­

Vanderbilt University Institute for Software Integrated Systems

PIRE Project

Stratification of Deep Neural Networks for Autonomous Driving

Author: Grant Bowlds

Advisors: Dimitrios Boursinos, Dr. Martin Fränzle, Dr. Willem Hagemann, Dr. Xenofon Koutsoukos

**Motivation:**

Recent developments in neural networks have brought society closer to the reality of full autonomous driving, but they are still far from perfect. Convolutional neural networks are the backbone of the computer vision techniques autonomous vehicles use to detect objects on the road. Certain variables such as time of day and weather can impede these networks’ ability to correctly classify objects and significantly reduce the safety of using these vehicles.

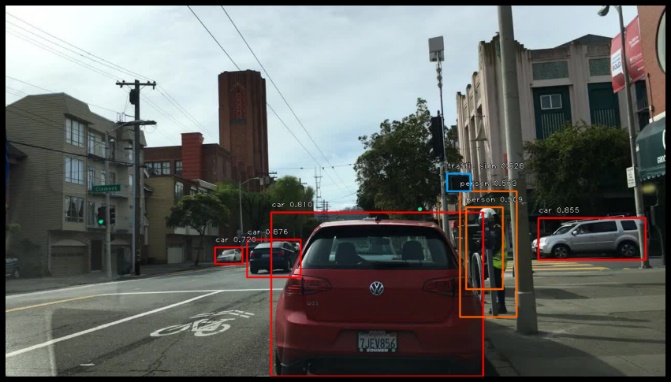
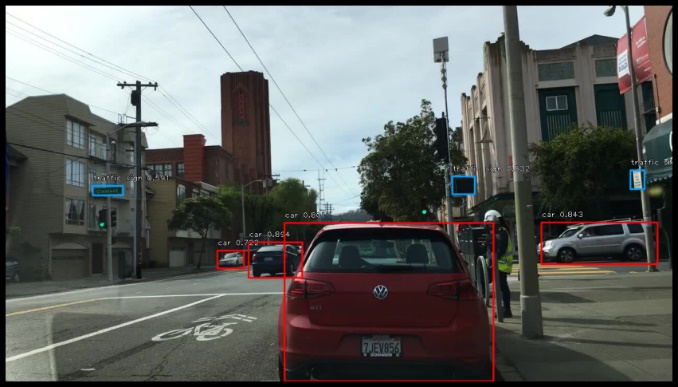
This project consisted of two main parts. The first task was to create both stratified and unstratified neural networks to classify objects. Once these were created, they are evaluated on the validation set to verify that networks stratified off weather condition and time of day do in fact perform better than an unstratified network. Once this is done, a neural network was created to determine the weather condition of an image. Since the appearance of a weather condition can vary widely, this network is then applied to a triplet network, and predictions are generated using Inductive Conformal Prediction on a computed confidence interval.

**Data:**

The dataset used for this project is Berkeley Deep Drive 100k. The dataset consists of 100,000 images of roadways taken from a 4k resolution camera mounted to the windshield of a vehicle. The images are of size 1280x720. The images are pre-split into training, validation, and test sets using a 70/20/10 split. The training and validation sets include a JSON file containing bounding boxes for objects, polygonal masks for driveable areas, and labels for weather, time of day, and scene. To fit into the memory of the GPU, the images were resized to 224x224. To train the neural network for object detection, a CSV file was generated containing the image path, the four coordinates of a single bounding box, and the label of the object. A CSV was generated for each weather condition

**Neural Networks for Object Detection:**

The neural networks for object detection were generated using the keras-retinanet framework. This framework adapts the ResNet50 model to perform multiple object classification. To reduce the variability between networks, each network was trained on one epoch of 10,000 random bounding boxes. Models were generated for Unstratified, Clear-Daytime, Overcast-Daytime, Partly Cloudy-Daytime, Rainy-Daytime, and Snowy-Daytime as these were the only subsets with greater than 10,000 bounding boxes. This method resulted in underfit models but gives a good prediction of how much better the stratified networks would perform if there was more data present. The models were evaluated by calculating the number of correct predictions over the number of correct predictions plus incorrect prediction plus missed predictions. A correct prediction was considered any bounding box with the correct label that overlapped the given bounding box. Small objects were frequently missed due to the resizing of images. On average, the stratified network performed about five percent better.

A car driving on a city street

Description automatically generatedA car driving on a city street

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | **On Stratified Network** | **On Unstratified Network** |
| Clear | mAP using the weighted average of precisions among classes: 0.3866 | mAP using the weighted average of precisions among classes: 0.3783 |
|  | mAP: 0.1339 | mAP: 0.1369 |
|  |  |  |
| Overcast | mAP using the weighted average of precisions among classes: 0.3962 | mAP using the weighted average of precisions among classes: 0.3879 |
|  | mAP: 0.1527 | mAP: 0.1459 |
|  |  |  |
| Partly Cloudy | mAP using the weighted average of precisions among classes: 0.4126 | mAP using the weighted average of precisions among classes: 0.3860 |
|  | mAP: 0.1611 | mAP: 0.1555 |
|  |  |  |
| Rainy | mAP using the weighted average of precisions among classes: 0.3812 | mAP using the weighted average of precisions among classes: 0.3651 |
|  | mAP: 0.1485 | mAP: 0.1427 |
|  |  |  |
| Snowy | mAP using the weighted average of precisions among classes: 0.3879 | mAP using the weighted average of precisions among classes: 0.3676 |
|  | mAP: 0.1464 | mAP: 0.1415 |

Top left: image from BDD100k dataset

Top right: image with mask and labels

Bottom left: object predictions from unstratified network

Bottom right: object predictions from stratified network

**Base Network for Weather Classification:**

The next task was to build a network for weather classification. This was done in keras by using transfer learning on ResNet50 with the ImageNet weights. After about 10 epochs of several thousand images, the network reached about 70% accuracy on both the training and validation sets. While this is not fantastic, weather classification is a notoriously difficult problem. At 224x224, the images do not have many differences, and class imbalances can cause prediction errors.

**Triplet Network:**

A picture containing drawing

Description automatically generatedAfter the base network was built, the weights were loaded into the triplet network framework developed by Dimitrious Boursinos and Dr. Xenofon Koutsoukos. This network consists of three copies of the same network that are then joined together with an additional layer to generate a class prediction. The network takes three inputs: the base image, a positive anchor (another image of the same class), and a negative anchor (an image of another class). Initially, the network is trained by generating random triplet pairs. Once the network is trained, it can generate embeddings for final predictions. The following confusion matrix and T-SNE plots were generated from the embeddings.

A close up of a logo

Description automatically generated

A picture containing rain

Description automatically generated

A close up of a logo

Description automatically generatedA picture containing drawing, clock

Description automatically generatedAfter the random training, the network can be trained on mined pairs. The method for mining searches for positive anchors that are further from the base input than the negative anchor. This results in difficult examples for the model to learn, and ultimately results in better training. The following confusion matrix and T-SNE plots were generated from the embeddings after mined training.

A picture containing rain

Description automatically generated

After generating the embeddings, an optimal confidence interval can be calculated to generate sets of predictions. At a confidence of around 0.25, a range of 1-3 predicitons are generated for the images in the test set.

**Next Steps:**

Next steps for this project include creating a software that collect image input, passes it to the triplet network, and then passes it to all the stratified networks that the triplet predicts. Then, the program would take the union of the objects identified by the networks and take the safer prediction if there are any discrepencies. In addition to this, the models should be recreated, hopefully with more data and be better fit to the validation sets.