# Traffic Sign Detection In Foggy Weather


Aditya Raj
Ahmedabad University
AU1920177

Dhruvanshu Parmar
Ahmedabad University
AU194016



**Abstract—** This project presents a traffic sign detection and recognition system based on a Convolutional neural network. In this project we are using EfficientNetB4 neural network architecture to create a model based on our own custom dataset. Training model in this project involves preprocessing the images, so that it can handle the system even in the foggy weather condition.

This project is focused on two subproblems first, since the data in the foggy condition is hard to get for the training, steps are taken to mitigate the problem by manipulating the images. And the second subproblem is repeating the process of training and testing to achieve the accepted result.


## I. Introduction

In this project we are trying to detect traffic signs and recognize them as well and we are going further in this project and taking care that it can also be recognized in the foggy weather condition. To implement this detection and recognition we need a model which can be trained using machine learning algorithms like SVM(support vector machine) or using deep neural networks. Since we are detecting the traffic sign in foggy conditions, chances of detection become low but we are trying to increase the accuracy as much as we can.

## II. Literature survey

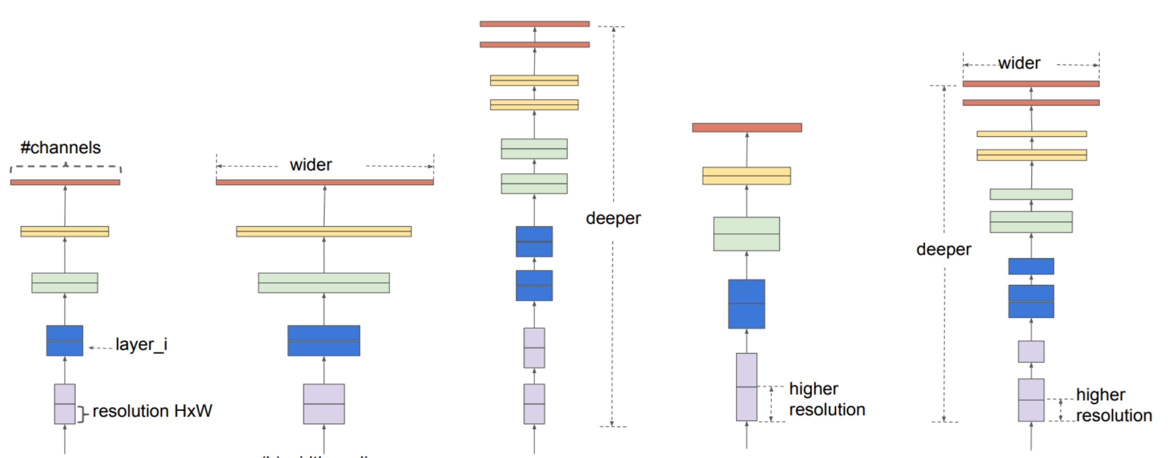
We Explored different Models for object detection which can be fit for our project. The models which we explored are:

- MobileNet
- RetinaNet
- Faster-RCNN
- SqueezeNet
- GhostNet
- SSD (Single Shot detection)

In these Models we are mainly focused on studying the mechanism, model size, and Inference Time of the model. We Come to the conclusion that depending on the depth of the neural network model size and inference time depends.

we are able to explore efficientNet neural network architecture developed by google AI. After Reading the Official Paper on *EfficientNet* [1], we are convinced that we are going to use this architecture in our Project.

**EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks**




(a) baseline (b) width scaling (c) depth scaling (d) resolution scaling (e) compound scaling

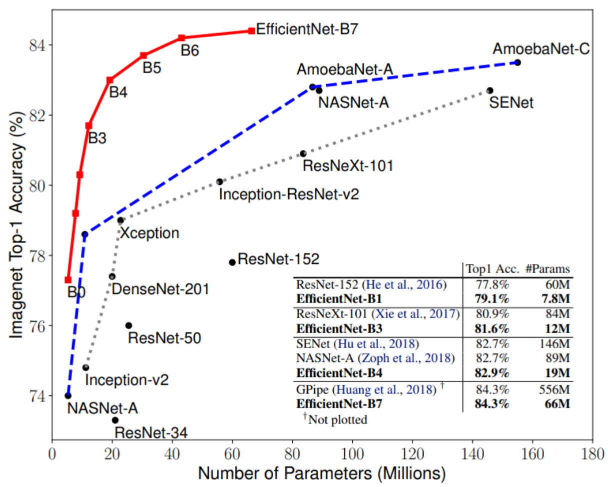



Fig: Model vs. ImageNet Accuracy

In this paper[1] they systematically study model scaling and identify that carefully balancing network depth, width, and resolution can lead to better performance. Based on this observation, they propose a new scaling method that uniformly scales all dimensions of depth/width/resolution using a simple yet highly effective compound coefficient.

They propose a new compound scaling method, which use a compound coefficient φ to uniformly scales network width, depth, and resolution in a principled way:

$$\text{depth: } d = \alpha^{\varphi}$$
$$\text{width: } w = \beta^{\varphi}$$
$$\text{resolution: } r = \gamma^{\varphi}$$
$$\text{s.t. } \alpha \cdot \beta^{2} \cdot \gamma^{2} \approx 2$$
$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

where α, β, γ are constants that can be determined by a small grid search. Intuitively, φ is a user-specified coefficient that controls how many more resources are available for model scaling, while α, β, γ specify how to assign these extra resources to network width, depth, and resolution respectively.

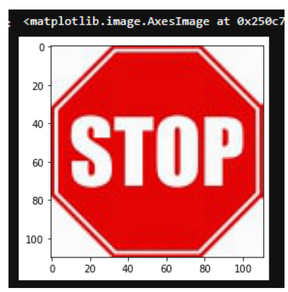
.

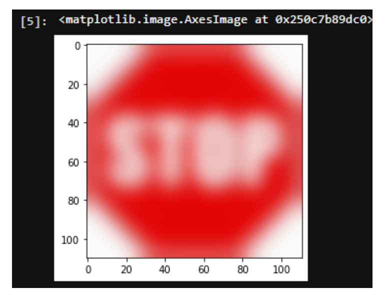
## III. Implementation.

The implementation is done in two parts: the first part of this project involves working with the data. In this part we took half of the dataset to add the noise, taking fogg into consideration. To add the noise in the dataset we have taken advantage of the GaussianBlur, using GaussianBlur we are trying to imitate the foggy condindition and with the manipulation of kernel size we tried to manipulate the level of visibility in the fogg. Increasing the Kernel size decreases the visibility in the image. And then the next step is to prepare the dataset for input in the model, since our data do not need a bounding box because it is already cropped, we will use image name and image label for the processing. The second part is training, we are using google colab for the training and google drive for saving the data. Training a model comes with its own sets of problems like deciding the number of epochs for the training dataset, overfitting the model, and this consumes lots of time.

## IV. Results

In the above image we can see that one image is blurred and the other is original. The blurred image is used in order to




compensate for foggy weather images. In this image kernel size of 31*31 is used.

## V. Conclusion

After training for 50 epochs in the example code, the accuracy of all the signs are around 40% but the problem here is not accuray instead there is very less detection of traffic signs. Hence, we come to the conclusion that Gaussian blur is not the key for solving the artificial fog problem.

## References

[1] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in International conference on machine learning PMLR, 2019, pp. 6105–6114.

[2]R. Belaroussi and D. Gruyer, "Impact of reduced visibility from fog on traffic sign detection," 2014 IEEE Intelligent Vehicles Symposium Proceedings, 2014, pp. 1302-1306, doi: 10.1109/IVS.2014.6856