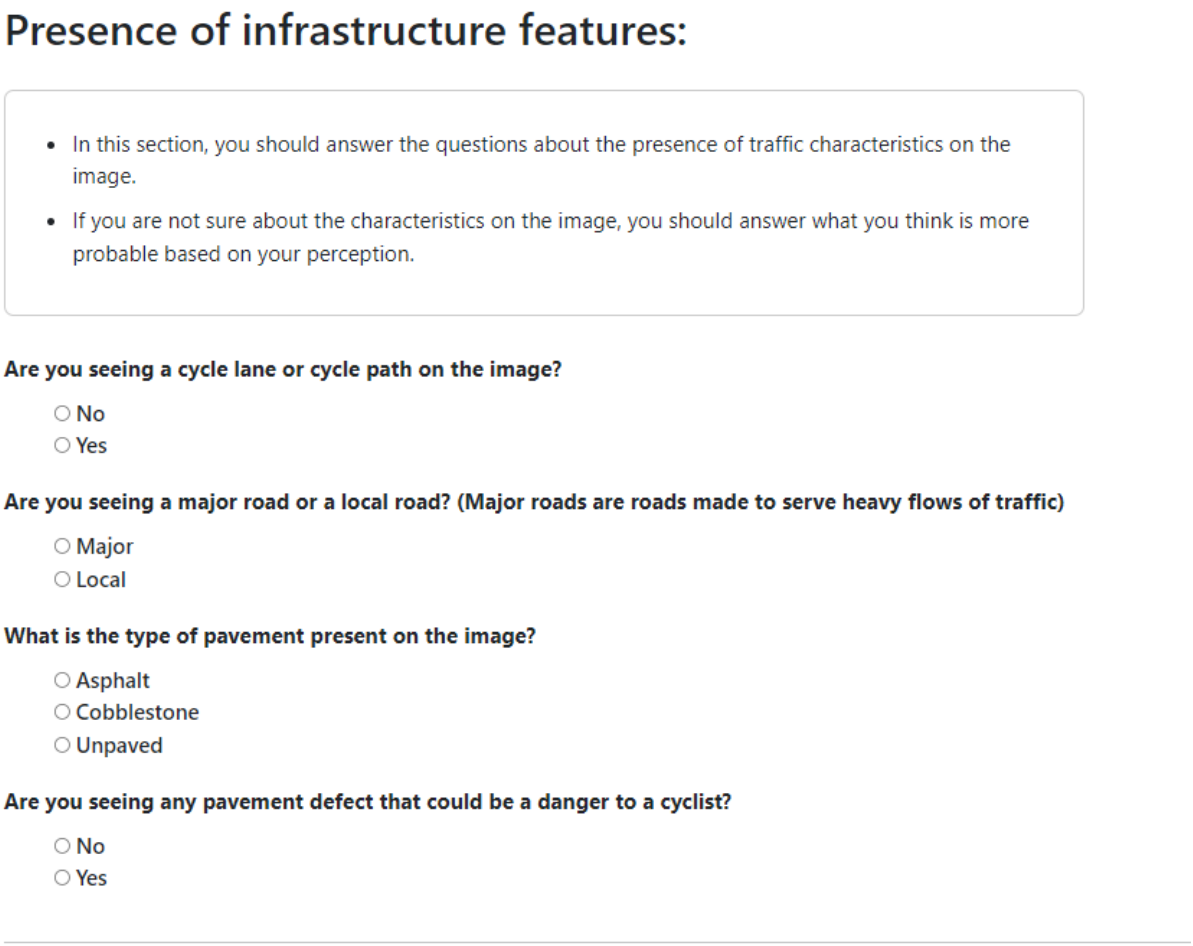


**Figure 22.** Close-ended questions .

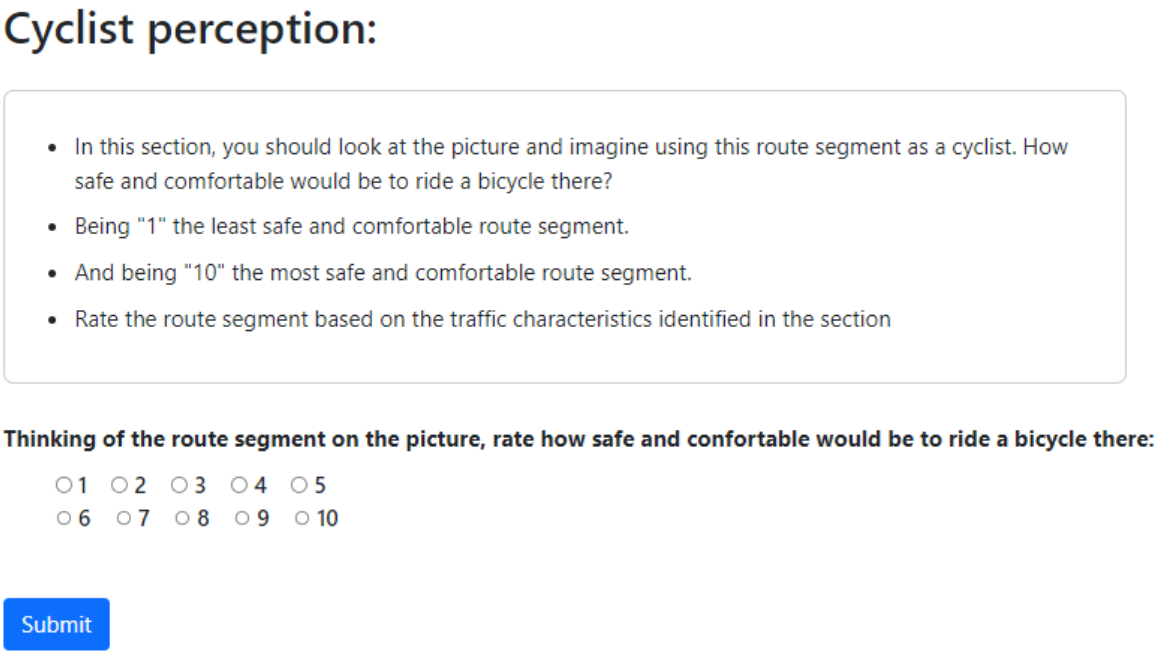


**Figure 23.** Score question.

# 5 Results

The survey received 500 answers for the 48 route segments. The same route segments were processed by the automatic assessment tool. The results of the survey were aligned with the literature, as the cyclists submitted higher scores for segments with pavement without defects, residential streets and the presence of cycle infrastructure. In contrast, the type of pavement did not influence the cyclists as expected. Appendix B has the average score given by the cyclists on the questionnaire, the score calculated by the assessment tool, the error between them and the result of the prediction. The images used in the experiment can be found in appendix A. It is unrealistic an automatic assessment predicting exactly the score from the survey, hence it was considered a correct prediction a difference within a threshold of 2 between the tool and the survey results. Using this threshold, the assessment tool had an accuracy of 67% predicting the score given by the cyclists. The average error between all the segments' scores was 1.93, but the median was 1.69, indicating that most of the segments belonged to the same scale of error. This becomes clearer in figure 24 that displays the error per route segment compared to the median and the threshold.

Exploring some of the segments individually can help to understand the reason of the larger discrepancies. For example, segment 9, shown in figure 35, had a much higher automatic score because the model detected a nonexistent cycle infrastructure and this feature has a heavy effect on the score. Along with this, the picture shows a tram rail, which is a traffic object very relevant to safety perception which was not included in this version of the assessment tool. Segment 30 is another important case to observe because it received an automatic score of 0. For this scoring system, a value of 0 should not be possible because there is no combination of features in real roads that results in such a score. It can be seen in figure 56 that this segment is obviously not a major road but it is rather positioned right above a major road, causing this misclassification when fetching the road network dataset.

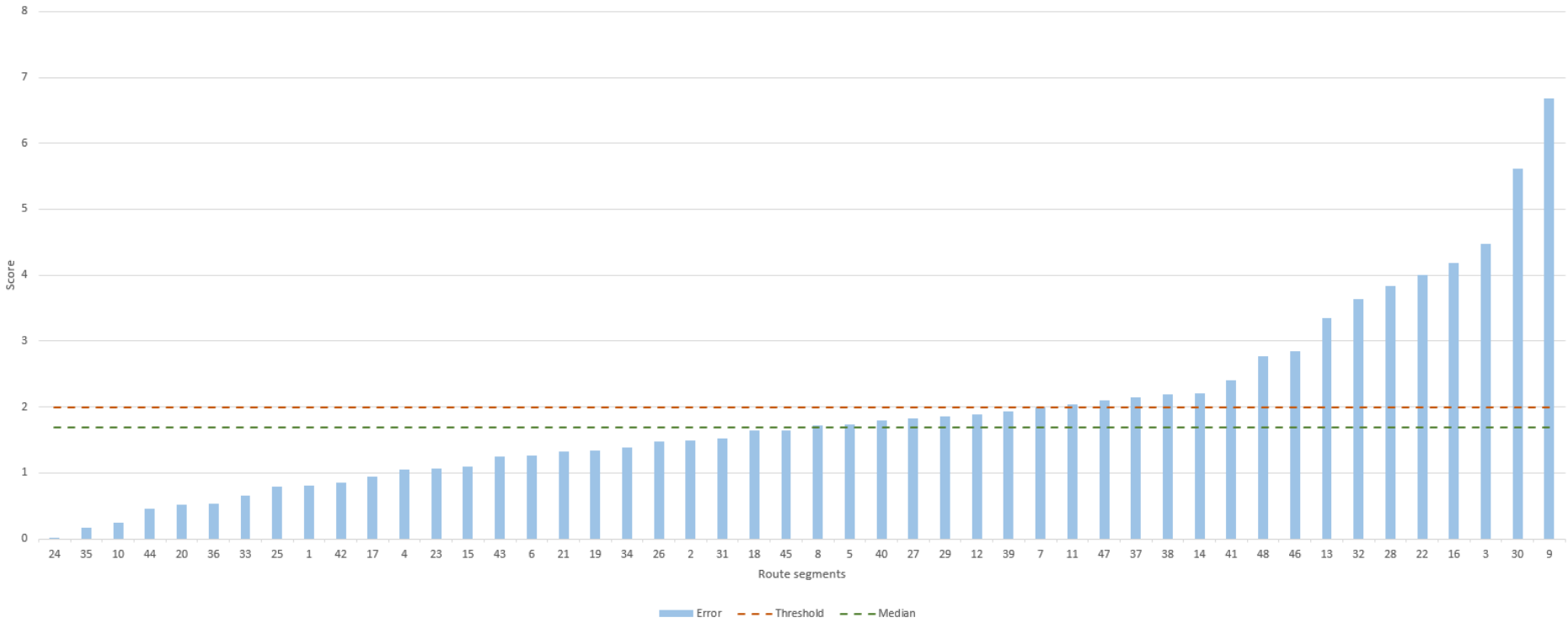


**Figure 24.** Error between automatic assessment and the cyclists' perception per route segment with threshold and median.


**Figure 25.** Segment 9 picture.


**Figure 26.** Segment 30 picture.

In order to evaluate the results, it is necessary to analyze the accuracy of the models individually. Appendix C contains, for each route segment used in the experiment, the ground truth, the model classification and the cyclists' perception for the pavement type. Considering that not all the questionnaire respondents answered the same for the same picture, the *cyclist perception* column is the most answered value, and this definition is the same for the data in the appendices E, D and F. The pavement type model obtained an accuracy of 89% in this dataset, which is in line with the performance reported in section 4.3.1. This result confirms the effectiveness of the strategy of building a new dataset to train the model, ensuring a high accuracy and good generalization. Table 6 shows that the model errors have a pattern of class *asphalt* being misclassified as *cobblestone*, probably because the picture contains part of sidewalk. Some of the respondents answered asphalt in segments that clearly contained cobblestone pavement. This happened in segments in which there was cycle infrastructure within a cobblestone street. Probably the question was not clear enough and lead respondents to answers about the pavement type of the cycle infrastructure instead of the street pavement. However, this did not affect the results of the experiments.

The pavement defects was a more knotty feature to compare with the respondents' perceptions. One of the challenges is the fact that pavement defect detection is a complex computer vision problem to solve. The pavement defects model reached a mAP of 59.4% as reported in section 4.3.1. This performance was higher than the model trained by De Bock and Verstockt (2022) for their assessment pipeline, which reported a mAP of 55%. They

**Table 6.** Pavement type prediction result on the experiment route segments represented in a confusion matrix.

| | | Predicted | | |
|---|---|---|---|---|
| | | Asphalt | Cobblestone | Unpaved |
| Ground truth | Asphalt | 24 | 5 | 0 |
| | Cobblestone | 0 | 10 | 0 |
| | Unpaved | 0 | 0 | 9 |

do not detail any different configurations and probably used the YOLOv5s configuration with default parameters. Certainly, the decision of using YOLOv5l configuration with label smoothing was the key to surpass this performance. The *Road Damage Detection Challenge* leaderboard[3] has a top of the list with even higher performances, which were achieved by creating DNN architectures specific for road damage detection, but it was not the focus of this research and can be explored in future versions. Although this model was not able to reach a high mAP, this does not mean a low performance for its role in this pipeline. This measure includes the validation of the bounding box in a certain IOU threshold and the correct class of the defect, but this is not essential considering the pipeline only needs the presence or not of a defect. This model was trained using the original IOU threshold and classes for the purpose of evaluation and comparison with the reference papers, but it could be improved by being trained with a larger threshold and condensing the classes into one.

The comparison between the respondents' perceptions and the automatic tool predictions on the existence of defects is available in appendix E. Only the segments

[3]https://crddc2022.sekilab.global/leaderboard/

classified as asphalt were considered, because this model only works in asphalt pavement. The confusion matrix in table 7 indicates that most of the misclassification reflects a hypersensitivity of the model. A pavement defect relevant for cyclists is a subjective concept, and for this reason, there is no ground truth column in this table. In this case, the main goal was to understand if the model sensitivity should be adjusted to match the human perceptions. The pavement defects model had an accuracy of 54% predicting the cyclists' perception. This is not a satisfactory performance, but the comparison gives a hint about the main problem. The model detected defects in 12 of the 24 segments, in contrast with the 3 segments pointed out by the respondents. Without a doubt, the model is much more sensible than the human perception. Experiments with higher confidence threshold in the inference or using only certain categories of defects could be a starting point to analyze this, because the RDD2022 dataset was built to allow the detection of every type of defect, even the minor cracks.

**Table 7.** Pavement defects prediction result on the experiment route segments represented in a confusion matrix.

| | | Predicted | |
|---|---|---|---|
| | | TRUE | FALSE |
| Cyclists' perceptions | TRUE | 2 | 1 |
| | FALSE | 10 | 11 |

Like the pavement type model, the cycle infrastructure model also had very good generalization with this dataset, with an accuracy of 87.5% in relation to the ground truth. Appendix D lists the ground truth, the respondents' perceptions and the model's predictions for the cycle infrastructure detection task. Table 8 summarizes the performance of this model. The value *DRAW* in the *Cyclist perception* column means that the answers were 50/50 for this segment. The performance of this model is

particularly impressive this result for this model, considering the limited dataset and the difference between the types of cycling infrastructure in Lisbon. The performance of the cycle infrastructure detection has also a high importance to the assessment tool, for the reason that the presence of these features impacts the score heavily. Certainly, a considerable part of the automatic assessment performance as a whole was positively impacted by this model's performance.

**Table 8.** Cycle infrastructure prediction result on the experiment route segments represented in a confusion matrix.

| | | Predicted | |
|---|---|---|---|
| | | TRUE | FALSE |
| Actual | TRUE | 12 | 2 |
| | FALSE | 4 | 30 |

The road type classification agreed with the cyclists' perception in 74% of cases. For segment 8 the respondents answered 50% for each category, therefore it was not considered in this measurement. This result indicates that using the hierarchy attribute from the road network dataset was a good decision and mostly aligned with the cyclists' perception, however, it has to be improved to be a reliable feature. The confusion matrix in table 9 suggests that the respondents had a clear idea about what is a local road, but they were in doubt about the pictures depicting major roads. Perhaps two categories were not enough to capture the complexity of this feature as so many segment types were not clear. Besides that, assessing only by a picture probably made this feature more subjective to the respondents, prejudicing the resulting accuracy. The perception of local or major roads is mostly determined by the traffic volume, speed and the number of lines, but only the number of lines is evident in a picture and this could explain the wrong classifications of the major roads. The comparison detailed by route segments is contained in appendix

F.

**Table 9.** Road type classification compared with the cyclists' perception on the experiment route segments represented in a confusion matrix.

| | | Cyclists' perceptions | |
|---|---|---|---|
| | | MAJOR | LOCAL |
| Tool classification | MAJOR | 11 | 9 |
| | LOCAL | 3 | 24 |

# 6 Conclusions

The main question of this thesis was if it is possible to automatically predict the cyclists' perceptions of safety and comfort about segments of routes. An approach based on a pipeline and machine learning was proposed, implemented and evaluated. To validate the hypothesis, an automatic assessment tool was built using machine learning to score route segments based on the presence of features that affects the comfort and safety of cyclists. Additionally, a web survey with a questionnaire was conducted to evaluate if these automatic tool predictions were aligned with real cyclists' perceptions. In a score scale between 1 and 10, the average error was 1.93 and 67% of the score predictions were accurate, considering a threshold of 2. Although this is not a result that validates the tool to be used in real situations, it showed a tendency to score the segments in line with the real perceptions. This experiment had promising results even though it only used four features and had improvised weighting values for the features. Some segments obtained a too lower score from the cyclists because they contain elements that cannot be identified currently by the tool. For example, tram rails are naturally dangerous for cyclists and lines exclusively for buses on the right side, forcing the cyclists to be between the cars and the buses. Another impact was the hypersensitivity of the pavement defects detection model, which detected defects that did not represent a danger in the cyclists' perception. In addition to that, some unbalance was expected between the weights used and the real importance given by the cyclists, for the reason that the weights were adapted from a survey made in a city with different characteristics. In the future, an incremented version of this tool including detection of the missing elements and parameterized by local specialists certainly would achieve more accurate results and could be used to suggest routes for cyclists and give insights for urban planners.

The combination of YOLOv5, Roboflow and PyTorch was a solid choice to prepare the datasets, train the models and integrate the inferences into an application. The cycle infrastructure model had an mAP of 0.941, demonstrating an outstanding performance, taking into account the challenge of learning multiple types of cycle infrastructures with a limited dataset. Likewise, the pavement type model also achieved very good results with a classification accuracy of 0.988. The pavement defects model did not reach the same level of performance and will need an improved approach as an mAP of 0.594 is not ideal for real world applications, but this was expected as this is a complex problem

still being explored by researchers. YOLOv5 models were relatively straightforward to set up, parameterize and train. The classification mode of YOLOv5 was released some months before this research was written, and even though the trained model brought good results, it is missing essential features for being a work in progress. The evaluation script is missing metrics such as confusion matrix, precision and recall, which was an obstacle when analyzing the model.

The approach to create the new datasets was effective for the size of this work, but needs to be improved in order to scale the solution. The Google Street View Static API saved a considerable amount of time and effort in this task, making it possible to capture the images without being physically in the locations, but the manual capture creates a limitation when adding other locations and features. The future work section discusses a suggestion to automate image capture. Annotation is also a relevant task when discussing scale, and even though the cycle infrastructure dataset was annotated manually, the Roboflow annotation tool made this task as easy and productive as possible. These datasets will be publicly available so they can be used and extended in other research.

The contributions of this thesis are as follows.

**Approach automatic for route safety and comfort assessment** This work proposed and evaluated a new approach for automating the route assessment for cyclists. The successful strategies for the different tasks were described and can be used to guide future research.

**Dataset of pavement type for ML model training** A dataset made of pictures of Portuguese roads focusing on the pavement. The pictures are separated into folders by the type of pavement in the YOLOv5 classification format. It can be accessed in the following GitHub repository `https://github.com/alannunescaetano/pavement_type_dataset`.

**Dataset of cyclist infrastructure for ML model training** A dataset very similar to the pavement type dataset, but having more pictures of roads with cycle paths and cycle lanes of different types in Lisbon. It also has these cycle infrastructures annotated for object detection. It is hosted publicly in the Roboflow platform to be easier to export in the desired format: `https://universe.roboflow.com/thesis-yuxwe/cycle-path-dataset`.

The work reported in this dissertation was also used as the basis for one scientific article accepted for presentation and publication in the proceedings of the 31st International Conference on Information Systems Development (ISD 2023) conference (Caetano, Estima, & Lima, 2023).

## 6.1 Limitations

This work did not define which traffic factors impact the cyclists' perception of safety and comfort in Portugal and the respective weight for each factor. There are researches describing factors not related to traffic in Portugal, socioeconomic factors for example, which are not useful in analyzing routes. Although these values have some degree of similarity in most cities, some of the factors can be more or less important for the cyclists depending on the topography, weather and urban characteristics of the location. This represents the main limitation of this work.

The results also showed that some segments had a higher difference between the cyclists' perceptions and the automatic score due to the presence of features that are not being mapped by the assessment tool. Tram rails, for example, are not present in most of the segments, but they affect heavily the cyclists' perceptions when they exist. Other examples are the width of the road, number of lines and presence of bus line on the right side. The slope is an important feature for a city like Lisbon, but it was not included in this work because it was not straightforward to represent it clearly in the questionnaire or in the picture. Certainly, implementing more features will allow the tool to achieve a higher accuracy.

In this first version of the assessment tool, the route segment picture had to be manually extracted to be assessed. Unfortunately, it is not possible to extract a picture from Google Street View Static API compatible with the one in the assessment tool based only on the coordinates, because there is a limitation in the API in defining the position of the camera related to the car. This means that the picture cannot be extracted automatically capturing the street up front. This experiment used only 48 pictures, hence it was not difficult to extract them manually, but this would represent a limitation to assess an entire city.

## 6.2 Future work

This thesis evaluated a new approach to cyclist route assessment and described the identified limitations. Future work could investigate these issues in order to advance the tool to a level in which it can have a practical use. Probably the most important future work is to conduct a study with experts to define real feature weights for the target city and run a new experiment. This is a way to also test the assumption that the weights vary according to the location. Additionally, adding new features would make the scoring system more complete and could improve the accuracy of the tool. Tram rails, bus lines, traffic lights and signalization are features relevant to cyclists and possible to be detected with machine learning. The slope also has a high impact on the route choice in cities like Lisbon and can be extracted from the road network datasets. The pavement type feature can be improved by adding new categories, for example dividing the asphalt category into smooth and rough.

Improvements from the point of view of usability are also important to prepare this tool for real use. The limitation of the Google Street View Static API on extracting the pictures heading in the correct direction probably could be solved by using the Google MAPS API[4] or another GIS platform. In a GIS platform, a road is represented by a sequence of points. The difference in latitude and longitude between the desired point and the nearest point of the road can be used to calculate the direction of the road and thus, the direction in which the car is moving. This would reveal the cardinal directions of the front and the back of the car, and consequently, the horizontal angle to capture the route segment in the desired position. With this information is possible to the tool to receive only the coordinates as parameters, automatically retrieve the picture from the API and calculate the score. This would open the possibility of assessing an entire city easily.

Finally, the assessment tool would benefit from improvements in the models' performances. During the development of this research, new versions of YOLO were released reporting better performances than YOLOv5. All three models could be retrained in a new version of YOLO or another deep learning framework to evaluate if the performance rises. At the same time, the pavement type and the cycle infrastructure dataset could be expanded with more images to be able to create models with a

[4]https://developers.google.com/maps/

higher generalization. The pavement defects model specifically represents a challenge to be improved to achieve the same level of performance as the other models. The starting point would be testing in a newer YOLO version. This could represent a little improvement but certainly would not be enough to reach a performance near the SOTA solutions. The next step would be to look for SOTA architectures at the top of the *Road Damage challenge* scoreboard. Furthermore, the evaluation could be different in new experiments, since the use of this model in this tool is simpler. The RDD dataset has multiple classes of defects and their bounding boxes. Firstly, the prediction of the bounding box is not important in this context, therefore the IOU could have a higher threshold during training. Secondly, the results showed that the cyclists did not consider most types of defects as dangerous, meaning that some classes could be excluded from training and the remaining classes could be grouped since the class prediction is not important for the score. With these adjustments, the pavement defects model training would focus on the presence of the relevant defects instead of the position or class, and thus, would probably achieve a better performance predicting the pavement defect feature of the assessment tool.

# References


Arellana, J., Saltarín, M., Larrañaga, A. M., González, V. I., & Henao, C. A. (2020). Developing an urban bikeability index for different types of cyclists as a tool to prioritise bicycle infrastructure investments. *Transportation Research Part A: Policy and Practice*, *139*, 310–334.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Mraz, A., Kashiyama, T., & Sekimoto, Y. (2020). Transfer learning-based road damage detection for multiple countries. *arXiv preprint arXiv:2008.13101*.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., & Sekimoto, Y. (2022). Rdd2022: A multi-national image dataset for automatic road damage detection. *arXiv preprint arXiv:2209.08538*.

Bassett, D. R., Pucher, J., Buehler, R., Thompson, D. L., & Crouter, S. E. (2008). Walking, cycling, and obesity rates in europe, north america, and australia. *Journal of physical activity and health*, *5*(6), 795–814.

Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Vol. 1). MIT press Cambridge, MA, USA.

Bradski, G., & Kaehler, A. (2008). *Learning opencv: Computer vision with the opencv library.* " O'Reilly Media, Inc.".

Caetano, A., Estima, J., & Lima, E. (2023). Cyclist route assessment with computer vision. In *Information systems development, organizational aspects and societal trends (isd2023 proceedings).*

Cafiso, S., Pappalardo, G., & Stamatiadis, N. (2021). Observed risk and user perception of road infrastructure safety assessment for cycling mobility. *Infrastructures*, *6*(11), 154.

Cavill, N., Kahlmeier, S., Rutter, H., Racioppi, F., & Oja, P. (2008). *Methodological guidance on the economic appraisal of health effects related to walking and cycling: summary: economic assessment of transport infrastructure and policies* (Tech. Rep.). World Health Organization. Regional Office for Europe.

De Bock, J., & Verstockt, S. (2022). Road cycling safety scoring based on geospatial analysis, computer vision and machine learning. *Multimedia Tools and Applications*, 1–22.

Forsyth, D., & Ponce, J. (2011). *Computer vision: A modern approach.* Prentice hall.

Google. (2023). *Street View Static API overview.* https://developers.google.com/maps/documentation/streetview/request-streetview/. ([Online; Accessed April 08, 2023])

Gullón, P., Badland, H. M., Alfayate, S., Bilal, U., Escobar, F., Cebrecos, A., ... Franco, M. (2015). Assessing walking and cycling environments in the streets of madrid: comparing on-field and virtual audits. *Journal of urban health*, *92*(5), 923–939.

Gupta, P., & Dixit, M. (2022). Image-based crack detection approaches: a comprehensive survey. *Multimedia Tools and Applications*, 1–49.

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of yolo algorithm developments. *Procedia Computer Science*, *199*, 1066–1073.

Jocher, G., Nishimura, K., Mineeva, T., & Vilariño, R. (2020). yolov5. *Code repository*.

Khan, A. A., Laghari, A. A., & Awan, S. A. (2021). Machine learning in computer vision: A review. *EAI Endorsed Transactions on Scalable Information Systems*, *8*(32), e4–e4.

Manton, R., Rau, H., Fahy, F., Sheahan, J., & Clifford, E. (2016). Using mental mapping to unpack perceived cycling risk. *Accident Analysis & Prevention*, *88*, 138–149.

Nolte, M., Kister, N., & Maurer, M. (2018). Assessment of deep convolutional neural networks for road surface classification. In *2018 21st international conference on intelligent transportation systems (itsc)* (pp. 381–386).

Padilla, R., Netto, S. L., & Da Silva, E. A. (2020). A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (iwssip)* (pp. 237–242).

Parker, J. R. (2010). *Algorithms for image processing and computer vision.* John Wiley & Sons.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... others (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pisco, V. G., & Marques-Neto, H. T. (2021). iwalk: Uma solução para medição e análise da caminhabilidade de cidades com portais de dados abertos. In *Anais do v workshop*

*de computação urbana* (pp. 84–97).

Pucher, J., Dill, J., & Handy, S. (2010). Infrastructure, programs, and policies to increase bicycling: an international review. *Preventive medicine*, *50*, S106–S125.

Rabl, A., & De Nazelle, A. (2012). Benefits of shift from car to active transport. *Transport policy*, *19*(1), 121–131.

Raschka, S. (2015). *Python machine learning*. Packt publishing ltd.

Roboflow, I. (2023). *Roboflow*. `https://app.roboflow.com/`.

Saxton, T. (2022). Mapping suburban bicycle lanes using street scene images and deep learning. *arXiv preprint arXiv:2204.12701*.

Seabra, M., Pinheiro, A., Marcelino, C., Santos, D., Leitão, J., & Rodrigues, A. (2011). *Rede viária-princípios de planeamento e desenho*. Instituto da Mobilidade e dos Transportes Terrestres, IP (IMTT).

Stinson, M. A., & Bhat, C. R. (2003). Commuter bicyclist route choice: Analysis using a stated preference survey. *Transportation research record*, *1828*(1), 107–115.

Winters, M., & Teschke, K. (2010). Route preferences among adults in the near market for bicycling: findings of the cycling in cities study. *American journal of health promotion*, *25*(1), 40–47.

Zhao, L., Wu, Y., Luo, X., & Yuan, Y. (2022). Automatic defect detection of pavement diseases. *Remote Sensing*, *14*(19), 4836.

Zhao, T., & Wei, Y. (2022). A road surface image dataset with detailed annotations for driving assistance applications. *Data in brief*, *43*, 108483.

# Appendices

## A Segment route pictures used in the experiment to compare the questionnaire answers and the assessment tool


**Figure 27.** Segment 1 picture.


**Figure 28.** Segment 2 picture.


**Figure 29.** Segment 3 picture.


**Figure 30.** Segment 4 picture.

**Figure 31.** Segment 5 picture.

**Figure 32.** Segment 6 picture.

**Figure 33.** Segment 7 picture.

**Figure 34.** Segment 8 picture.

**Figure 35.** Segment 9 picture.

**Figure 36.** Segment 10 picture.


**Figure 37.** Segment 11 picture.


**Figure 38.** Segment 12 picture.


**Figure 39.** Segment 13 picture.


**Figure 40.** Segment 14 picture.


**Figure 41.** Segment 15 picture.


**Figure 42.** Segment 16 picture.


**Figure 43.** Segment 17 picture.


**Figure 44.** Segment 18 picture.


**Figure 45.** Segment 19 picture.


**Figure 46.** Segment 20 picture.


**Figure 47.** Segment 21 picture.


**Figure 48.** Segment 22 picture.

**Figure 49.** Segment 23 picture.

**Figure 50.** Segment 24 picture.

**Figure 51.** Segment 25 picture.

**Figure 52.** Segment 26 picture.

**Figure 53.** Segment 27 picture.

**Figure 54.** Segment 28 picture.


**Figure 55.** Segment 29 picture.


**Figure 56.** Segment 30 picture.


**Figure 57.** Segment 31 picture.


**Figure 58.** Segment 32 picture.


**Figure 59.** Segment 33 picture.


**Figure 60.** Segment 34 picture.


**Figure 61.** Segment 35 picture.


**Figure 62.** Segment 36 picture.


**Figure 63.** Segment 37 picture.


**Figure 64.** Segment 38 picture.


**Figure 65.** Segment 39 picture.


**Figure 66.** Segment 40 picture.

**Figure 67.** Segment 41 picture.

**Figure 68.** Segment 42 picture.

**Figure 69.** Segment 43 picture.

**Figure 70.** Segment 44 picture.

**Figure 71.** Segment 45 picture.

**Figure 72.** Segment 46 picture.


**Figure 73.** Segment 47 picture.


**Figure 74.** Segment 48 picture.

# B Score values from the questionnaire and the assessment tool results

**Table 10.** Comparison between the scores from the questionnaire and the automatic assessment, error and result considering a threshold of 2.

| Route segment | Cyclists score | Automatic score | Error | Predicted (T=2) |
|---|---|---|---|---|
| 1 | 7.15 | 6.35 | 0.8 | TRUE |
| 2 | 3.67 | 5.16 | 1.49 | TRUE |
| 3 | 7.38 | 2.9 | 4.48 | FALSE |
| 4 | 4.86 | 5.91 | 1.05 | TRUE |
| 5 | 4.75 | 3.01 | 1.74 | TRUE |
| 6 | 4.64 | 5.91 | 1.27 | TRUE |
| 7 | 4.25 | 2.26 | 1.99 | TRUE |
| 8 | 5.38 | 7.1 | 1.72 | TRUE |
| 9 | 2.57 | 9.25 | 6.68 | FALSE |
| 10 | 4.91 | 5.16 | 0.25 | TRUE |
| 11 | 4.3 | 2.26 | 2.04 | FALSE |
| 12 | 4.14 | 2.26 | 1.88 | TRUE |
| 13 | 6.25 | 2.9 | 3.35 | FALSE |
| 14 | 3.71 | 5.91 | 2.2 | FALSE |
| 15 | 4 | 2.9 | 1.1 | TRUE |
| 16 | 2.92 | 7.1 | 4.18 | FALSE |
| 17 | 6.11 | 5.16 | 0.95 | TRUE |
| 18 | 8.75 | 7.1 | 1.65 | TRUE |
| 19 | 7.91 | 9.25 | 1.34 | TRUE |
| 20 | 2.5 | 3.01 | 0.51 | TRUE |
| 21 | 5.77 | 7.1 | 1.33 | TRUE |
| 22 | 6 | 10 | 4 | FALSE |
| 23 | 4.09 | 5.16 | 1.07 | TRUE |
| 24 | 5.17 | 5.16 | 0.01 | TRUE |

| | | | | |
|---|---|---|---|---|
| 25 | 5.56 | 6.35 | 0.79 | TRUE |
| 26 | 3.73 | 2.26 | 1.47 | TRUE |
| 27 | 4.09 | 2.26 | 1.83 | TRUE |
| 28 | 6.73 | 2.9 | 3.83 | FALSE |
| 29 | 4.75 | 2.9 | 1.85 | TRUE |
| 30 | 5.62 | 0 | 5.62 | FALSE |
| 31 | 7.73 | 9.25 | 1.52 | TRUE |
| 32 | 6.36 | 10 | 3.64 | FALSE |
| 33 | 4.5 | 5.16 | 0.66 | TRUE |
| 34 | 3.78 | 5.16 | 1.38 | TRUE |
| 35 | 5 | 5.16 | 0.16 | TRUE |
| 36 | 4.63 | 5.16 | 0.53 | TRUE |
| 37 | 4.4 | 2.26 | 2.14 | FALSE |
| 38 | 5.2 | 3.01 | 2.19 | FALSE |
| 39 | 4.2 | 2.26 | 1.94 | TRUE |
| 40 | 8.14 | 6.35 | 1.79 | TRUE |
| 41 | 4.67 | 2.26 | 2.41 | FALSE |
| 42 | 8.4 | 9.25 | 0.85 | TRUE |
| 43 | 8 | 9.25 | 1.25 | TRUE |
| 44 | 6.8 | 6.35 | 0.45 | TRUE |
| 45 | 8 | 6.35 | 1.65 | TRUE |
| 46 | 5.75 | 2.9 | 2.85 | FALSE |
| 47 | 5 | 2.9 | 2.1 | FALSE |
| 48 | 5.67 | 2.9 | 2.77 | FALSE |

# C Results for pavement type classification task

**Table 11.** Comparison between the model's pavement type prediction and the questionnaire answers.

| Route segment | Ground truth | Cyclists perception | Model classification |
|---|---|---|---|
| 1 | asphalt | asphalt | asphalt |
| 2 | asphalt | asphalt | asphalt |
| 3 | unpaved | unpaved | unpaved |
| 4 | asphalt | asphalt | asphalt |
| 5 | asphalt | asphalt | asphalt |
| 6 | asphalt | asphalt | asphalt |
| 7 | cobblestone | cobblestone | cobblestone |
| 8 | asphalt | asphalt | asphalt |
| 9 | asphalt | asphalt | cobblestone |
| 10 | cobblestone | cobblestone | cobblestone |
| 11 | cobblestone | cobblestone | cobblestone |
| 12 | asphalt | asphalt | asphalt |
| 13 | unpaved | unpaved | unpaved |
| 14 | asphalt | asphalt | asphalt |
| 15 | unpaved | unpaved | unpaved |
| 16 | asphalt | asphalt | asphalt |
| 17 | asphalt | asphalt | asphalt |
| 18 | asphalt | asphalt | asphalt |
| 19 | cobblestone | asphalt | cobblestone |
| 20 | asphalt | asphalt | asphalt |
| 21 | asphalt | asphalt | asphalt |
| 22 | asphalt | asphalt | asphalt |
| 23 | cobblestone | cobblestone | cobblestone |
| 24 | cobblestone | cobblestone | cobblestone |
| 25 | asphalt | asphalt | asphalt |
| 26 | asphalt | asphalt | asphalt |

| | | | |
|---|---|---|---|
| 27 | asphalt | asphalt | cobblestone |
| 28 | unpaved | unpaved | unpaved |
| 29 | unpaved | unpaved | unpaved |
| 30 | unpaved | unpaved | unpaved |
| 31 | cobblestone | asphalt | cobblestone |
| 32 | asphalt | asphalt | asphalt |
| 33 | cobblestone | cobblestone | cobblestone |
| 34 | asphalt | asphalt | asphalt |
| 35 | asphalt | asphalt | cobblestone |
| 36 | asphalt | asphalt | asphalt |
| 37 | asphalt | asphalt | cobblestone |
| 38 | asphalt | asphalt | asphalt |
| 39 | asphalt | asphalt | cobblestone |
| 40 | asphalt | asphalt | asphalt |
| 41 | cobblestone | cobblestone | cobblestone |
| 42 | asphalt | asphalt | asphalt |
| 43 | cobblestone | cobblestone | cobblestone |
| 44 | asphalt | asphalt | asphalt |
| 45 | asphalt | asphalt | asphalt |
| 46 | unpaved | unpaved | unpaved |
| 47 | unpaved | unpaved | unpaved |
| 48 | unpaved | unpaved | unpaved |

# D Results for the cycle infrastructure detection task

**Table 12.** Comparison between the model's cycle infrastructure prediction and the questionnaire answers.

| Route segment | Ground truth | Cyclists perception | Model detection |
|---|---|---|---|
| 1 | TRUE | TRUE | TRUE |
| 2 | FALSE | FALSE | FALSE |
| 3 | FALSE | TRUE | FALSE |
| 4 | FALSE | FALSE | FALSE |
| 5 | FALSE | FALSE | FALSE |
| 6 | FALSE | FALSE | FALSE |
| 7 | FALSE | FALSE | FALSE |
| 8 | FALSE | DRAW | TRUE |
| 9 | FALSE | FALSE | TRUE |
| 10 | FALSE | FALSE | FALSE |
| 11 | FALSE | FALSE | FALSE |
| 12 | FALSE | FALSE | FALSE |
| 13 | FALSE | FALSE | FALSE |
| 14 | FALSE | FALSE | FALSE |
| 15 | FALSE | FALSE | FALSE |
| 16 | FALSE | FALSE | TRUE |
| 17 | TRUE | TRUE | FALSE |
| 18 | TRUE | TRUE | TRUE |
| 19 | TRUE | TRUE | TRUE |
| 20 | FALSE | FALSE | FALSE |
| 21 | TRUE | TRUE | TRUE |
| 22 | TRUE | TRUE | TRUE |
| 23 | FALSE | FALSE | FALSE |
| 24 | FALSE | FALSE | FALSE |
| 25 | FALSE | TRUE | TRUE |
| 26 | TRUE | FALSE | FALSE |

| | | | |
|---|---|---|---|
| 27 | FALSE | FALSE | FALSE |
| 28 | FALSE | TRUE | FALSE |
| 29 | FALSE | FALSE | FALSE |
| 30 | FALSE | FALSE | FALSE |
| 31 | TRUE | TRUE | TRUE |
| 32 | TRUE | DRAW | TRUE |
| 33 | FALSE | FALSE | FALSE |
| 34 | FALSE | FALSE | FALSE |
| 35 | FALSE | FALSE | FALSE |
| 36 | FALSE | FALSE | FALSE |
| 37 | FALSE | FALSE | FALSE |
| 38 | FALSE | FALSE | FALSE |
| 39 | FALSE | FALSE | FALSE |
| 40 | TRUE | TRUE | TRUE |
| 41 | FALSE | FALSE | FALSE |
| 42 | TRUE | TRUE | TRUE |
| 43 | TRUE | TRUE | TRUE |
| 44 | TRUE | TRUE | TRUE |
| 45 | TRUE | TRUE | TRUE |
| 46 | FALSE | FALSE | FALSE |
| 47 | FALSE | FALSE | FALSE |
| 48 | FALSE | TRUE | FALSE |

# E Results for the pavement defects detection task

**Table 13.** Comparison between the model's pavement defects prediction and the questionnaire answers.

| Route segment | Cyclists perception | Model detection |
|---|---|---|
| 1 | FALSE | TRUE |
| 2 | FALSE | TRUE |
| 4 | FALSE | FALSE |
| 5 | FALSE | FALSE |
| 6 | FALSE | FALSE |
| 8 | FALSE | FALSE |
| 12 | FALSE | TRUE |
| 14 | FALSE | FALSE |
| 16 | FALSE | FALSE |
| 17 | FALSE | TRUE |
| 18 | FALSE | FALSE |
| 20 | FALSE | FALSE |
| 21 | FALSE | FALSE |
| 22 | FALSE | FALSE |
| 25 | FALSE | TRUE |
| 26 | FALSE | TRUE |
| 32 | FALSE | FALSE |
| 34 | TRUE | TRUE |
| 36 | TRUE | TRUE |
| 38 | TRUE | FALSE |
| 40 | FALSE | TRUE |
| 42 | FALSE | TRUE |
| 44 | FALSE | TRUE |
| 45 | FALSE | TRUE |

# F Results for road type retrieval task

**Table 14.** Comparison between the retrieved road type and the questionnaire answers.

| Route segment | Cyclists perception | Tool classification |
|---|---|---|
| 1 | LOCAL | MAJOR |
| 2 | MAJOR | LOCAL |
| 3 | LOCAL | LOCAL |
| 4 | LOCAL | LOCAL |
| 5 | MAJOR | MAJOR |
| 6 | LOCAL | LOCAL |
| 7 | LOCAL | MAJOR |
| 8 | DRAW | MAJOR |
| 9 | LOCAL | LOCAL |
| 10 | LOCAL | LOCAL |
| 11 | MAJOR | MAJOR |
| 12 | MAJOR | MAJOR |
| 13 | LOCAL | LOCAL |
| 14 | MAJOR | LOCAL |
| 15 | LOCAL | LOCAL |
| 16 | MAJOR | MAJOR |
| 17 | LOCAL | LOCAL |
| 18 | MAJOR | MAJOR |
| 19 | LOCAL | LOCAL |
| 20 | MAJOR | MAJOR |
| 21 | LOCAL | MAJOR |
| 22 | MAJOR | LOCAL |
| 23 | LOCAL | LOCAL |
| 24 | LOCAL | LOCAL |
| 25 | MAJOR | MAJOR |
| 26 | MAJOR | MAJOR |
| 27 | LOCAL | MAJOR |

| | | |
|---|---|---|
| 28 | LOCAL | LOCAL |
| 29 | LOCAL | LOCAL |
| 30 | LOCAL | MAJOR |
| 31 | LOCAL | LOCAL |
| 32 | LOCAL | LOCAL |
| 33 | LOCAL | LOCAL |
| 34 | LOCAL | LOCAL |
| 35 | LOCAL | LOCAL |
| 36 | LOCAL | LOCAL |
| 37 | LOCAL | MAJOR |
| 38 | LOCAL | MAJOR |
| 39 | LOCAL | MAJOR |
| 40 | MAJOR | MAJOR |
| 41 | LOCAL | MAJOR |
| 42 | LOCAL | LOCAL |
| 43 | LOCAL | LOCAL |
| 44 | MAJOR | MAJOR |
| 45 | MAJOR | MAJOR |
| 46 | LOCAL | LOCAL |
| 47 | LOCAL | LOCAL |
| 48 | LOCAL | LOCAL |