Traffic Sign Classification Project Report

Melih Can KANMAZ
Statistics Department
METU
2502169
Ankara, Türkiye
can.kanmaz@metu.edu.tr

Refik Toprak TELLİ
Statistics Department
METU
2502391
Ankara, Türkiye
toprak.telli@metu.edu.tr

Mustafa Uğur YALÇIN
Statistics Department
METU
2502458
Ankara, Türkiye
ugur.yalcin@metu.edu.tr

***Abstract*—This study aims to find the best machine-learning model for traffic sign classification using the Mapillary Traffic Sign Dataset. Started with data cleaning steps such as matching labels with the images and cropping the signs from the photos. Utilized different transformation methods to used the data more efficiently. Detected class imbalance and implemented SMOTE to reduce the effect of class imbalance on the data. To reach the best learning algorithm, we tried Classical ML models, CNN and Ensemble models. Used Google Colab for runs and tracked the metrics of experiments on Neptune.ai. Evaluating the results, we reached the conclusion for best model for prediction for the traffic signs.**

***Index Terms*—traffic sign, classification, machine learning, neural networks, deep learning, transfer learning, SMOTE, ensemble learning**

I. Introduction

In today's world where autonomous technologies are increasing, autonomous driving has become a popular concept. The detection and classification of road signs for safe driving is a crucial element for autonomous driving. Thanks to traffic signs, the system can receive basic information about the road it is on and adjust the driving to prevent potential hazards. Unfortunately, detecting and correctly perceiving traffic signs presents a challenge. Our main goal is to ensure that traffic signs are perceived correctly for safe driving. In line with this goal, we hope to contribute to the safe driving of autonomous vehicles in the future. This project focuses on creating a robust classification model using Mapillary Traffic Sign Dataset. Our goal is to contribute to safer autonomous driving in the future by finding solutions to real-world problems through these models.

II. Related Work

In general, traffic sign classification projects uses various models and data preprocessing techniques. Previous works showed us to how to handle traffic sign data efficiently[4]. Common approach to get better performance from the models is utilizing various data augmentation methods. These methods can help reducing the overfitting on the model and generalise well on the traffic sign data.

Investigating recent papers on the field, mostly CNNs and state of the art methods are used because they perform well on the traffic sign classification task [6]. Some of them have created their own CNN architectures or used SOA methods such as Vision Transformers[1]. Another approach is to use transfer learning and ensemble learning at the same time to achieve high performing models. In [3], they have trained CNN, ResNet and GoogleNet and amalgamated their results to create an ensembled model. Also, another approach for ensemble learning is to use a model such as AdaBoost to improve the performance of weak learners. [2].

These projects mainly use several datasets to train and test the models. Two main datasets are German Traffic Sign Benchmark dataset [3] and Belgium Traffic Sign dataset (BTSDS)[5].

After doing a literature search on the traffic sign classification problem, these papers have helped us to set an expectation on how well our models should perform and which methods we should use.

III. Method

A. About Dataset

In this project, we have used 'The Mapillary Traffic Sign Dataset for Detection and Classification on a Global Scale'. This dataset has three fully annotated and three partially annotated parts and more than 300 manually annotated classes. Since each part is too big and we are only interested in the traffic signs's label, not the full photo, we have decided to use first fully annotated part and add other parts if needed. The first part has around 12 thousand fully anotated photos. After cutting the bounding boxes, we had 59,552 traffic sign images to train our models.

B. Data Preprocessing

As mentioned in the data sets section, images usually contain traffic signs within a larger scene. Therefore, we have to crop the according to the bounding boxes given in the dataset. However, traffic signs are not the same size, which some models require. Hence, we have added a black padding to the images and resized them to a fixed size while protecting the aspect ratio.

1) Data Augmentations: To make models more robust and add more variability to the data, we have applied different data augmentation techniques. For this project, we have used different combinations of these techniques from albumentations library:

- RandomBrightnessContrast
- HorizontalFlip
- VerticalFlip
- Normalize
- Rotate

2) Class Imbalance Problem: The first part of the dataset, has a massive imbalance problem. The amount of images for each class ranges from 30 thousand images to 1 image. Therefore we have omitted very small classes, applied SMOTE to small classes to oversample them and get a random sample from bigger samples. Each sample had 1200 photos at the end.

Fig. 1. Synthethic images generated using SMOTE.

C. Model Selection

In this project, we wanted to test how well both classical and more modern approaches would perform on this traffic sign classification task.

Classical ML approach:
In this section, we selected to test classical ML approaches as the first model trials. Aimed to find more efficient models in simpler and faster trains. Main approach was the SGD classification with different parameters. The primary benefit of these models was their simplicity compared to other models (In this study CNN and ensemble models). Having numerous instances for train, simplicity was crucial to make the model more efficient. On the other hand, the visual data is not strong hand of the classical ML models. CNN and ensemble models generally perform better. Therefore, while SGDClassifier is picked for its simplicity and efficiency, its low performance on model adequacy is expected.

CNN + Maching Learning:
This approach combines the power of convolutional neural networks's ability to extract features from images with the classical machine learning approach. We aim to boost to performance of the classical ML approach.

CNN:
This approach just uses the convolutional neural networks to classify the traffic signs. We used CNN with 3 hidden layers. After that, 2 new hidden layers added and better results have been accomplished. However, complex and heavy neural network models did not work well. Therefore, more simply models were used for better results. Our goal is increased the strength of the model due to flexibility of CNN. Also, Adam optimizer with learning rate equal to 0.001 was used.

Ensemble Learning:
We used this method to obtain better results due to using different learners. Combination of models generally give better results for datasets including images. Then, ensemble learning, mostly 3 different method were used to make the model simply and better in terms of performance. To obtain clear results, 5-Fold cross-validation were used.

Transfer Learning:
This approach is a popular approach in this field, and it combines the performance of models that are trained on large datasets with our smaller dataset. We expect this approach to perform the best.

D. Experimenting and Visualization

During the experiments, we used Google Colab as cloud service for our runs. To track the experiment metrics of the models, we utilized Neptune.ai platform. Connecting two platforms together, we run our tests for model selection and tracked the metrics of different models with Neptune.ai.

Our approach for conducting extensive experiments is:

Tracking Metrics:
We plan to use softwares such as Neptune.AI to track each experiment's metrics. Compare them with different experiments to ensure that we don't miss anything in the process of conducting experiments.

Using Different Algorithms:
On each experiment, we try to change algorithms that we use in our pipeline to find best performing data augmentations, models etc.. Also, we want to find the best combinations of them to find which one is working best among all iterations.

Hyperparameter Tuning:
While changing the algorithms, we also want to tune the hyperparameters such as batch size, dropout rate, rotation angle etc. to get the best-performing iteration.

IV. Results

Classical ML:
In this model we utilized SGDClassifier. Along with that

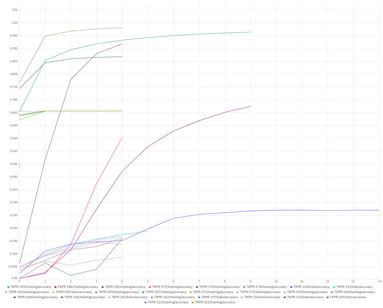

Fig. 2. Train Accuracy

there were couple of parameter tests including alpha level, loss function, penalty, max iteration and batch size. Since the model only used CPU in runs because of libraries used, it takes long time to fit the model. Therefore, the tests conducted on proportionally randomly selected 1/5 of the data, approximately 10000 photos. Multiple trials showed that the best parameter for this fit is logistic loss function, l1 penalty, 0.001 alpha level, 1000 as max iterations and batch size as 32.

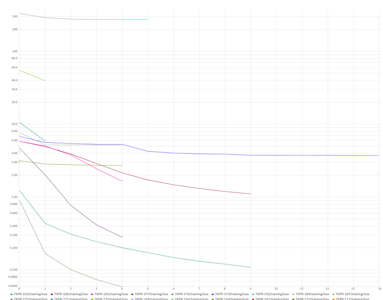

Fig. 3. Train Loss (Logarithmic)

CNN + Maching Learning:
This approach actually performed better than the classical machine learning models. Before applying SMOTE, the models got stuck around 0.65 accuracy and 0.52 f1 score. After applying SMOTE the performence of the models increased to the 0.75 accuracy and 0.76 f1 score.

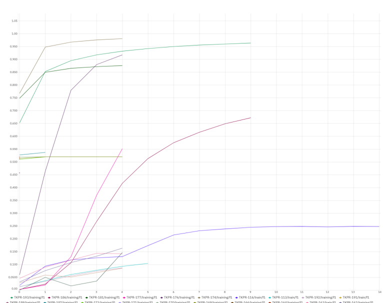

Fig. 4. Train F1 Score

CNN:
This approach performed better than the CNN + ML approach, however before applying SMOTE to solve the class imbalance problem, the models got stuck around the similar accuracy and f1 score and they seemed not to be learning. After applying SMOTE the best performing model got 0.97 accuracy and f1 score after training for 5 epochs. On the other hand, other models did not perform as well as this one, and they got 0.80 - 0.85 accuracy and f1 score.

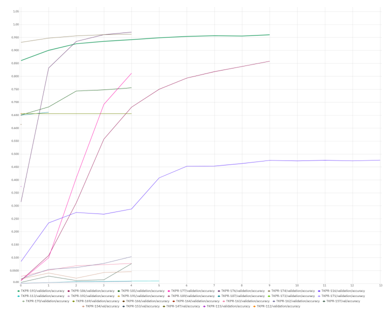

Fig. 5. Validation Accuracy

Transfer Learning:
This is the best performing approach on average. We have used ResNet50 model from the torchvision library and fine tuned it with our dataset, but this approach had the same problems before SMOTE. After the SMOTE, the best performing model got 0.96 accuracy and f1 score and we can say that on average these ResNet50 models performed better then only custom CNN models.

Fig. 6. Validation Loss

Ensemble Learning:
When we talk about ensemble learning, 3 different methods which are random forest, decision tree and XGBoost were used. Also, 5-Fold cross-validation were used in stacking. When the overall results have been compared, maximum 0.725 accuracy score has been accomplished. However, other scores are not good enough which mostly lay between 0.1 and 0.2.

V. Discussion

In addition to our efforts, there are several other things that can be applied to try to achieve better performing models. First, using bigger dataset can be a good idea because the first part of the Mapillary dataset is used to train our models,

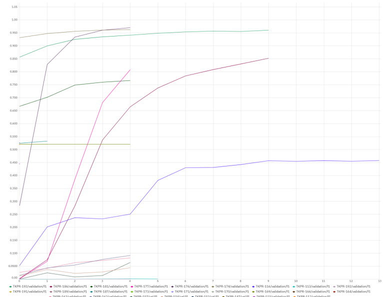

Fig. 7. Validation F1 Score

and there was a class imbalance problem. Including other parts or finding different datasets might solve this problem. Second, applying SMOTE to classical machine learning approach would probably make the models perform better. Our resources for training big classical machine learning models were not enough, therefore we decided to train them on a much smaller sample that includes photos from each class. However, the SMOTE method can be applied to train them on a larger dataset. Third, other large vision models can be tested in the transfer learning part. For this project, we had limited time and resources, therefore we didn't try different models after getting a good performing models with fine tuning ResNet50.

Even though a better and more detailed study can be done, we believe that our work shows that how the classical and more modern approaches performs on the traffic sign image classification problem. In addition, we have achieved good performing models in the process.

VI. Conclusion

In this paper we explored the performance of classical machine learning models, convolutional neural networks and ensemble learning models on traffic sign dataset. The results showed that SGD Classifier had the lowest accuracy scores while CNN model has the highest value. On the other hand, transfer learning gave the most consistent performance having couple of most of the highest accuracy scores in total.

In addition, the results indicated the importance of parameter tuning, data augmentations and solving the class imbalance problem. In every model we captured consistent improvement in performance of the models as a result of the parameter tuning and data augmentations. Also we can see the huge leap in performance on every approach after solving the class imbalance problem.

The outputs can show that the necessity of the comparison of different models in the study. While each model have its strength on the data, the combination of them did not perform the highest in this case. Eventually, the leading performance came from transfer learning.

References

[1] A. Farzipour, O. N. Manzari, and S. B. Shokouhi, "Traffic sign recognition using Local Vision Transformer," arXiv.org, Nov. 11, 2023. https://arxiv.org/abs/2311.06651

[2] A. Z. Kouzani, "Road-Sign Identification using ensemble learning," IEEE Intelligent Vehicles Symposium, pp. 438–443, Jun. 2007, doi: 10.1109/ivs.2007.4290154.

[3] D. Dey, S. Majumder and A. Halder, "Traffic Sign Recognition System using Ensemble based Deep Learning Models," 2023 International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC), Silchar, India, 2023, pp. 1-8, doi: 10.1109/ISACC56298.2023.10083886.

[4] M. A. Vincent, V. K. R and S. P. Mathew, "Traffic Sign Classification Using Deep Neural Network," 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS), Thiruvananthapuram, India, 2020, pp. 13-17, doi: 10.1109/RAICS51191.2020.9332474.

[5] S. Mehta, C. Paunwala and B. Vaidya, "CNN based Traffic Sign Classification using Adam Optimizer," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 1293-1298, doi: 10.1109/ICCS45141.2019.9065537.

[6] Youssouf, N. (2022). Traffic sign classification using CNN and detection using faster-RCNN and YOLOV4. Heliyon, 8(12), e11792. https://doi.org/10.1016/j.heliyon.2022.e11792