Chapter 6

Results and Discussion

This section discusses the results from training of the YOLOv5 model on the COCO 2017 annotations and manual annotations, the crowdsourcing on Atlas, as well as the modeling of the obtained accessibility scores.

6.1 YOLOv5 Model Trained with COCO 2017 Annotations and Manual Annotations

When evaluating object detection models like YOLOv5, metrics such as precision, recall, and the mean average precision (mAP) are used as a basis to measure how accurate the model is in its detection (Gad, 2021). Precision measures the model's accuracy in classifying a sample as positive, while recall measures the positive samples the model correctly classified as positive. The mean average precision (mAP) is a way to summarize the precision-recall curve into a single value representing the mean average of all precision values. It returns a prediction score based on how accurate the detected bounding box is compared to the ground-truth bounding box. For YOLOv5, the mAP is measured through intersection over union (IoU) thresholds of 0.5 and 0.5:0.95. The IoU is a quantitative measure to score how the ground-truth and predicted boxes match, and helps to know if a region has an object or not.

Training the YOLOv5 model using YOLOv5x (the largest and most accurate model of YOLOv5) with 300 epochs and a batch size of 64 took 39.770 hours to complete. In Figure 6.1, we can see the graph of the precision, recall, and mean average precision training results. The values for precision, recall, mAP@0.5, and mAP@0.5:0.95 peaked at 0.6665, 0.5447, 0.5359, and 0.3300 respectively. It can be seen in the graphs that the precision continued to increase with the higher number of epochs, while the recall, mAP@0.5, and mAP@0.5:0.95 all peaked at around 60 to 70 epochs before decreasing as the number of epochs reached 300. The model was overfitting early and we could have reduced the number of epochs during training. However, we were no longer able to train the model again to see its results with a lower number of epochs. We only borrowed the resources from the cloud server of College of Computer Studies(CCS) Jupyterhub as mentioned in 4.7.1, and it is a limitation of our study to train the model only once as other users also had to utilize the resources of the cloud server. We also attempted to train the model again on local machines, however, without a powerful graphics processing unit (GPU), it would take days to train the same model.

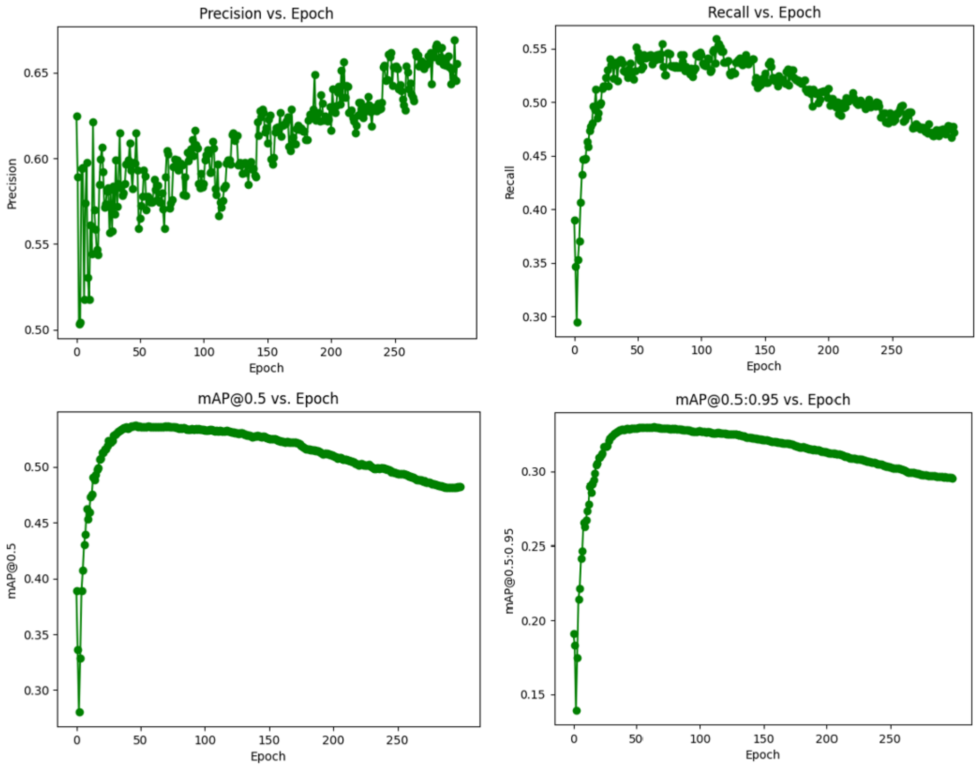


Figure 6.1: Precision, Recall, and Mean Average Precision (mAP) results of the YOLOv5 model

The main metric used to measure YOLOv5's performance is its mAP. After the training was completed, YOLOv5 generated the best training weights, and we tested its performance on our test set and validation set as seen in Figure 6.2. For the test set, it scored 0.524 mean average precision (mAP)@0.5 on all classes. The highest average precision (AP) scores were on objects such as *fire hydrant* and *stop sign*, with 0.882 and 0.865 respectively. This means that the model had lower false positives and lower false negatives for these two objects, resulting to more chances of correct predictions when seen on the streetview images. On the other hand, it scored the lowest on objects such as *cracked pavement*, *curb ramp*, and *construction materials* with 0.073, 0.106, and 0.187 respectively. This means that the model had higher false positives and higher false negatives for these three objects, resulting to more mislabels when seen on the streetview images.

Additionally, the best training weights scored 0.534 mAP@0.5 on all classes for the validation set. The highest AP scores were on objects such as *fire hydrant* and *stop sign*, with 0.866 and 0.860 respectively. This means that the model had lower false positives and lower false negatives for these two objects, resulting to more correct labels when seen on streetview images. On the contrary, it scored the lowest on objects such as *cracked pavement*, *curb ramp*, and *construction materials* with 0.061, 0.136, and 0.225 respectively. This means that the model had higher false positives and higher false negatives for these three objects, resulting to more mislabels when seen on streetview images.

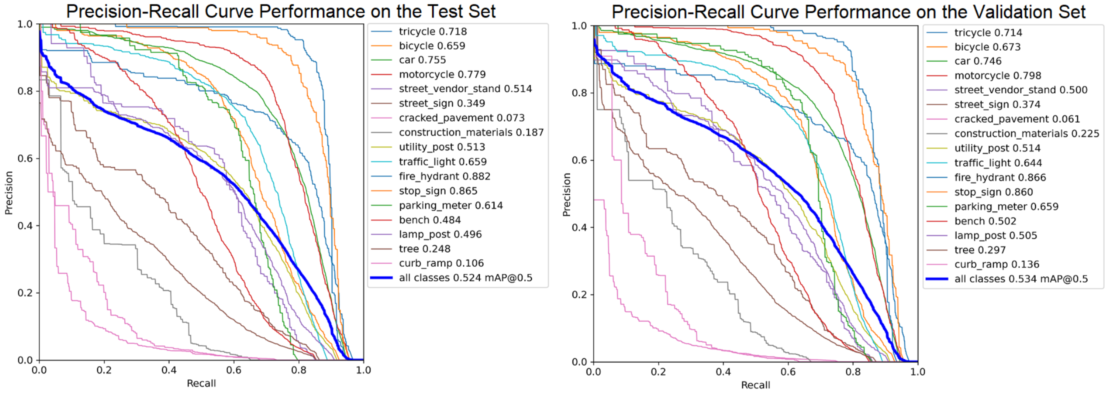


Figure 6.2: Precision-Recall curve performance of the best training weights on the test set and validation set

For the pretrained checkpoints of YOLOv5x on their main GitHub repository (Jocher, 2021b), its benchmark mAP score for the mAP test 0.5:0.95, mAP val 0.5:0.95, and mAP val 0.5 are: 50.4, 50.4, and 68.8 respectively. Comparing it to the performance of our best training weights on the test and validation sets, our trained model only had a score of: 0.324, 0.329, and 0.534 respectively. The performance scores of our trained model were not as high as the benchmark scores of the YOLOv5x trained on the COCO dataset.

For the performance of the best training weights on the test and validation sets, we noticed a trend on the mAP@0.5 scores of the classes seen in Figure 6.2. On both the test set and validation set, the classes of: *fire hydrant* and *stop sign* had scores above 0.8. Conversely, the classes of: *curb ramp* and *cracked pavement* had scores below or near to 0.1. One of the recommended tips for best training results according to the main author of YOLOv5 would be having more than 10,000 number of instances per class (Jocher, 2021c). We initially thought this would be one of the main factors that would influence the YOLOv5 model being able to properly learn the class or object. However, the number of instances we had for *fire hydrant*, *stop sign*, *curb ramp*, and cracked pavement were: 1,970, 2,061, 1,130, and 2,609 respectively as seen in Table 4.5. All the four objects were less than the 10,000 number of instances that was recommended by the author of YOLOv5.

We then decided to look at the annotations of the objects and noticed that the *fire hydrant* and *stop sign* objects both had their images and annotations from the COCO 2017 dataset, while *curb ramp* and *cracked pavement* were both from the street view images and annotated by our volunteers. Objects that we filtered from the COCO 2017 Dataset all had images obtained from the internet with a corresponding XML file that contains the bounding box annotation around the object. In Figure 6.3, we can see a sample image of a stop sign and fire hydrant obtained from our train set.

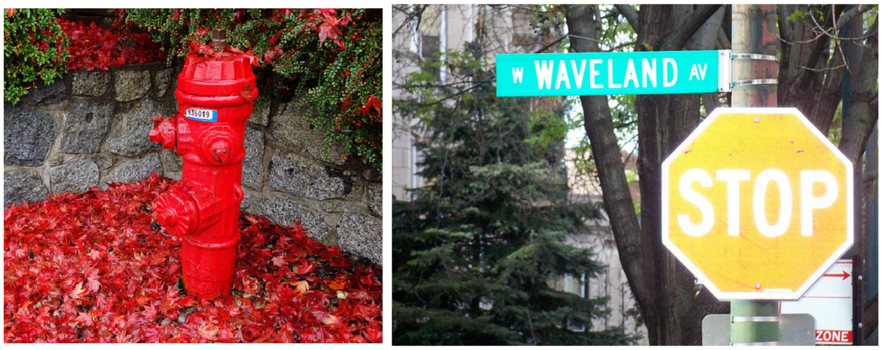


Figure 6.3: Sample image of stop sign and fire hydrant from our train set

On the other hand, looking at the images annotated by our volunteers for *cracked pavement* in Figure 6.4, we can notice inconsistencies on the cracks of sidewalk pavements. The pavement could either include small pieces of cement or line cracks on the surface of the sidewalk. The lighting and color of the pavement also differ on both images and could be a possible reason as to why our YOLOv5 model had a difficult task in recognizing the *cracked pavement* object. Unlike the pavement distresses seen in Figure 2.4 from the research of Majidifard et al. (2020), the pavement distresses in our street view images are not exactly small cracks on an asphalt-surfaced roadways.

Additionally, the *curb ramp* object was also annotated by our volunteers. In Figure 6.5, we can see the annotations for curb ramps which are both correct;

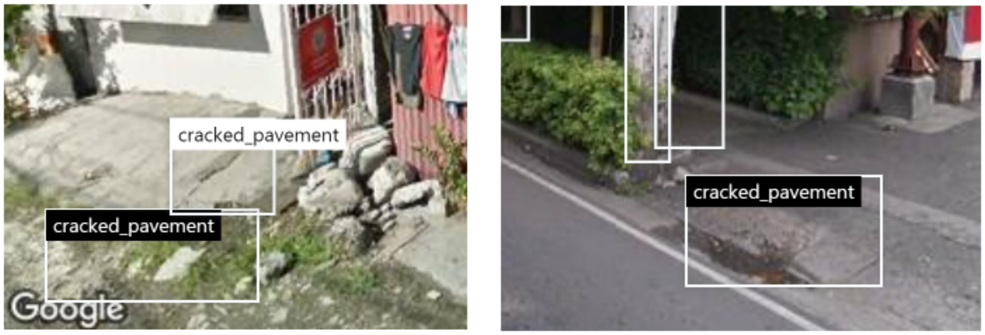


Figure 6.4: Comparison of cracked pavement annotations from our train set

however, the steepness and color painting of the curb ramp may affect how the model views the object. A curb ramp that is not painted and as steep as the image on the left may easily blend in as part of the entire sidewalk pavement, making it difficult for the YOLOv5 model to differentiate whether it is a curb ramp along the sidewalk or not. Also, the image on the right has shadows and harsher lighting conditions, affecting the overall clarity of the curb ramp on the image.

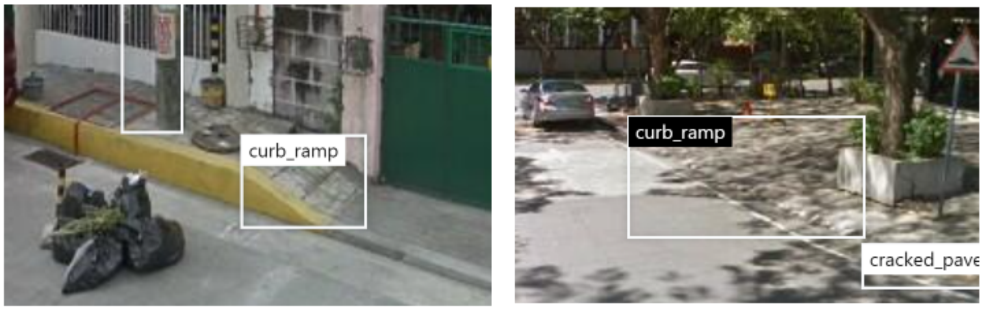


Figure 6.5: Comparison of curb ramp annotations from our train set

Lastly, another possible factor that could affect the YOLOv5 model in detecting these objects would be the image quality of the streetview images we collected. The street view images were somewhat pixelated compared to the images from the COCO 2017 dataset that were more detailed. Although all images from both COCO 2017 and streetview images have a highest resolution of 640x480, the images we obtained from the Google Street View API were still of lower quality. Also, the images from the COCO 2017 were focused on the specific object, whereas the street view images were focused on the street itself and not on the object in particular.

6.2 Crowdsourcing with Atlas

6.2.1 Annotation Results

As of August 13, 2021, we were able to collect a total of 5,047 annotations and accessibility scores from 48 unique users. As shown in Figure 5.4 of Chapter 5, users were able to select *"No sidewalk"* as an option for surface type, and due to that filtering we were able to reduce that number to 4,096 total annotations. To further substantiate the data for our accessibility model, we manually collected sidewalk widths for images annotated by users as described in Section 4.5. This led to a final data set containing 2,476 annotations with corresponding sidewalk widths.

For accessibility scores, we can observe in Figure 6.6 that the most common scores given to sidewalks in Manila and Makati were 4, 7, and 8. The perceived mean accessibility score is 5.643 with a standard deviation of 2.673. Some annotations, however, were done on the same image by multiple users. If we instead observe the accessibility scores per unique image then we obtain a mean accessibility score of 5.733 with a lower standard deviation of 1.316. The higher standard deviation despite having repeating images led us to believe that users may have had their own biases or standards when scoring sidewalks. For this, we decided to look at user demographics and activity to find a possible explanation.

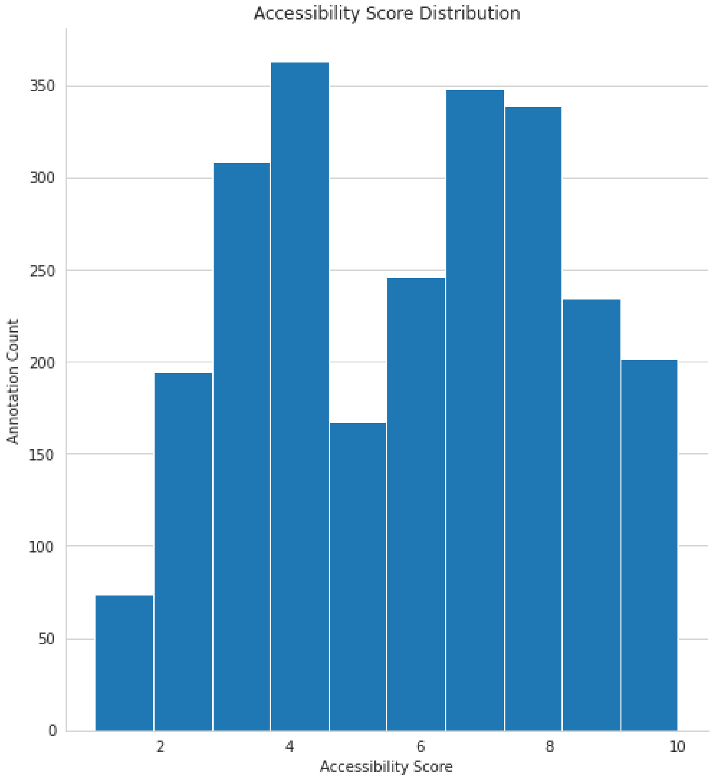


Figure 6.6: Perceived accessibility score distribution

6.2.2 User Demographics and Activity

We first look at the age of our users seen in Figure 6.7, where we see that most of our users are in their early 20's. It is common for people of this age to be fairly computer literate as well as have an understanding of commuting and pedestrian infrastructure, given that those in their early 20's in the Philippines usually take public transportation to school or work. We also have a few users above the age of 40 who have experienced more change in sidewalks from previous years compared to now. This may contribute to a better understanding of scoring accessibility. We also looked at the city of residence of our users as seen in Figure 6.8 where we see that the majority of our users come from Manila, one of the cities we selected for our crowdsourcing phase; we did not have any registered users though from Makati, the other city we had collected street view images from. Unfortunately, we were not able to gather users that make use of mobility aids who would have been ideal users to score accessibility.

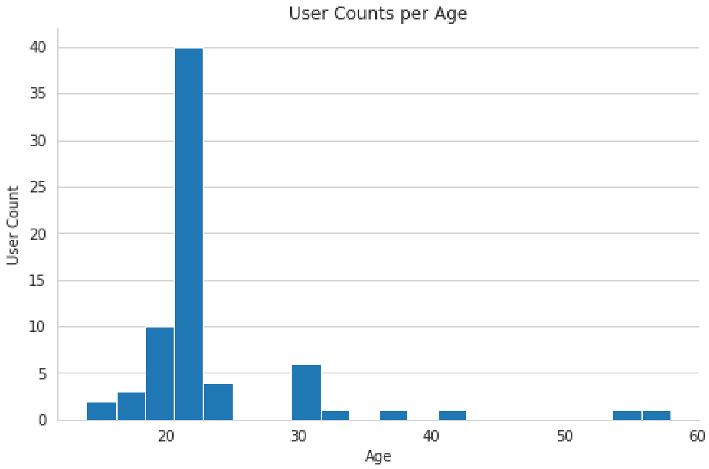


Figure 6.7: Registered user ages

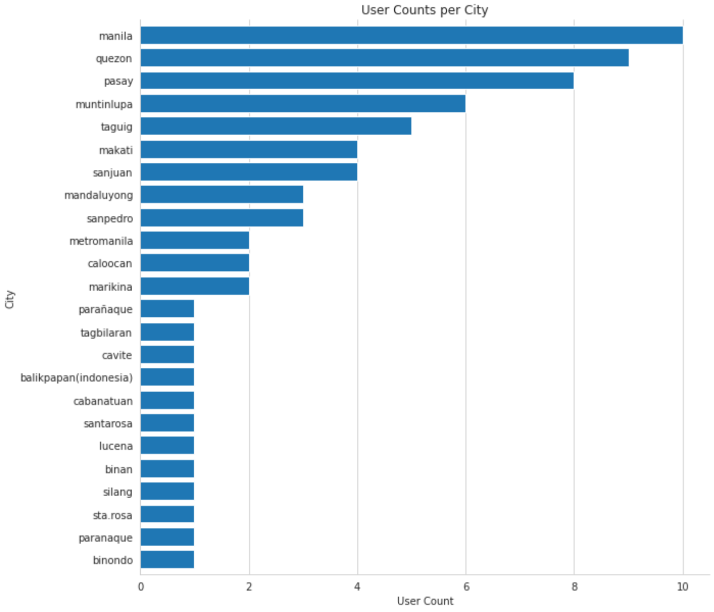


Figure 6.8: Registered user residences

Regarding user annotation activity, we observed a large distribution from the mean number of images annotated by users. With an average of 54 images annotated per user, we have a high standard deviation of 131.404, with our highest user annotating a total of 729 images. Figure 6.9 shows that 69% of our annotations come from the top 5 annotators, with the top annotator contributing 28% of all annotations just by themselves. The most likely reason for the top annotators having such high activity was the giveaway competition we conducted to gather more volunteers as mentioned in Section 4.4.

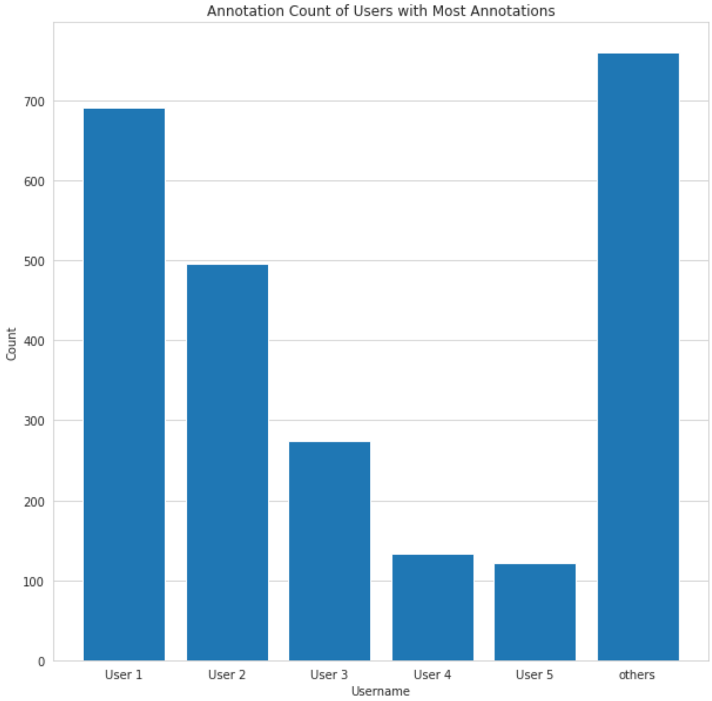


Figure 6.9: Annotations of top scorers

Additionally, there is no distinguishable bias from the top annotators, or from any annotators for that matter, in terms of their given accessibility scores, seen in Figure 6.10. The users were shown different images and therefore we did not expect their accessibility scores to be similar.

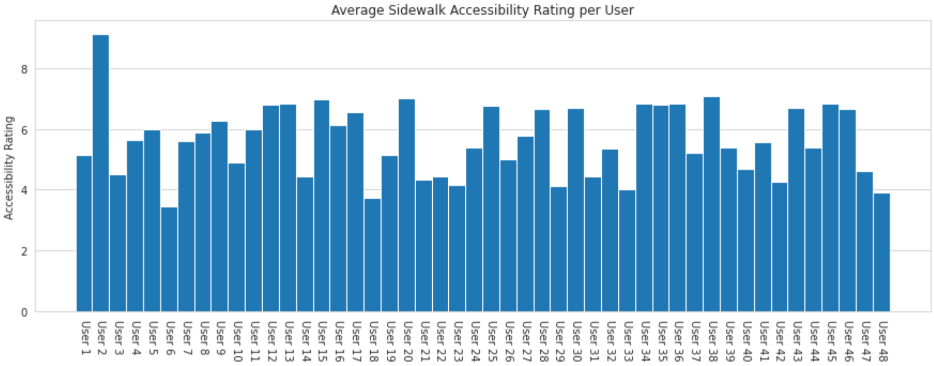


Figure 6.10: Perceived average accessibility scores of all users

6.2.3 Sidewalk Obstructions

In total, we detected 21,590 objects from all the 12,372 images used on Atlas, with only 6,485 (29.9%) of objects being considered as obstructions by users. We recall from Section 4.2.5 that the images uploaded to Atlas were the combination of the manual annotations from our volunteers and the annotations of the YOLOv5 trained on the COCO 2017 annotations to allow users to quickly provide an accessibility score for the sidewalk. Users were also able to label new obstructions in case they were not pre-labeled by YOLOv5, and 1,943 new objects were annotated by users, equating to 8.9% of all objects. We should also consider that we included an *Others* class from the list of objects that users could select in case they found an obstruction that didn't fall under any class. In total, we found that there were 544 objects selected as *Others* which we manually labeled and grouped into the following classes and categories:

- **Aesthetics** - Plants, Greenery, Sidewalk Design, Grass Wall
- **Utilities** - Barriers, Infrastructure, Tower, Water Meter, Shed
- **Commercial/Livelihood**: Sign, Tables, Chairs, Bottle Case, Refrigerator, Basketball
- **Clutter**: Rocks, Debris, Concrete Box

The counts of these objects as well as sample images from each category can be seen in Figure 6.11 and Figure 6.12.

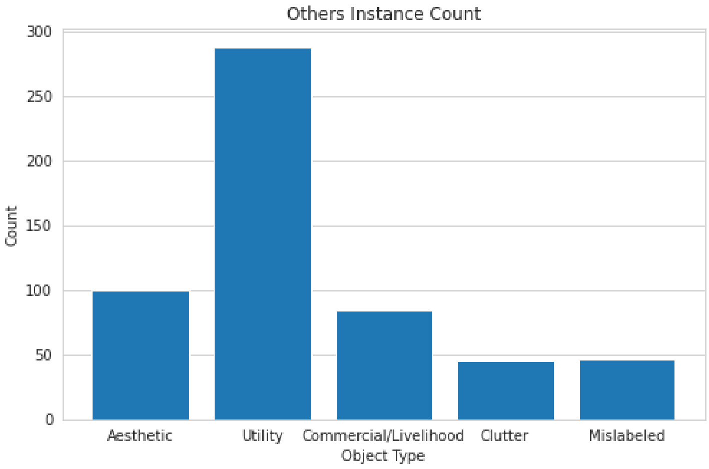
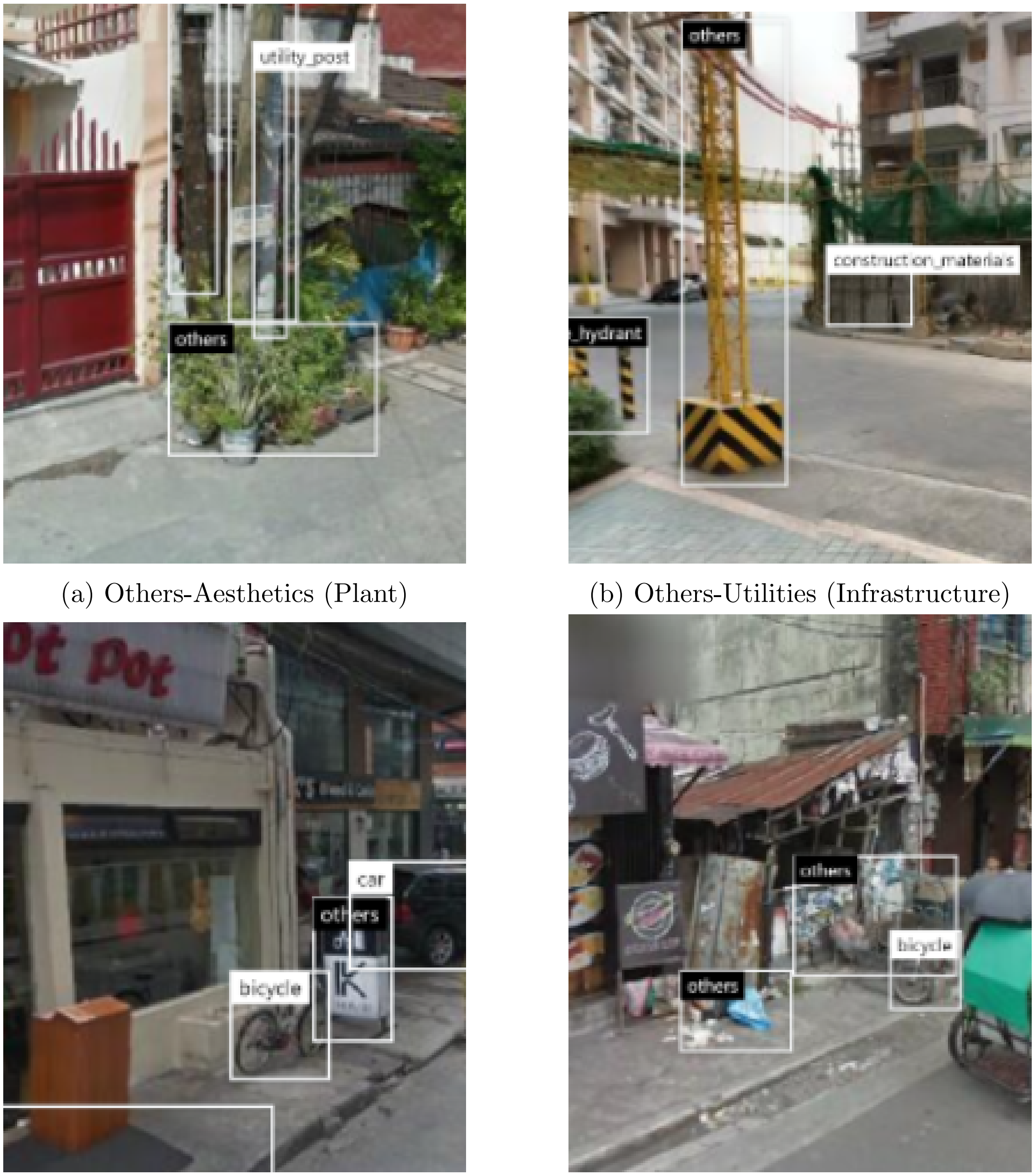


Figure 6.11: Counts of others objects



(a) Others-Aesthetics (Plant)

(b) Others-Utilities (Infrastructure)

(c) Others-Commercial (Sign)

(d) Others-Clutter (Debris)

Figure 6.12: Example images containing the ”others” object type

6.2.4 What Constitutes a Good Sidewalk

Based on all the data we’ve collected, we can come up with some basic assumptions for what makes a good sidewalk and what makes a bad one using accessibility scores as the dependent variable. Firstly, there is an observable pattern that sidewalk width is directly proportional with accessibility scores as seen in Figure 6.13. To further validate this claim, we used a Spearman Rank-Order Correlation statistical test. The spearman's rho correlation coefficient is as follows: r_s=0.3306, p=2.9395 x 10^{-64} indicating a positive correlation between sidewalk width and accessibility score.

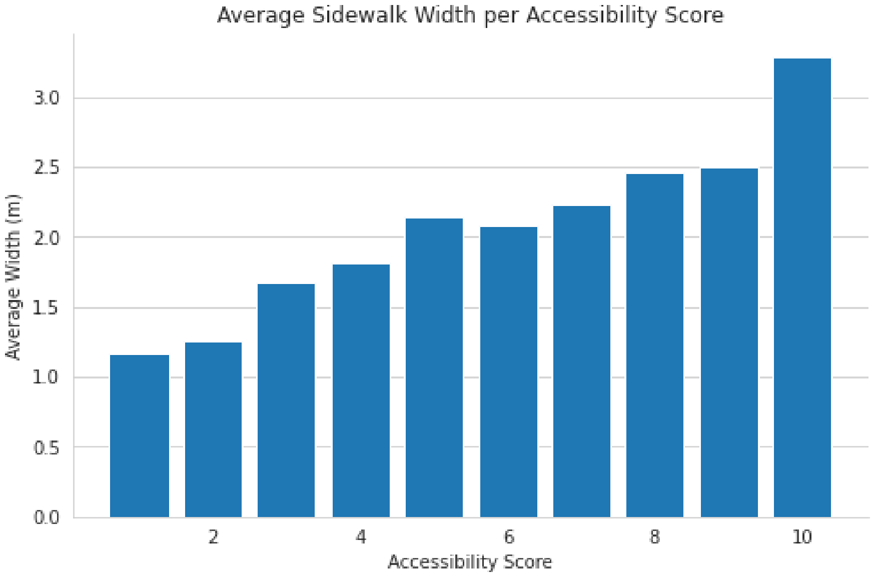


Figure 6.13: Comparison of sidewalk widths and perceived accessibility scores

Next, we take into account the surface types of sidewalks which we asked users to select among *Rough*, *Smooth*, and *Slippery*. Similarly to sidewalk width, the count of smooth sidewalks increases from accessibility scores 1 to 10 as seen in Table 6.1. The count of rough surfaces also seems to follow a downward trend, but slippery surfaces do not seem to affect the perception of accessibility. In accessibility score 10, we even see that 91% of surfaces were considered smooth.

Finally, we take a look into obstructions which negatively affect accessibility score, while exploring those objects that may actually provide a positive effect on sidewalks. Table 6.2 displays key figures which allow us to analyze and observe patterns between objects and accessibility. We observed that there is a downward trend of the ratio between total objects found in the image to obstructions as accessibility score rises; however, this correlation is not as significant. Spearman's rho correlation coefficient was used to asses the relationship on total objects found in the image and accessibility score. There was no significant correlation between the two, r_s=-0.036, p=0.0717. On the other hand, we observed a clearer correlation with the number of obstructions found in each image. Again, the spearman's rho correlation was used to assess the relationship between accessibility score and

Accessibility Score	Count of Surface Type			
	Rough	Smooth	Slippery	Percentage of Smooth Surface
1	38	34	1	46%
2	83	106	6	54%
3	110	179	19	58%
4	106	236	21	65%
5	47	115	3	69%
6	54	166	25	67%
7	42	250	56	71%
8	29	261	49	76%
9	7	200	26	85%
10	5	185	12	91%

Table 6.1: Perceived accessibility scores and surface types table of comparison

Accessibility Score	Mean Total Objects	Mean Obstructions
1	7.652	1.859
2	7.865	3.200
3	8.099	2.701
4	9.082	3.068
5	8.827	2.751
6	8.785	2.931
7	8.890	2.789
8	8.637	2.345
9	7.811	1.859
10	6.797	0.856

Table 6.2: Perceived accessibility score and objects table of comparison

the obstruction count. There was a negative correlation between the two: r_s=-0.110, p=1.945 x 10^{-8}.

Seeing that these numbers are found at the extremities of the accessibility score scale, this led us to the inference that some objects contribute to a positive score just as much as obstructions contribute negatively to the score. We dove deeper into this by comparing objects commonly present in images with scores of 1 and 10, which can be seen in Figure 6.14. In this figure we also show the obstruction status of these objects as determined by our users. The figure shows that trees are more present in 10-scored images and that most trees were not considered as obstructions. We can observe the same thing with benches, so we can interpret that locations with non-obstructing trees and benches tend to seem more accessible such as outdoor parks and public transportation spots. On the other hand, motorcycles, tricycles, and construction materials are more prevalent in 1-scored images than in 10-scored ones.

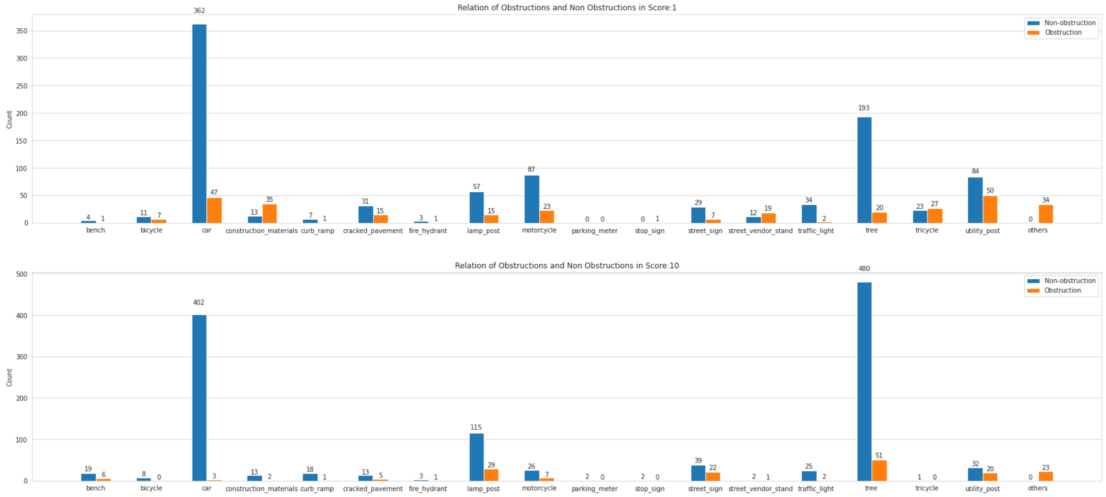


Figure 6.14: Comparisons of perceived objects in accessibility rating 1 and 10

6.3 Accessibility Score Model

As mentioned in Section 6.2.1, 2,476 annotations with accessibility scores were left after pre-processing. The annotations are split with a ratio of 80% and 20% respectively. This gives us 1,980 entries for the train set and 496 entries for the test set. In choosing the right model, we ran different models and also perform hyperparameter tuning by getting the most optimal parameters through GridSearchCV. We also used ridge as the regularization for both the linear regression and XGBoost models. We used ridge regularization because we did not want to disregard features when performing regularization. From this, we select the model that performed the best in terms of cross-validation score through 5-fold cross validation. The cross-validation score represents the accuracy of the model. We also look at other performance measures such as the root mean squared error (RMSE) for the regression models while we looked at F1-score for classification models.

6.3.1 Accessibility Score Model Performance

First, we ran a linear regression model with ridge regularization from `scikit-learn`. After getting a accuracy of 29.14% and an RMSE of 2.13, we decided to look at

Model	Train-Test Split	Train Score	Test Score	5-Fold Cross Validation Score (Accuracy)	RMSE
Linear Regression	80/20	32.6%	29.33%	29.14%	2.13
XGBRegressor (Most Optimal)	80/20	50.08%	33.66%	33.43%	2.13
XGBRegressor (Base)	80/20	44.70%	31.66%	31.95%	2.10

Table 6.3: Regression model performances

Model	Train-Test Split	Train Score	Test Score	5-Fold Cross Validation Score (Accuracy)	F1-score (weighted)
XGBClassifier (Base)	80/20	45.80%	22.18%	23.02%	20.21%
XGBClassifier (Optimized)	80/20	36.52%	21.77%	22.70%	17.81%

Table 6.4: Classifier model performances

XGBoost in order to see if the model performance would be better. Similar to the linear regression model, we used ridge as the reguralization parameter for the XGBoost models. Hyperparameter tuning is also made easier in XGBoost because of its popularity in literature. Initially, we opted to use XGBClassifier since the response variable for our model is a discrete value. However, when we ran the XGBClassifier model, we got a accuracy of 23.02% which was marginally worse than the linear regression model. Additionally, we looked at the weighted F1-score of the classifier and found that it was 20.21% meaning that it's precision and recall were low. After this poor result, we tuned the hyperparameters of the XGBClassifier with use of GridSearchCV. GridSearchCV runs several configurations of hyperparameters and helps in selecting the best configuration of hyperparameters given the training set.

Hyperparameter	Value
`learning_rate`	0.01
`max_depth`	3
`min_child_weight`	3
`n_estimators`	500

Table 6.5: Best hyperparameter values found for the XGBClassifier model

The following parameter were tuned: *learning_rate*, *min_child_weight*, *max_depth*, and *n_estimators*. With the tuned XGBClassifier, we got a accuracy of 22.70% and a weighted F1-score of 17.81%. The poor performance of the optimized XGBClassifier led us into using the XGBRegressor model to model our data. Without any hyperparameter tuning, the XGBRegressor got a 5-fold accuracy of 31.95% and an RMSE or 2.10. While the performance of the XGBRegressor was the best that we had so far, we tried to tune it and compare its performance.

The XGBRegressor parameters that were tuned are the following: *colsam-*

Hyperparameter	**Value**
`colsample_bytree`	0.7
`learning_rate`	0.01
`max_depth`	5
`min_child_weight`	1
`n_estimators`	500
`subsamples`	0.5

Table 6.6: Best hyperparameter values found for the XGBRegressor model

ple_bytree, *learning_rate*, *max_depth*, *min_child_weight*, *n_estimators*, and *subsample*. With the tuned XGBRegressor model, we got a accuracy of 33.43% with an RMSE of 2.13. While the RMSE of the tuned XGBRegressor model was slightly higher than the base XGBRegressor model, it had a better accuracy than the base XGBRegressor model. In the end, we opted for the tuned XGBRegressor due its better accuracy based on its 5-fold cross-validation score. As seen in figure 6.15, we can see multiple predicted entries with a noticeable margin of error. From this, we realize that the accuracy of the predictions are considerably low. 33.43% accuracy of the model and the RMSE of 2.13. Given that the accuracy of the tuned XGBRegressor is low, we can look into improving the accuracy of the model by providing a robust dataset. The inconsistencies in crowd sourced accessibility scores and labels could also have contributed to the accuracy of the model. Re-training the model with a dataset filled with images with multiple accessibility scores can also benefit the predictions of the model.

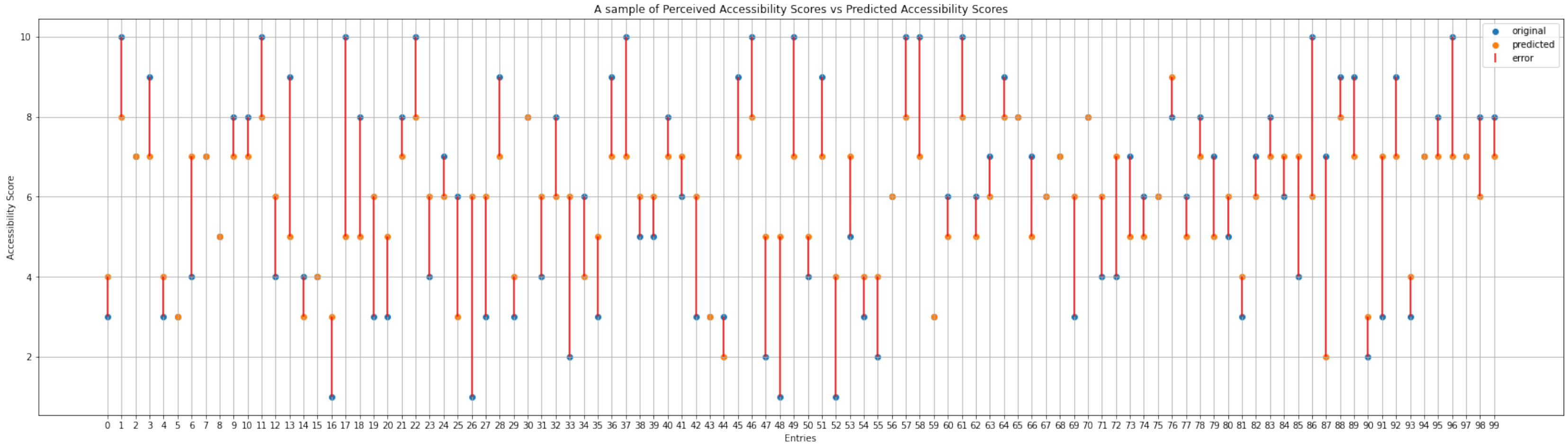


Figure 6.15: A sample of perceived vs the predicted values of the accessibility score model

6.3.2 Feature Importance of Predicted Accessibility Scores

Another important finding that we get from the XGBRegressor is the calculation of feature importance used in the model. Quantifying the influence of each feature in predicting the accessibility score allows us to analyze the features that are desirable and undesirable in assessing sidewalk accessibility. By using SHAP values, we are able to see the features with the most influence on the accessibility score.

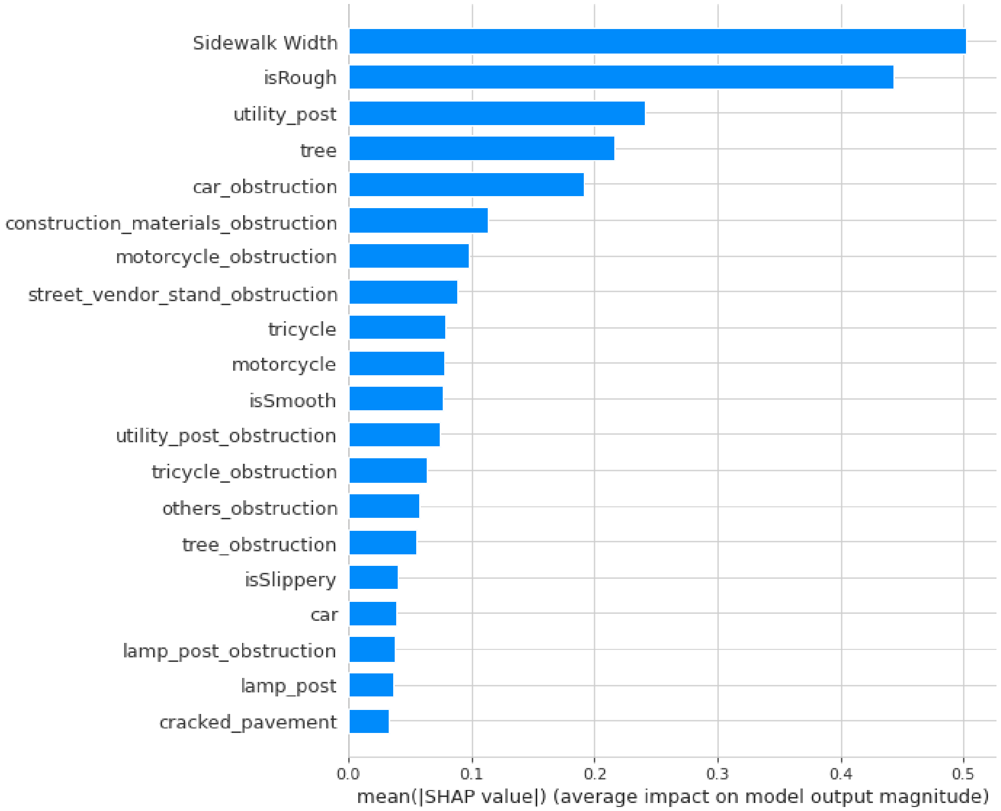


Figure 6.16: Average absolute SHAP value of top 20 influential features on predicted sidewalk accessibility score

In Figure 6.16, the influence of a feature is measured in a mean absolute SHAP value. The mean absolute SHAP value does not look at the positive or negative influence but the total influence a feature has on a predicted sidewalk accessibility score. Based on Figure 6.16, the features with the considerable influence on a predicted sidewalk accessibility score are sidewalk width, rough sidewalk surface type, utility post non-obstruction, tree non-obstruction, car obstruction, construction material obstruction, motorcycle obstruction, steet vendor obstruction, tricycle non-obstruction, motorcycle non-obstruction, smooth sidewalk surface type, utility post obstruction, tricycle obstruction, tree obstruction, slippery sidewalk surface type, car non-obstruction, lamp post obstruction, lamp post, and lastly cracked pavements.

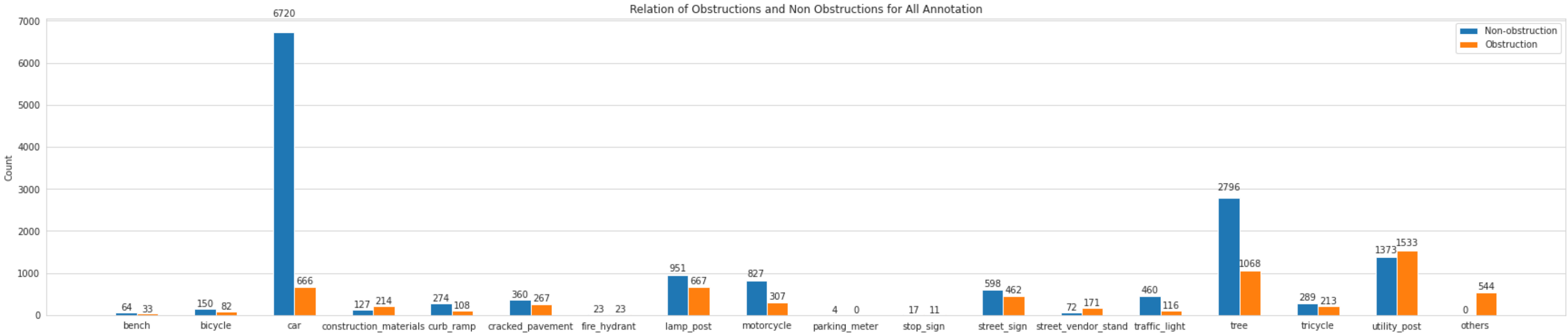


Figure 6.17: Total Number of Objects as Obstructions and Non-Obstruction

Most of the top 20 features listed in Figure 6.16 have a commonality within them that makes them influential to the predicted sidewalk accessibility score. Each of them has a high number of occurrence in the annotations. For sidewalk width and surface types, every annotation must have at least one sidewalk width and one surface type. Because of this, their mean absolute SHAP value is the highest among the features. In the top 20 features, we could also see the same objects with the obstruction and non-obstruction pairing. Objects such as cars, motorcycle, utility post, trees, tricycle and lamp post are the features that both have their obstruction and non-obstruction form in the top 20 features. In looking at the objects with the most counts on Figure 6.17, we confirm that these objects have the most count among the other objects. The occurrences of features has a direct relationship with the overall influence it has on the SHAP value.

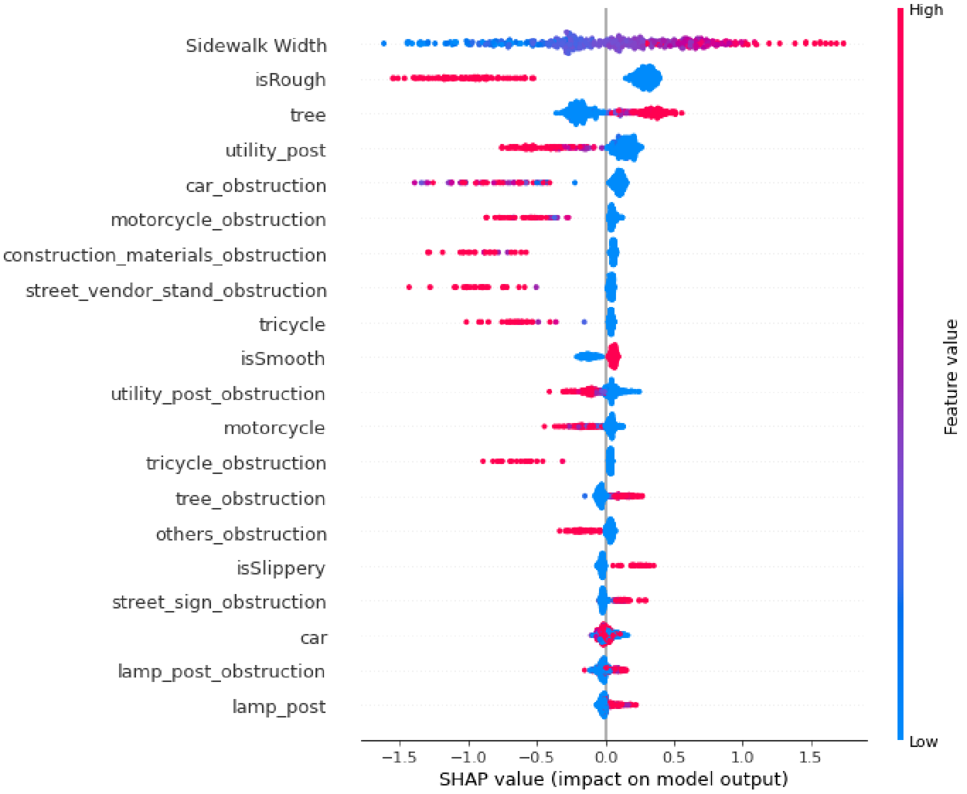


Figure 6.18: SHAP value distribution of top 20 influential features

Now that we know the overall influence of a feature, we look at another summary plot to see how which values of these features make them have a positive or negative effect on the predicted sidewalk accessibility score. In Figure 6.18, we can see the distribution of high and low feature values and their respective influence on the model. For sidewalk width, wider sidewalks tend to have a positive influence on the predicted accessibility score, with most red data points having

0.5 - 1.5 SHAP values. For rough sidewalk surfaces, a high value will lead to a negative influence on the score. For most objects, high values have a negative influence on the predicted sidewalk accessibility score. It does not matter whether that object is an obstruction or not, as long as it has a high value, it will have a negative influence on the predicted sidewalk accessibility score. However, the tree object is an exception to this trend. Its presence, whether as an obstruction or not, has a positive influence on the predicted sidewalk accessibility score. It's also good to point out that cars, utility posts, motorcycles, construction materials, street vendor stands, and tricycle are obstructions that have the largest negative influence on sidewalk accessibility. This is because these objects are notorious in being found in sidewalks. Some common examples of this are when tricycles, motorcycle, and street-vendors park on sidewalks. Besides this, construction materials such as cement powder disrupting the surface of a sidewalk are often the pain of pedestrians that walk in Metro Manila. To find out what constitutes a high value or low value for features, we can look at individual breakdowns of three levels of accessibility scores.

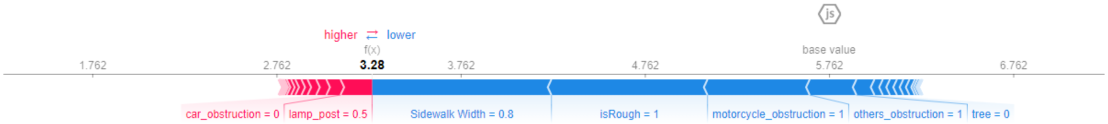


Figure 6.19: Breakdown of low accessibility score prediction

The three levels of the predicted accessibility scores are low, mid, and high as seen in Figures 6.19, 6.20, 6.21. The grouping of these levels were based on the average or base value of the predicted scores. Looking at the base value, the average score that was predicted by our model was 5.762. Now we look at an instance of a low accessibility score prediction. In this case it is 3.28. Looking at the features, we see that a sidewalk width of 0.8, a rough surface type, and a motorcycle obstruction push the predicted sidewalk accessibility score down. We also see that a lack of car obstructions and the presence of lamp posts have a little positive effect on the predicted sidewalk accessibility score.

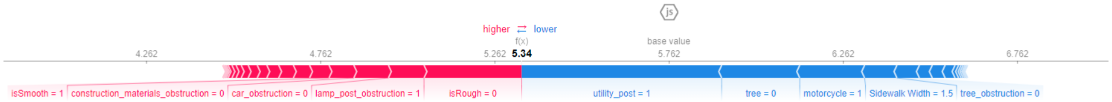


Figure 6.20: Breakdown of baseline accessibility score prediction

For the mid accessibility score prediction, the presence of utility posts, even as a non-obstruction, negatively influenced the accessibility score. The lack of trees also negatively influences the accessibility score. A sidewalk width of 1.5 meters also has a slight negative effect on the prediction. However, we can see that when the surface type of the sidewalk is smooth and not rough, it has a large positive effect on the accessibility score.

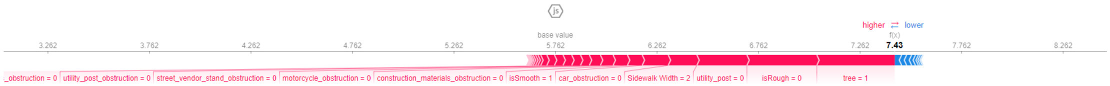


Figure 6.21: Breakdown of high accessibility score prediction

Lastly for the high accessibility score prediction, the presence of trees, lack of utility posts, car obstructions, construction material obstructions, motorcycle obstructions, street vendor obstructions, utility post obstructions push the score higher. The sidewalk width of 2 meters also produces a positive effect on the accessibility score.

From the three observations, we can see that the surface type of a sidewalk heavily influences the predicted sidewalk accessibility score due to the fact that the `isRough` feature is among the features with the biggest portion of the breakdown. We can also see that the sidewalk width's influence slowly turns from negative to positive as its value increases. When the width of the sidewalk is 0.8 meters, it has a huge negative influence on the score, with 1.5 meters, it still has a negative influence but it is noticeably less. With 2 meters, we can see that it has a slightly positive influence on the predicted sidewalk accessibility score. From this we can say that the point of change from negative to positive influence could be in the region between 1.5 meters and 2 meters. Lastly, the presence of trees also provide a positive influence. This could be because of the protection that trees provide from the road. Also the trunk of a tree also makes the sidewalk width increase resulting in wider sidewalks.