CURRENT TRENDS IN COMPUTING
VOL.1 NO.1 PP.44-53 (2023)
ISSN: 2980-3152
Received: 01.05.2023
Accepted: 19.05.2023

NAVIGATR: DETECTING AND RECOGNIZING TURKISH TRAFFIC SIGNS USING A NEW DATASET BASED ON DEEP LEARNING

ERDI TUNA[1], KASIM OZACAR[1*]

[1]*Computer Engineering Department, Karabük Üniversitesi, Karabük, 78050, Türkiye*

Abstract. While cars are becoming smarter than ever with built-in sensing technologies, thanks to the spreading availability of low-cost wearable devices, millions of cars in traffic lack such technologies. However, detecting and recognizing traffic signs is essential in ensuring the safety of pedestrians and drivers. To provide this safety, we conducted a study first to prepare a dataset using collected data in different weather conditions. Then, we used TensorFlow's Object Detection API to detect and recognize traffic signs in Turkey. Initially, we collected over 5000 pieces of data for training. We labeled the data in the dataset using a web-based helper application and selected a suitable deep-learning model. After the training process, we evaluated the results of the models and assessed the quality of our prepared dataset. After training the model, we imported it into an Android application that we developed. This application helps navigate drivers by providing information about the signs in front of their cars using text-to-speech technology.

1. INTRODUCTION

According to the Centers for Disease Control and Prevention [3], 1.35 million people die each year as a result of not following traffic rules. Similarly, there are more than one million road accidents in Turkey every year, with more than a tenth of these resulting in injury or death [2].

Among the advanced technology vehicles produced in recent years, very few vehicles imported to our country have built-in traffic sign recognition. Despite this, even if the traffic signs of the countries are similar to each other, the recognition systems integrated into these vehicles may be insufficient due to their unique traffic signs. Furthermore, most vehicles in our country do not have the ability to recognize traffic signs. For this reason, our study aims to create a system that will detect traffic signs and markers using deep learning, utilizing a new dataset we collected. By doing this, we aim to provide real-time

E-mail address: kasimozacar@karabuk.edu.tr (*).

Key words and phrases. TensorFlow Object Detection API, TTSD, traffic sign detection.

FIGURE 1. Samples from our dataset.

information to drivers about traffic signs through an application installed on a smartphone, which will warn drivers about possible dangers and allow them to take precautions.

A 2-step process is involved in detecting traffic signs. The first step is the detection of traffic signs, and the second step is the classification of traffic signs. Similar to the GTSDB dataset, this study examines signs in 3 main categories: Prohibitory, Mandatory, and Danger. Our dataset consists of 10 classes, and an example of the dataset is shown in Figure 1. We performed model training using TensorFlow's object detection API on Faster R-CNN Inception v2 and SSD MobileNet v2, and we compared the performance of each model. Since the SSD MobileNet model is more lightweight, it was tested in a real environment on a mobile device. As a result of the test we conducted, the model achieved a performance accuracy of 94%. Additionally, the model built with Faster R-CNN Inception v2 provided an accuracy of 99%.

This article is organised as follows. Related work in section 2, data set preparation and labelling in section 3, network structure in section 4, experimental results and research discussion in section 5, and conclusion in section 6.

2. RELATED WORK

Since traffic signal detection and recognition is one of the most important topics of computer vision and artificial intelligence, numerous studies have been conducted to enlighten the darkness in these fields.

Even if traffic signals are similar in different countries, each has specific to their country. Therefore, it's better to analyze the studies conducted in different countries.

In their studies, [11] compared Chinese traffic signs with German traffic signs and presented a new faster and more robust detection algorithm based on a deep convolutional network that decreased the convolutional layers in top layers on top layers YOLOv2, which reduced the computational complexity.

[8] implemented a new algorithm for traffic sign recognition using CNN and compared several CNN architectures. Detections and recognitions were executed for GTSDB [9] and GTSRB [4] datasets in real-time using mobile GPU, resulting in 99.94% of correctly classified images.

[7] proposed a new dataset, which includes more than 80000 images distributed in 164 classes, collected from 6 different European countries for traffic sign classification. They also analyzed and compared the classification performance of 5 CNN architectures. They trained the models with GTSDB and their proposed dataset using NVIDIA GTX 1080ti GPU to compare the dataset with a different network. In the GTSRB and European datasets, while the accuracy obtained without data augmentation was 97.88%, the accuracy obtained by performing data augmentation was 98.99%.

In all these valuable studies, either the ready dataset has been used, or the newly created dataset was unsuitable for traffic signs in Turkey. For this reason, in this study, we created a new dataset and then tested the accuracy of our dataset using two of the most widely used and proven successful deep learning models. In addition, we tested the model we trained on a mobile device and evaluated its real-time performance.

3. Preparing the dataset

This section explains how we collect and label the dataset by the TensorFlow Object Detection API. We followed the procedures in the paper by [5] to prepare the dataset.

3.1. **Dataset Collection.**

In this dataset, we took pictures of traffic signs while we were both in the car and on foot. The horizontal and vertical images were taken at 1536x2048 pixels while the car was in motion. On the other hand, for the pictures taken on foot, we use three different distances: near, medium and far, from different angles and in different weather conditions. The resolution ratio of each image has been cropped to 1024x1024, preserving the aspect ratio of the images taken. The number of images per class is shown in Figure 2. Some of the images have been reduced to a lower resolution. While over 5000 images were collected for the training of the model, over 1200 images were collected for the test.

3.2. **Data tagging.**

We tagged each image to assign each traffic sign to the correct class and enclosed it in a bounding box. We manually tagged the data and used LabelImg [1]. Traffic Sign Types are given in below:

- Option sign for Left turn and Straight
- Give way

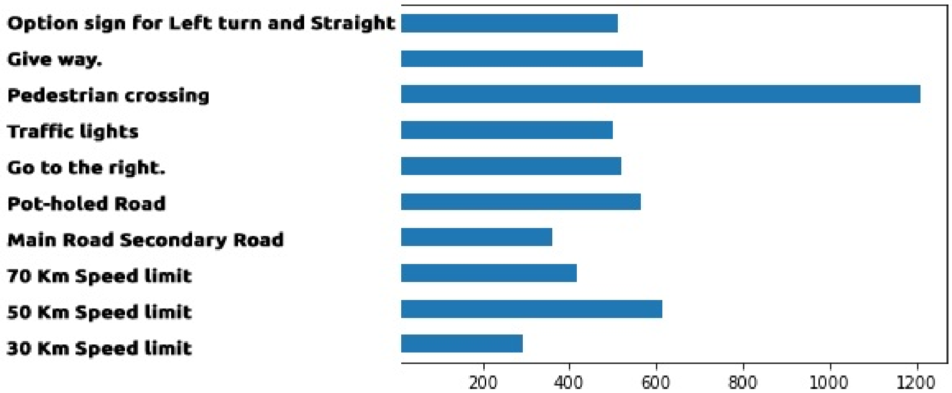


FIGURE 2. Number of pictures by classes.

- Pedestrian crossing
- Traffic lights
- Go to the right
- Pot-holed Road
- Main Road Secondary Road
- 70 Km Speed limit
- 50 Km Speed limit
- 30 Km Speed limit

The traffic signs shown in Figure 3 are enclosed in the bounding box. (Xmin) is the upper left horizontal value, (Ymin) is the upper left vertical value, (Xmax) is the upper right horizontal value and (Ymax) is the lower vertical value of the selected area. Weight and height values are placed in automatic fields. A unique value is assigned to each traffic sign class. As can be seen in Figure 4, an XML file was created for each tagged image. After this process, the image information and class tags of each image are added to a single csv file. In addition, all the image files were added to a RECORD folder. In this way, the dataset was prepared for model training in the TensorFlow Object Detection API.

4. Network Structure

This study used Faster R-CNN Inception and SSD models to train and test.

4.1. **Faster R-CNN Inception v2n.**

This model consists of 2 neural networks, a Region proposal network (RPN) and a detection network [10].

Faster R-CNN uses the RPN as a region proposal algorithm to find the region of interest (ROI) as a bounding box and uses the Non-Maximum Suppression algorithm to select the best region.

	filename	width	height	class	xmin	ymin	xmax	ymax
0	00166.jpg	143	128	Yol Ver	16	12	130	113
1	00170.jpg	67	62	Yol Ver	8	7	61	57
2	00178.jpg	53	49	Yol Ver	7	8	47	45
3	00289.jpg	50	48	Yol Ver	9	8	46	42
4	00426.jpg	63	60	Yol Ver	8	7	60	53
...	...	...	...	...	...	...	...	...
5555	yolver95.JPG	284	246	Yol Ver	12	10	191	174
5556	yolver96.JPG	284	246	Yol Ver	17	11	191	171
5557	yolver97.JPG	284	246	Yol Ver	17	20	189	172
5558	yolver98.JPG	284	246	Yol Ver	17	18	189	174
5559	yolver99.JPG	284	246	Yol Ver	18	17	190	174

FIGURE 3. Sample data.

$$IoU = \frac{AreaofOverlap}{AreaofUnion} \tag{1}$$

If the Intersection over Union (IoU) value given in Equation 1 value is greater than 0.7, it will recognize the traffic signs correctly; otherwise, it won't. An example of IoU for different bounding boxes is given in Figure 5.

4.2. **SSD Mobilnet V2.**

Single-Shot Multi-Box Detection (SSD) MobileNet [6] is a deep neural network architecture implemented specifically for mobile applications due to its lightweight structure. SSD uses a single shot to detect multiple objects in an image. It is based on a feed-forward convolutional network, determines the bounding box using the NMS algorithm, and specifies the accuracy rate. It has been developed by adding extra layers to the standard VGG-16 architecture. These layers become smaller and smaller, allowing detection estimates at multiple scales. The layer for each feature of the convolutional model is different to predict detections. This is simple compared to methods that require architectural objects. All computations are performed on a single mesh by resampling pixels or features. The computations here are performed with ROI, where the predictive value of each trained class and its bounding box are determined. To make these estimates, the IOU (Intersection over Union) is used to calculate the acuity values in the grids.

```
<annotation>
	<folder>train</folder>
	<filename>IMG_6604.JPG</filename>
	<path>..\train\IMG_6604.JPG</path>
	<source>
		<database>Unknown</database>
	</source>
	<size>
		<width>1024</width>
		<height>1024</height>
		<depth>3</depth>
	</size>
	<segmented>0</segmented>
	<object>
		<name>Ana Yol Tali Yol</name>
		<pose>Unspecified</pose>
		<truncated>0</truncated>
		<difficult>0</difficult>
		<bndbox>
			<xmin>402</xmin>
			<ymin>392</ymin>
			<xmax>489</xmax>
			<ymax>453</ymax>
		</bndbox>
	</object>
</annotation>
```


FIGURE 4. XML Structure.

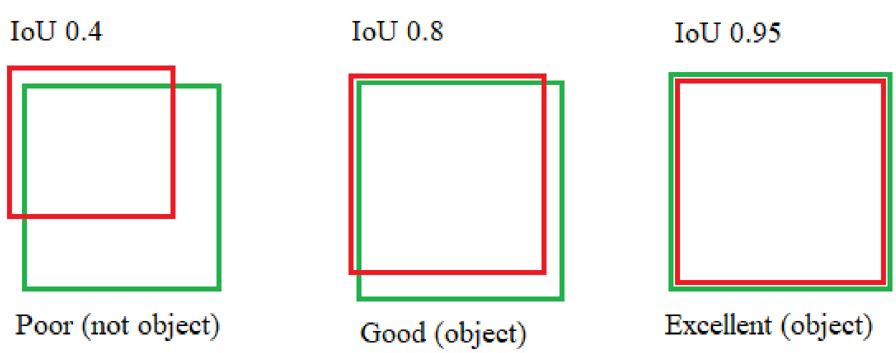


FIGURE 5. An example of IoU for different bounding boxes.

4.3. **Non-Maximum Algorithm (NMS).**

Once the network has been trained, specific boxes can be matched to a single definitive reference box. If the output does not meet the desired threshold, the box will be removed. The post-processing algorithm used in the design of the NMS system combines all detections of the same object. The image is scanned along the direction of the image gradient and set to zero if the pixels are not part of the local maxima. This has the following effect: if the values with the box of the traffic sign are higher than 0.5, they are assigned to its class.

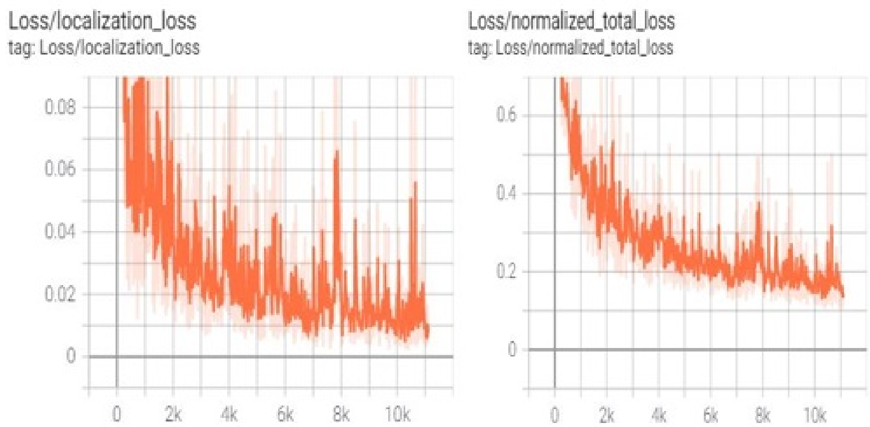
TABLE 1. SSD MobileNet v2 structure.

Type	Input size	Activation function
	1×300×300×3	Relu6
DepthwiseConv2D	**1×150×150×32**	**Relu6**
Conv2D	1×150×150×96	Relu6
Conv2D	1×75×75×144	Relu6
DepthwiseConv2D	**1×75×75×144**	**Relu6**
Conv2D	**1×19×19×384**	**Relu6**
DepthwiseConv2D	1×19×19×576	Relu6
Conv2D	1×10×10×960	Relu6
DepthwiseConv2D	1×10×10×1280	Relu6
Conv2D	1×5×5×512	Relu6
DepthwiseConv2D	1×3×3×256	Relu6
Conv2D	1×3×3×256	Relu6
Conv2D	1×1×1×128	Relu6

5. EXPERIMENTAL RESULTS

Building a machine learning model that recognizes multiple objects in a single image and knows their locations is one of the most challenging areas in computer vision. For this reason, the TensorFlow Object Detection API has been released as open source by Google. In this study, ten different traffic signs, our benchmark dataset (TTDS) and more than 5000 images were trained using TensorFlow with SSD Mobilnet v2 and F-R-CNN Inception v2 models. The study used a GTX 1050 graphics card, Python 3.7, and TensorFlow 1.15. The training of the SSD model took 13000 steps and the training of the model took 12 hours. During training, the classification loss values and the bounding box are shown in Figure6 and Figure7.

For the models evaluated at different numbers of epochs, the validation accuracy seems to have flattened and started to decrease, indicating the presence of overfitting. Consequently, we decided to use augmentation techniques to mitigate the overfitting concerns. The augmentation parameters included a 45° rotation and a horizontal flip.



a) b)

FIGURE 6. Tensorboard Loss values a) localization loss b) normalized total loss

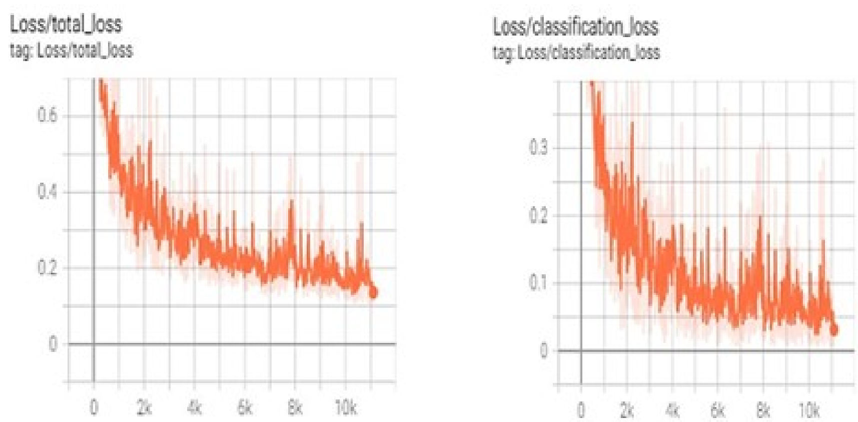


FIGURE 7. Tensorboard Loss values: total loss (left) and classification loss.

After training the SSD model, the developed application was tested with an Android phone. The success rates in traffic sign recognition are shown in Figure 8 as an example. The performance comparison is shown in Table 2. The faster R-CNN Inception v2 model performs much better in terms of accuracy (99%) and speed (35 fps) compared to the opponent.

6. CONCLUSIONS

Detecting and recognizing traffic signs ensures the safety of both pedestrians and drivers. To this end, we conduct a study to first collect a dataset, and then detect and recognize traffic signs in Turkey using

FIGURE 8. Detected Traffic Sign in the mobile app.

TABLE 2. Performance comparison on GTX 1050 GPU.

Model	**Accuracy**	**Avg FPS**
SSD MobileNet v2	%94	35
F-RCNN Inception v2	%99	20

TensorFlow's Object Detection API. In this study, the detection and recognition of traffic signs in Turkey was performed using deep learning methods. We collected more than 5000 images for training and more than 1200 for testing. These images were taken in different weather conditions according to their location in the vehicle and on foot. The performance of the Faster R-CNN Inception v2 and SSD mobileNet v2 models on our dataset was compared using TensorFlow's Object Detection API. The Faster R-CNN Inception v2 model produced accurate, reliable and fast results. After training the SSD MobileNet model, we import the model into an Android application. This application uses text-to-speech to navigate and warn a driver according to the traffic lights ahead.

REFERENCES

[1] labelimg. `https://github.com/tzutalin/labelImg`. [Accessed 01-May-2023].

[2] Tuik. `https://data.tuik.gov.tr/Bulten/Index?p=Road-Traffic-Accident-Statistics-2020-37436`. [Accessed 01-May-2023].

[3] CDC. Road Traffic Injuries and Deaths—A Global Problem — cdc.gov. `https://www.cdc.gov/injury/features/global-road-safety/index.html`. [Accessed 01-May-2023].

[4] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–8. Ieee, 2013.

[5] Irfan Kilic and Galip Aydin. Traffic sign detection and recognition using tensorflow's object detection api with a new benchmark dataset. In *2020 international conference on electrical engineering (ICEE)*, pages 1–5. IEEE, 2020.

[6] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[7] Citlalli Gamez Serna and Yassine Ruichek. Classification of traffic signs: The european dataset. *IEEE Access*, 6:78136–78148, 2018.

[8] Alexander Shustanov and Pavel Yakimov. Cnn design for real-time traffic sign recognition. *Procedia engineering*, 201:718–725, 2017.

[9] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

[10] Marco Magdy William, Pavly Salah Zaki, Bolis Karam Soliman, Kerolos Gamal Alexsan, Maher Mansour, Magdy El-Moursy, and Kerolos Khalil. Traffic signs detection and recognition system using deep learning. In *2019 Ninth international conference on intelligent computing and information systems (ICICIS)*, pages 160–166. IEEE, 2019.

[11] Jianming Zhang, Manting Huang, Xiaokang Jin, and Xudong Li. A real-time chinese traffic sign detection algorithm based on modified yolov2. *Algorithms*, 10(4):127, 2017.