MARMARA UNIVERSITY

FACULTY OF ENGINEERING

CSE4288

Introduction to Machine Learning

TERM PROJECT

Model Development Progress

Group: 2

Table of Contents

1. MODELLING WORK COMPLETED

1.1 Decision Tree

In this part of our project, we aimed to determine whether images contain “crosswalks” using the Decision Tree classifier. We performed the classification process by representing the images with image embedding instead of raw data. After loading the embedding data required for training and validation from `.npy` files, we processed the tags in the JSON file and matched them with the image names.

Using the embedding data and tags, we trained our Decision Tree model and then tested it on the validation set. We evaluated the model's performance using metrics such as accuracy rate, classification report and confusion matrix, and recorded these data for comparison with the performance of other classifiers.

1.2 Naïve Bayes

In this section of the project, we employed the Naive Bayes classifier to determine the presence of “crosswalks” in images. Rather than using raw image data, we utilized the precomputed features from .npy files. Additionally, we obtained the corresponding labels from the JSON file, where each image was annotated as containing a crosswalk or not.

After mapping the labels to the feature vectors, we trained our Naive Bayes classifier. We evaluated the performance on a validation set using standard metrics such as accuracy score, classification report, and confusion matrix. These metrics allowed us to quantitatively gauge how well the classifier distinguished between images containing crosswalks and those that did not. The results serve as a baseline for comparing different classifiers’ performances.

1.3 K-Nearest Neighbours (k-NN)

In this phase, we used the K-Nearest Neighbors (KNN) algorithm to classify images as containing "crosswalks" or not. Instead of raw image data, we used pre-computed image embeddings loaded from .npy files for training and validation. Labels were extracted from a JSON file and matched with image names to prepare the dataset.

We trained the KNN model with K=5 neighbors and evaluated its performance on the validation set using metrics such as accuracy, a classification report, and a confusion matrix.

The confusion matrix was visualized as a heatmap to analyze the model's performance and compare it with other classifiers in the project.

1.4 Logistic Regression

In this phase of the project, we utilized the Logistic Regression classifier to determine the presence of “crosswalks” in the images. Instead of using raw image data, we leveraged image embeddings to represent the features. These embeddings were loaded from pre-saved .npy files for both training and validation datasets. The labels, indicating the presence or absence of crosswalks, were extracted and matched to the corresponding images using annotations provided in a JSON file.

After mapping the labels to their respective image embeddings, we trained the Logistic Regression model with a maximum iteration limit of 1000 to ensure convergence. We then tested the trained model on the validation set and evaluated its performance using key metrics such as accuracy, a classification report, and a confusion matrix.

The results provided insights into the classifier’s ability to differentiate between images containing crosswalks and those that do not, with visual analysis facilitated by a confusion matrix heatmap. These findings will later be compared to the performance of other classifiers in the project.

1.5 YOLO

After obtaining the preprocessed data on the previous step of the project, we were ready to add the crosswalk weight to the pretrained YOLO model. First, we trained our model using YOLOv8 with the parameters of 10 epochs, 640 image size and 16 chunk size. With these parameters, the first obtained model had a lower performance than expected due to low image size with possible insufficient epochs time. Moreover, the first training is done using CPU, which also increased the training time significantly. A simple test for the first model and its result is given in Figure 1.

Figure 1: Test result of the first model

Second training is done with much higher epochs value with 50 and a higher image size with 1024p. The performance of the model has increased, but still the performance is quite low when compared with the ideal. Most probably, 50 epochs caused an overfitting, as the value is larger than necessary. A simple test for the second model and its result is given in Figure 2.

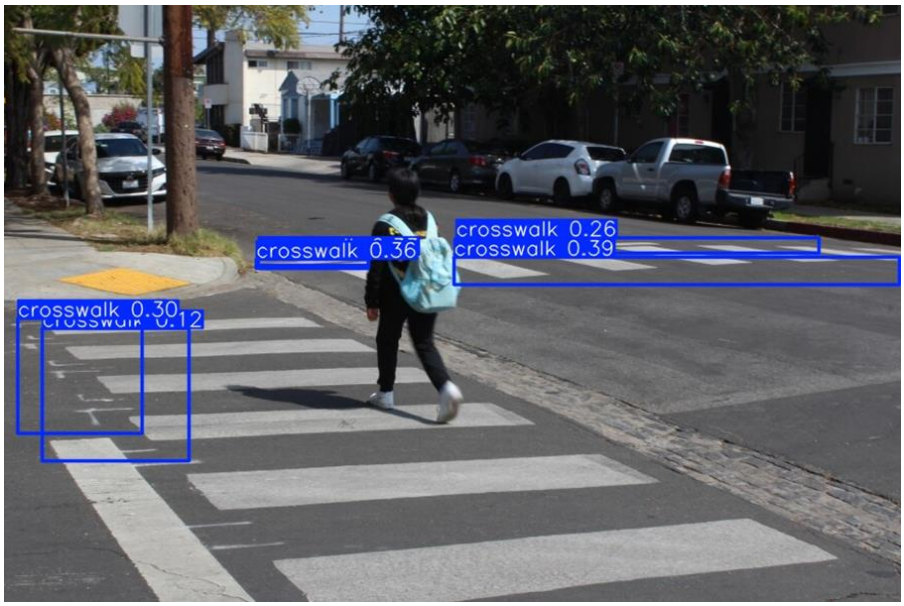

Figure 2: Test result of the second model

Additionally, we believe the dataset should be cleared further for better performance. This aspect will be handled during the model evaluation and model optimization phases.

2. CHALLENGES ENCOUNTERED AND SOLUTIONS

2.1 Decision Tree

Once the image embedding was finished, the classification process became simpler. Although the dataset being unbalanced (due to the difference between images with and without "crosswalks") was a downside, the large size of the dataset helped reduce the impact of this issue. Still, the model's accuracy was low, which indicated that the embeddings might not be clear enough for the classifier. A different classification method was tried with the available data to compare with other models.

2.2 Naïve Bayes

One of the main challenges we encountered while using Naive Bayes was the sensitivity of to variations in feature distributions. Since our dataset had an imbalance of “crosswalk” vs. “no crosswalk” images, the model sometimes leaned toward predicting the majority class. To mitigate this, we closely monitored not just accuracy but also recall and precision to ensure that minority cases were not completely overlooked.

2.3 K-Nearest Neighbours (k-NN)

After the image embedding process, KNN was used for classification. The main challenge was the dataset imbalance, with more images without "crosswalks." This caused the model to potentially favor the majority class. To address this, we focused on additional metrics like precision, recall, and F1-score, rather than just accuracy. The KNN model was trained with K=5, and its performance was evaluated using accuracy, classification report, and confusion matrix, which helped assess its effectiveness compared to other models.

2.4 Logistic Regression

One of the main challenges faced during the Logistic Regression phase was handling the imbalance in the dataset, where images without "crosswalks" significantly outnumbered those with "crosswalks." This imbalance led to the risk of the model favoring the majority class, potentially reducing its ability to correctly classify the minority class. To address this, we ensured the use of appropriate metrics, such as the classification report, to evaluate the performance beyond just accuracy, focusing on precision, recall, and F1-score for both classes.

Another challenge was ensuring that the model converged effectively during training. Logistic Regression's convergence depends heavily on the optimization process, and setting the maximum iteration parameter to 1000 helped address potential convergence issues.

Despite these challenges, the model achieved reasonable accuracy and provided clear insights into its predictive performance through metrics such as the classification report and confusion matrix. This helped in evaluating its performance against other classifiers used in the project

2.5 YOLO

The main challenge of training the YOLO model was the time needed for the training phase and the device requirements. The first training is done using CPU and with 10 epochs, the training took 2-3 hours for only 1000 images. The second training is done using GPU with 50 epochs, the training took 12 hours for 25000 images. Although we increased the parameters and size of the dataset, the training time did not increase as much due to GPU usage.

Second current main challenge of the final model is the variety of images in the dataset. The dataset we are currently using includes many different types of crosswalks. There are even images of barely seen, dark and blurry crosswalks, which probably is the main cause of the data defilement. We are expecting to solve this issue by doing a further cleaning of the dataset.

3. CONCLUSION

In summary, the classification methods explored in this project—including Decision Tree, Naive Bayes, KNN, and Logistic Regression—provided different perspectives on how to detect crosswalks in images effectively. While imbalances in the dataset posed challenges, meticulous attention to appropriate evaluation metrics helped ensure a more robust assessment of each classifier's performance. Our findings indicate that further improvements, such as refining the dataset and tuning model parameters, can enhance classification accuracy. Moving forward, the experience gained from these experiments will inform subsequent steps in refining the overall system for real-world applications.

4. REFERENCES

[1] Fisher Yu. 2024. “BDD100K Documentation”. Date Accessed: 25.11.2024. (https://doc.bdd100k.com/index.html)

[2] Ultralytics, 2024. “Ultralytics YOLO Documentation”. Date Accessed: 25.11.2024. (https://docs.ultralytics.com/tr)