

Road Sign Recognition: Using Machine Learning to Spot Road Signs

Adhya Abi



Research Question

Can machine learning programs be used to teach computerized objects to identify road signs?

Hypothesis

I hypothesize that the machine learning tool *Teachable Machine* can be used to create a program that would allow a computerized object to identify road signs.

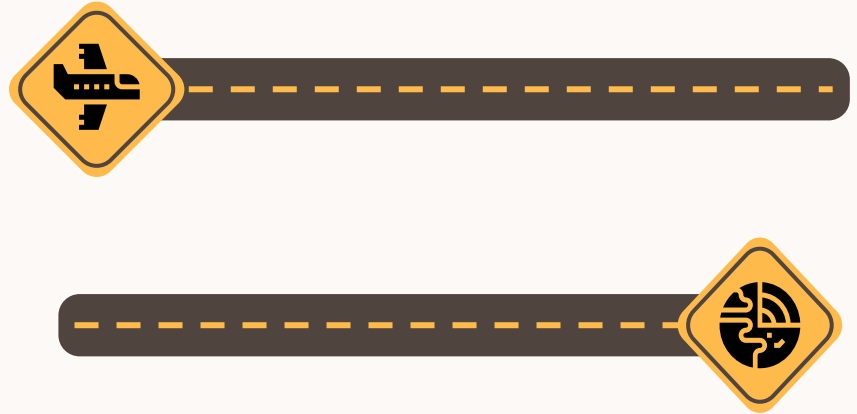
Other information that supports my hypothesis is that many people use such technology everyday for things like facial recognition on their phones and devices. This technology works the same way, just with road signs and not people. I know that this hypothesis is testable because it uses only a computer, phone, and notebook, and all variables of concern can be controlled.



Variables

- ◆ **Independent Variable:** The road sign
- ◆ **Dependent Variable:** Whether or not the computer recognizes the sign
- ◆ **Controlled Variables:**
 - The sign
 - The program
 - The number of pictures

$y = f(x)$, where 'x' is the independent variable, in this situation, and f is a function (basically $y = f(x)$ means the dependant variable 'y', is a function of 'x', the independent variable, so as x changes, so does 'y').



Materials

- 1 mobile phone (for taking pictures)
- 1 computer with internet connection (for *Teachable Machine*)
- 1 notebook or observation and data collection table





Procedures

1. Gather materials.
2. Select two different types of road signs that can be used for your machine learning model. Begin to take pictures of each street sign.
3. Make sure you take at *least* ten pictures of each different street sign that you see. Remember that *Teachable Machine* will crop everything into a square shape, so be careful not to take your pictures too close to the sign, or they might be cropped, which could affect the outcome of the experiment.
4. Upload the pictures into your files on a computer so that you are ready to begin your *Teachable Machine* project.
5. Go to <https://teachablemachine.withgoogle.com/> and click the Get Started button.
6. Click *Image Project* and then *Standard Image Model*.



7. Make the names of the two classes each of the two signs you selected. For example, “stop sign” and “pedestrian crossing sign”.

8. Under the *Add Image Samples*, click the *Upload* button and then upload all the images of the road sign matching to the class. Make sure that you upload an even amount of pictures for each of the two classes.

9. Repeat step 8 for the second class with that road sign’s specific pictures.

10. Click the *Train Model* button. Wait for it to finish training. During this process, the program is learning what and how to identify each sign.

11. After that, click the dropdown arrow next to the *Advanced* sign. The click *Under the hood*.

12. Then read the information that will show up on your screen.

(*Teachable Machine* creates what’s referred to as a “confusion matrix”, which shows how many of the images were categorized correctly, and how many were not.)

13. Try to now test with other additional images. Try to get images that are different from the ones you used before in the training stage. For this stage, you can even search for sign pictures on the internet.

The screenshot shows the Teachable Machine interface with two classes defined:

- stop sign**: 10 Image Samples, Model Trained.
- pedestrian crossing sign**: 10 Image Samples.

Training parameters:

- Epochs: 50
- Batch Size: 16
- Learning Rate: 0.001

Accuracy per class table:

CLASS	ACCURACY	# SAMPLES
stop sign	1.00	2
pedestrian c...	1.00	2

Confusion Matrix:

Class	stop sign	pedestrian c...
stop sign	2	0
pedestrian c...	0	2

14. Under the *Preview* section, change the dropdown menu's selection to *File*.

15. Under the same section, upload a new image, to which the program will assign the probability that it is one of your created classes. For each new image, record in a data table the probability that the program assigns.

16. Now, test the model with as many different pictures of signs as you can, all with different backgrounds and environments, and colors, too!

17. After you have assessed how well your project works by calculating the average, click the *Teachable Machine* logo in the left corner, then click *Save to Google Drive*. Now you can access it whenever you want!

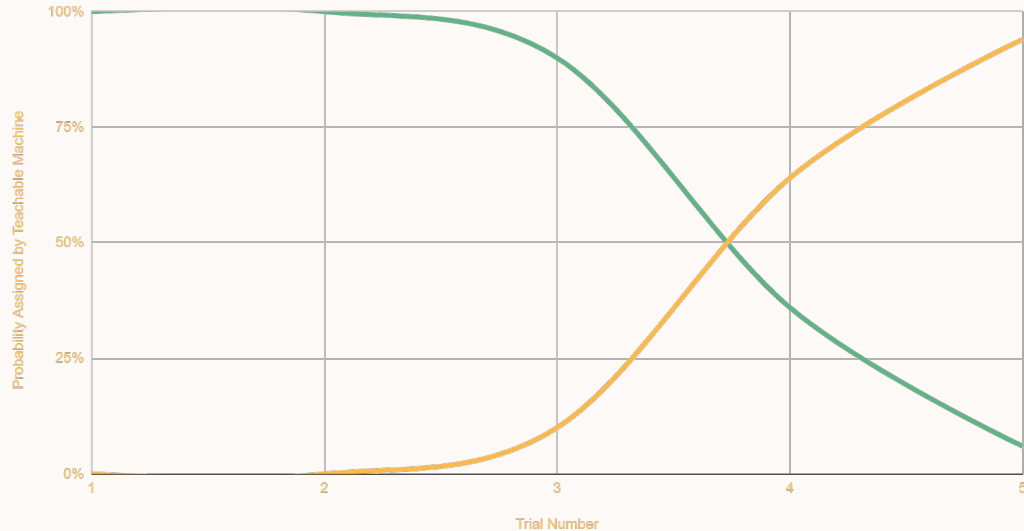
18. Create a New Project after you have saved this one.

19. This time, train the model with all of the training images you had. You might notice that this increases the accuracy of the result.

Data

Probability of Sign Recognition

This chart shows the probability of a sign matching its specific class (stop sign or ped. crossing sign), as assigned by Teachable Mach...




This graph shows the probability of a test image I uploaded being either a pedestrian crossing sign or a stop sign. I saw that the algorithm got the answer correct every time by majority percentage.

Observations

- The algorithm seems to find it harder to recognize stop signs in comparison to pedestrian crossing signs
- All of the probabilities the algorithm assigns are above 50%
- The more similar the environment in the testing picture is to the environment in the pictures the model was trained with, the easier it is for the algorithm to recognize it.

Preview [Export Model](#)

[Import images from Google Drive](#)



↓

Output

stop sign	<div style="width: 50%;"></div>
pede...	<div style="width: 100%;"></div> 100%
C...	<div style="width: 50%;"></div>

Summary

In this experiment, I learned that external sources can be used to train models to recognize road signs, and I believe that the results and the data from my project displayed this.

To preview this model, use the following link:

<https://teachablemachine.withgoogle.com/models/BgeaesJOD>

