Developing A Convolutional Network for the GTSRB Dataset

Pranav Chandupatla
Department of Computer Science
The University of Texas at Austin
Austin, TX 78712
pranav.chandupatla@gmail.com

Abstract

We trained a convolutional neural network to classify street sign images in the German Traffic Sign Benchmark (GTSRB) dataset. On the test data, we achieved a 96.081% accuracy rate. Following the creation and analysis of the network, we performed two experiments. The first of these experiments was an attempt to use gradient visualization to understand what features of an input image the model used to classify the image. Through this process, the model was found to focus on sign shape and content, with varying degrees of focus on each factor given a classification image. The second experiment analyzed the relationship between the depth of a convolutional layer, the size of the kernel at that depth, and the resultant testing accuracy rate. The conclusions of this experiment were largely indeterminate and more testing should be done to determine the exact relationship.

1 Background

In class, the concept of convolutional neural networks was introduced as an intuitive way to solve computer vision problems such as classification. Though Homework 2 briefly covered the creation of image classification neural networks, the objective of these models was to identify relatively simple, black-and-white images of clothing. We wanted to explore image classification further into color images in which the object of interest was not guaranteed to be centered within the frame. Through the creation of a new model, we hoped to see how complex image classification tasks could be solved with deep convolutional networks. To these ends, we elected to create a convolutional model to classify the images found in the German Traffic Sign Benchmark (GTSRB) dataset.

1.1 GTSRB Dataset

The GTSRB dataset is a publicly available image database comprised of German street signs. There are 43 variants of signs, each with a different appearance and meaning. The training dataset contains 39,209 images while the testing set contains 12,630 images. A validation set containing 3,920 images was created from the training set for the purpose of model training.

1.2 Model Basis

Ideas for creating the model were drawn from the work of Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton at the University of Toronto. Their paper on ImageNet Classification and the model described therein was used as the basis for several ideas in our constructed model, including the use of max pooling, dropout, and increased model depth.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Model Creation and Training

2.1 Model Description

Our model was based on AlexNet, a convolutional neural network created by Krizhevsky et al. We began by transforming the input images to a size of 112 x 112. This image size was largely chosen due to system constraints; increasing the image size to include more detail caused a large slow down in the training process. The transformed images were then fed into a series of four layers, each of which comprised a convolution with a size 4 kernel, max pooling with a size 2 kernel, and a ReLU. The outputs of these layers were then flattened and fed through a series of dense and linear layers, yielding a final output of size $(B, 43)$, where B designates the batch size. See Figure 1 for a complete diagram of the model.

2.1.1 Model Justification

Convolutional layers were used to learn the image features necessary to classify the image. The kernel size of 4 was was chosen using a grid search based on minimizing training loss and training time. Larger kernel sizes were found to both reduce accuracy and greatly increase execution time. The first and last convolutional layer used a stride of 2 to further reduce the image size beyond $(112, 112)$. This was again done to decrease run-time to a reasonable level; we found that convolving the pixels together such that less data points were fed through each layer was an effective method of reducing run-time. The difficulty with this method was that reducing convolving too many data points together would significantly increase training loss. Grid search was again used to determine the optimal stride. We used less convolutional/max pooling layers than AlexNet becaus we believed that the increased depth was unnecessary. AlexNet was developed to classify disparate images; our model was meant to classify similar images. Therefore, we felt that the potentially increased accuracy of an additional layer was not enough to compensate for the greatly increased treaining time that including it would bring. Dropout was used as a regularization method to reduce overfitting in the dense linear and ReLU layers. Two dense layers were used, as one dense layer resulted in a loss plateau after 5 epochs.

2.2 Training Process

Training was conducted using a training and validation dataset. The training set was comprised of 35,289 images while the validation set contained 3,920. The overall process consisted of 15 epochs at a learning rate of .001 using categorical cross entropy loss. In each epoch, batches of size 200 were run through the model. Over the course of training, training loss fell from 2.184 to 0.042, while validation loss fell from 0.881 to 0.031, resulting in a final testing accuracy of 96.081% (Figure 2, 3). The epochs and learning rate used were derived using grid search.

3 Gradient Visualization

3.1 Background

Gradient visualization is a technique used to determine what parts of an input are computationally important to the model. Areas of the image that create large magnitude gradients are said to be more significant to the resulting classification than areas with smaller gradients. We wanted to run a gradient visualization process on an image from each class label to better understand how the model was classifying the images; determining which parts of the image were relevant to classification across each label would help us determine this.

3.2 Hypothesis and Procedure

We hypothesized that the relevant portions of each image class would be in the broad areas of shape and content, with special emphasis on the content. As the shape of the signs is similar across many image classes, we determined that this feature should not be as expressive.

To create a visualization of input image gradients, we ran an image from each class label through our trained model and saved the computed gradients. We then normalized these gradients to be within the range $[0, 1)$ to enable visualization. These normalized gradients were then visualized using the Img library.

3.3 Results and Conclusions

Our hypothesis was partly supported by our results. For circular signs, the significant features were largely the content of the sign. For non-circular signs (diamond, triangle, etc.), the shape was as important as the content. Overall, the model seemed to emphasize the details necessary to differentiate the image from other, similar labels. For example, on speed limit signs, the model would often focus on only the first digit of the sign, as the second digit was always 0 (Figure 4). On uniquely shaped signs, the visualizations would often show more emphasis on the shape than on circular signs, with an equal focus on the content of the sign (Figure 5). From our results, we concluded that the important features for each image could be somewhat dependent on other images in the training set. If the images were very similar with few differentiating factors, the shape of the sign was less important. On the other hand, if the images were very different, the features became more shape- and content-focused.

4 Depth, Kernel Size, and Accuracy

4.1 Background

The importance of specific layers in determining the overall accuracy of a convolutional model is difficult to determine with simple inspection. On one hand, it could be that shallower layers are more significant, as they could be learning the general details of an image. On the other hand, there is a case to be made that deeper layers are more relevant because they are learning the fine details that allow the model to distinguish one class from another. Additionally, the kernel size of the chosen convolutional layer is an important determinant of accuracy. We wanted to see how depth and kernel size interacted to produce effects on the overall accuracy of our model's testing accuracy.

4.2 Hypothesis and Procedure

We hypothesized that making changes in kernel size to shallower layers would produce greater effects on accuracy than the same changes made on deeper layers. Our reasoning lay in the assumption that shallower layers were the most important to learning the general features of each class of image while deeper layers learned fine details that would enhance these base predictions. To determine the effects of kernel size and depth on testing accuracy, we measured the model's testing accuracy at a number of varying depth and kernel sizes (Figure 6). We then determined the magnitude of difference between the normal model's accuracy and the experimental accuracy (Figure 7). Though we wished to run multiple trials, we found that each experiment could take up to a day of computing time on our machines making the prospect of multiple trials impractical given our current resources.

4.3 Results and Conclusions

Our hypothesis was not supported by our results. As we used the kernel size of 32 across each layer, the magnitude of change spiked then steeply dropped before rising again. This trend was not consistent with our expectation of a declining effect on accuracy as the change was made in deeper layers. The kernel size of 2 somewhat supported our hypothesis, with an overall negative correlation between layer depth and change in testing accuracy. This, however, is a tenuous correlation at best. We concluded that each layer may have a "Goldilocks" kernel size, and that the depth is but one factor that determines this optimal size. Additionally, we determined that perhaps shallower layers need smaller kernels to preserve data while deeper layers need larger kernels to better combine learned feature information across the image. We concluded that more trials should be conducted before any strong conclusions are drawn.

5 Conclusion

Over the course of creating model, we learned how convolutional neural networks could be used on more complex images with a variety of orientations and classes. Through gradient visualization, we were able to glimpse into how the model was making its decisions, and we determined that shape and content were some of the key factors the model used to classify images. Finally, by modifying kernel

sizes across a variety of layer depths, we found that the relationship between kernel size, layer depth, and testing accuracy was not as simple as we had initially thought.

6 Figures

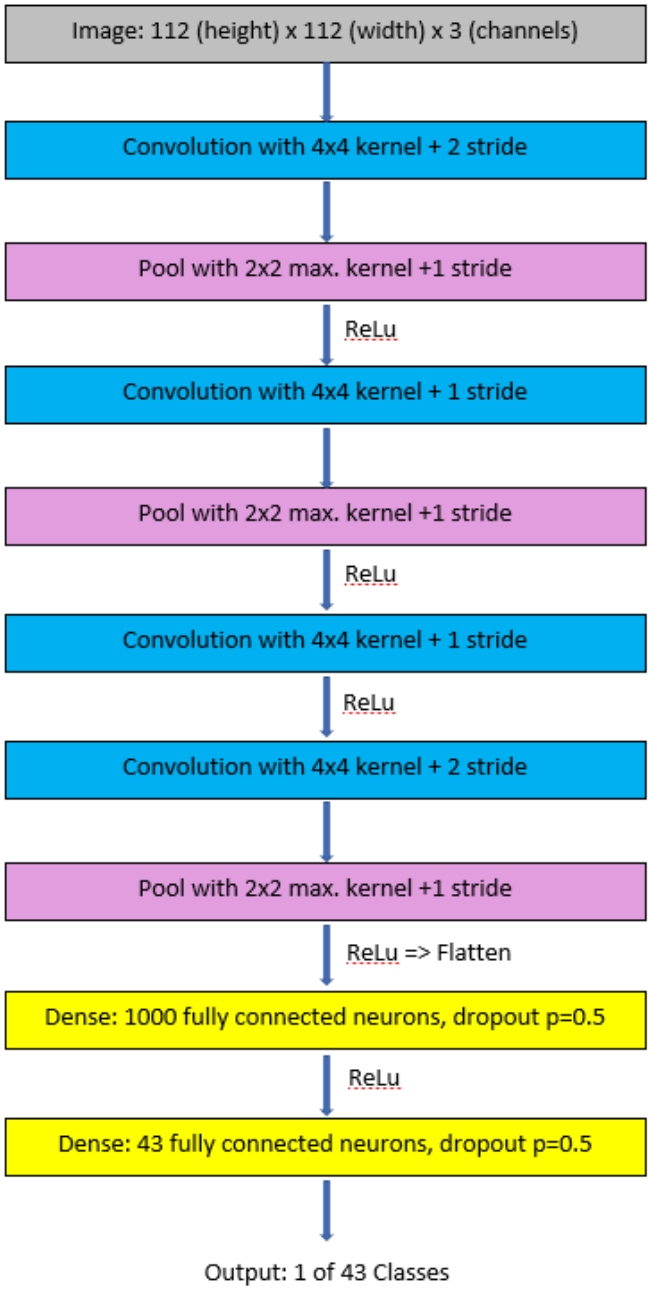

Figure 1: Model Visualization

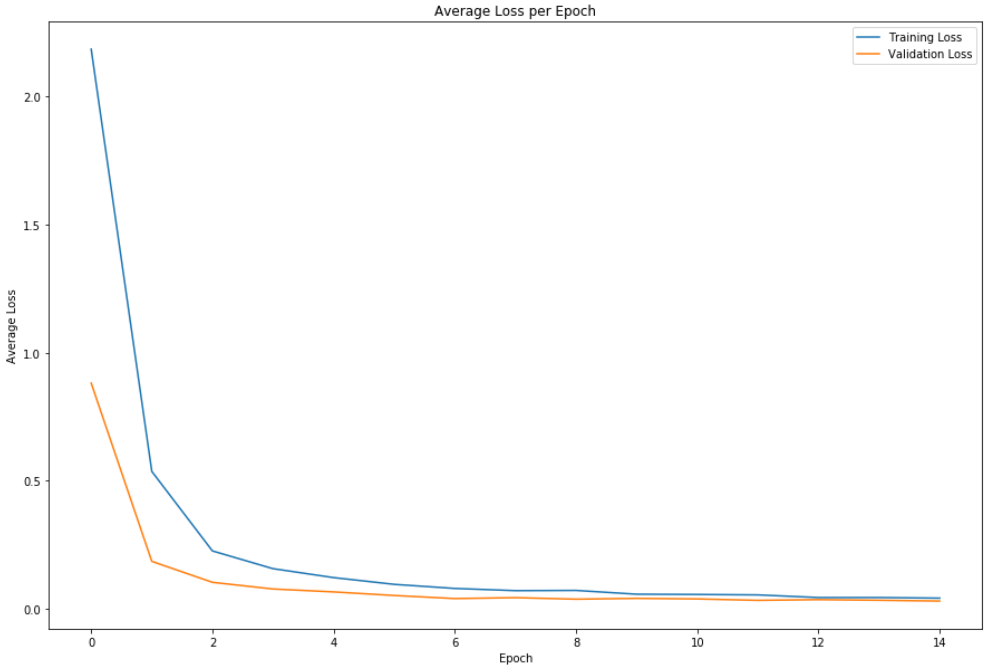

Figure 2: Training and Validation Loss

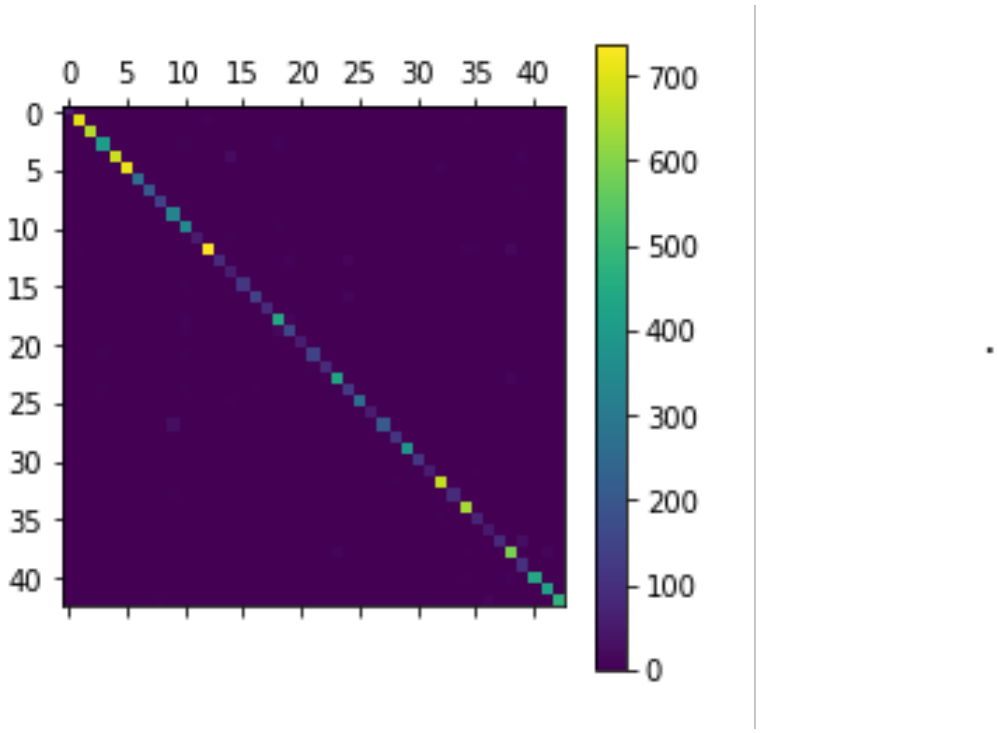

Figure 3: Contingency Matrix

Figure 4: 20 Speed Limit Gradient Visualization

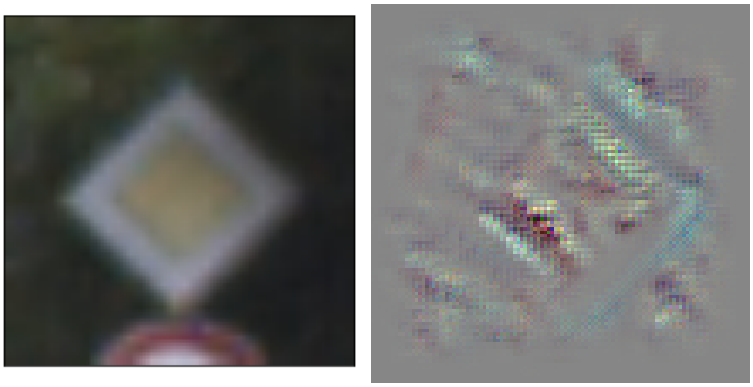

Figure 5: Yellow Diamond Sign Gradient Visualization

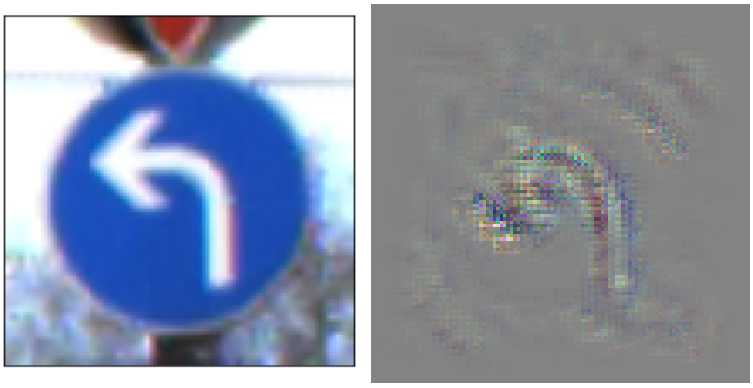

Figure 6: Left Turn Gradient Visualization

Size of Kernel	Depth of Conv Layer 1	2	3	4	Trend Line
2	94.703%	93.761%	94.078%	96.215%	
4	96.081%	96.081%	96.081%	96.081%	
32	95.661%	93.397%	95.687%	95.067%	

Figure 7: Testing Accuracy as a Function of Layer Depth and Kernel Size

Size of Kernel	Depth of Conv Layer				
	1	*2*	*3*	*4*	*Trend Line*
2	1.378%	2.320%	2.003%	0.134%	
4	0.000%	0.000%	0.000%	0.000%	
32	0.420%	2.684%	0.394%	1.014%	

Figure 8: Change in Magnitude of Testing Accuracy as a Function of Layer Depth and Kernel Size

7 References

[1] Krizhevsky Alex, Sutskever Ilya, and Hinton Geoffrey E.. 2012. ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25, F. Pereira, Burges C. J. C., L. Bottou, and Weinberger K. Q. (Eds.). Curran Associates, Inc., 1097–1105.