**CISC 453 Final Report**

**Garett MacGowan - 10197107 - 15gm1@queensu.ca**

**Perceptual Importance Alignment Using Machine Learning to Improve AI Safety**

In July of 2016, researchers from Google Brain, UC Berkeley, Stanford University, and OpenAI released a collaborative paper titled *Concrete Problems in AI Safety*. The goal of the paper was to outline a safety critical set of problems in AI for which solutions can be developed. In this paper, I will discuss one of the problems outlined in *Concrete Problems in AI Safety*- robustness to distributional shift. I will propose a partial solution, one which is touched upon in the paper, and describe my implementation. Finally, I will discuss the scalability of the approach, as well as some problems it may have. For the remainder of this paper, I will refer to the *Concrete Problems in AI Safety* paper as CPAIS.

Robustness to distributional shift, as CPAIS describes it, is the problem of ensuring that an AI agent recognizes, and behaves robustly, when in environments that are different from their training environments. It is useful to demonstrate the problem of robustness to distributional shift by an example from which the remainder of this paper will be based. Consider a scenario where you are driving a car. In regular conditions, you would expect that you would be able to discern between a fifty kilometer per hour sign and an eighty kilometer per hour sign. However, if the conditions were poor, for example, in the case of a snow storm, it may be difficult to discern between them. In this case, the distributional shift is the snow storm and its effect on the visual interpretation of the stop sign.

As humans, context plays a huge role in our perception. If you were on a country road, you would be more likely to interpret the sign as eighty kilometers per hour. If you were in a city, you would be more likely to interpret it as fifty kilometers per hour. Ignoring all of this, your uncertainty would cause you to slow down in these conditions. This is not true for most AI systems. The problem of robustness to distributional shift has two main components; recognizing incompetence and making inference with incomplete information. Recognizing incompetence refers to the uncertainty of our interpretation of the sign. Making inference with incomplete information refers to the ability to discern the meaning of the sign with a partially obfuscated or altered view. Many AI systems lack the ability to recognize incompetence and make inference with incomplete information. In the remainder of this paper, I would like to focus on the problem of making inference with incomplete information, as it lends itself well to current AI implementations which tend to be vertically integrated with their domains. That is, many current AI implementations only make inference based on a small subset of the total available information space.

Making inference with incomplete information is critically important for the success of AI systems. Humans do not generally mistake a stop sign with stickers on it as a forty-five kilometer per hour sign, however, some current AI systems do. In a United States senate hearing in June of 2018, it was brought to the attention of the public that it was possible to trick a modern computer vision system into believing that a stop sign was a forty-five kilometer per hour sign by the simple application of some stickers. This is really bad news for the safety of self driving vehicles. Firstly, it shows that a malicious actor could easily manipulate a traffic regulation device and cause a crash, and secondly, it showcases that our machine learning agents do not interpret the visual components of the world in in the same way that humans do. The tricky stop sign was very clearly still a stop sign. The stickers were no more than a little bit of ‘stuff’ on the image, and could have easily been graffiti, dirt, or some other abnormality.

There are two components to perceiving a stop sign. The first is a symbolic representation; they are generally placed on the intersection of some path where it is beneficial to stop. The second is the stop sign’s physical representation, that it is octagonal, red, and has white text that says ‘stop’. The physical and symbolic components of a stop sign are not straight forward. For instance, imagine showing someone an octagonal red sign without any words on it. If you asked the person what their first thought was when they saw it, they would probably say ‘stop sign’. This answer is highly dependant on context, and whether they have memory of any other red octagonal shapes with different meanings. The split-second accurate inferencing that humans can make from incomplete information is what AI needs to be able to match. What it means to be a stop sign is a dimensionality reduction problem at its core. It can be translated to a few true and false statements. Is the sign at an intersection? Is the sign red? Is the sign octagonal? Does the sign have the words ‘stop’ on it? Even if one of these components is not true, the likelihood of the sign being a stop sign is still very high. For AI to be able to make inferences as accurately as humans, it needs to be able to align its understanding of the importance of different features with our understanding of the importance of different features.

One potential solution to the problem of robustness to distributional shift, and thus, perceptual importance alignment, is briefly discussed in CPAIS. CPAIS suggests that it may be helpful to train agents on multiple distributions. The hope is that a model that is trained and works well on multiple distributions will also work well on novel distributions. In the example of the stop sign, training a network on images of stop signs with stickers, as well as regular images, will hopefully allow the network to be more robust against other abnormalities. Through my desire to test this theory, I have created a supplementary codebase which is meant to accompany this essay as proof of the findings I am about to discuss.

My codebase investigates robustness to distributional shift through the above problem of street sign recognition. The first thing to do when creating a machine learning solution is to consider the data. In this case, and in most cases, the data is the key to the effectiveness of the system. The data is what is used to teach the network the features that are relevant to the problem. For this purpose, I have harnessed the BelgiumTS dataset. Specifically, I used the BelgiumTS for Classification dataset since it contains cropped and labeled images of many types of street signs. My code contains two main components and one supplementary one. The first component, dataNoising.py is a python script which is meant to take the original training data and add noise, creating a more varying distribution. The second component main.py is a python script which creates two different convolutional neural networks, one with the original training data, and one with the new training data with noise added. These networks are compared to reveal the effectiveness of the strategy. Finally, the supplementary component inspectImages.py is available to view the images in the old and new datasets.

The first component dataNoising.py is the most important item in the codebase. It is used to create a new dataset of images by adding noise to the original dataset. The noise which is added to the dataset is analogous to adding stickers to street signs. I approached noising by creating random rectangular shapes of a random volume and random color, placing them randomly on the images. The volume of a single component of noise is limited so that it does not block out the original content which needs to be classified. The thought behind this noising strategy is that the neural network will be forced to learn a more robust set of features that define each street sign. It theoretically accomplishes this because it cannot rely on memorizing a small set of local features which may be covered up by noise in the remainder of the dataset.

The second most important component main.py contains the convolutional neural network I designed to classify the street signs. The implementation here is not too important. It essentially uses a series of convolutional layers, batch normalization, and max pooling layers, followed by fully connected layers and a SoftMax output activation function. It uses the Adam optimizer for stochastic gradient descent and sparse categorical cross entropy as the loss function since the class labels are not one-hot encoded. The architecture was developed with trial and error and trained on a Nvidia GTX 780 3gb GPU. What is important here is that it allows the user to train two neural networks, one on the original un-noised dataset, and one on the new noised dataset. The networks are evaluated on testing sets from both the noised and un-noised datasets.

Results from my experiments were quite interesting. Models trained on the noised dataset performed worse (96.03% accuracy) than the models trained on the original dataset (97.42%) when tested on the un-noised (original) test set. However, when tested on the noised test set, the model trained on the noised dataset performed better (94.02%) than the model trained on the original dataset (92.47%). These results hint that there may be some benefit to training on noise added datasets, but it is not definitive proof. Given more time, I may have been able to tweak the parameters of dataNoising.py to produce more differentiated results, but I think that some important things can be noted from these results. Training on datasets with noisy samples will allow the network to perform better when given noisy samples in the real world, however, it also seems to reduce the overall performance of the network. This trade-off may be able to be overcome with a deeper network, a different noising strategy, or a different noisy-to-un-noisy training distribution. It may be more beneficial to pursue strategies which trade peak performance with robustness to distributional shift because it is more important that a model is predictable. A model that is robust can be worked on to improve its peak performance, but if a model has a good peak performance and poor robustness, then it is much less valuable in the real world.

What is nice about this method of producing robust models is that it can be easily automated. Existing data can be run through a noising function in a similar manner to how I have done it, and models can be retrained with the new data. This means that if the method is tweaked and produces quantifiable benefits, it can be easily scaled up and applied to all existing domains of computer vision. Admittedly, this method of data noising is more of a band-aid solution to the problem of robustness to distributional shift. If you remember back to the beginning of this paper, I stated that I would focus on the problem of making inference with incomplete information, which is what I have discussed so far. I have not made any contributions to the problem of recognizing incompetence. This problem is not to be left aside as creating solutions for it will make AI agents much more effective.

This leads me to the fact that this paper is only a small component of a potential solution to the problem of robustness to distributional shift. Furthermore, robustness to distributional shift in the way that I have discussed it is only a small component to the problem of perception. There are many complexities to human visual perception that were omitted from this paper for which we have barely begun exploring with machine learning equivalents. There is so much more work to be done in the field of AI, and it will be a long time before fully integrated intelligent systems are developed. For now, the best thing to do is further the development of our AI systems in ways which keep them safe and beneficial to society.