Distracted Driver Detection with Convolution Neural Network

Liu, Shijie | Dr. Liu, Xinlian | Computer Science | Hood College | Frederick, Maryland, USA | sl23@hood.edu

ABSTRACT

Safety has become more and more important with the rapidly development of highway infrastructure around the world. With the assessment of road safety in 178 countries, the World Health Organization (WHO) had reported there are approximately 1.3 million people die each year on the world's roads, and between 20 and 50 million sustain non-fatal injuries [4]. According to National Highway Traffic Safety Administration (NHTSA), in 2018, over 2,800 people were killed and an estimated 400,000 were injured in crashes involving a distracted driver in the U.S [3]. The Centers Disease Control and Prevention (CDC) Transportation Safety also defined three types of distraction: eyes off the road as visual distraction, hands off the wheel as manual distraction, and minds off driving as cognitive distraction [1]. State Farm, an insurance company had collected a dataset of 2D dashboard camera images that are properly labeled with 9 distraction classes identified, and the dataset is available on Kaggle for competition [2]. Therefore, this project aims to build and train convolutional neural network models that can detect and classify distracted drivers, which includes a convnet from scratch and a fine-tuned VGG-19 network. Throughout this project, techniques like image augmentation, batch normalization, and dropout are used to overcome the overfitting issue. At end of the project, the best model can achieve 87% prediction accuracy with a small labeled dataset that the model never seen before.

KEYWORDS: Deep Learning, Computer Vision, Convolutional Neural Network, Transfer Learning, Feature Extraction, Fine tuning, Deep Neural Network, Image Augmentation, Batch Normalization, Distracted Driver

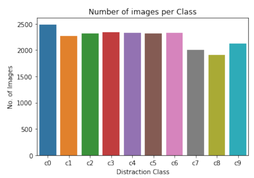
DATA DESCRIPTION

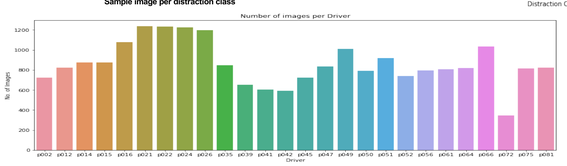
State Farm Distracted Driver Detection dataset [2] has been chosen for this project. The dataset is only available for competition that all image metadata (creation dates) has been removed. Drivers included in this dataset have been separated to train and test data and one driver can only appear in one set of data, which ensures the test data are 100% unknown to the trained model when performing prediction. All image data are collected from 2D dashboard cameras that captures different distraction activities of the driver, such as texting, talking on the phone, drinking, and reaching behind, etc. Below is a list of classes this dataset supports:

- c0: safe driving
- c1: texting - right
- c2: talking on the phone - right
- c3: texting - left
- c4: talking on the phone - left
- c5: operating the radio
- c6: drinking
- c7: reaching behind
- c8: hair and makeup
- c9: talking to passenger

DATA EXPLORATION

Sample image per distraction class





DATA PREPROCESSING

Image Augmentation

- Generates more data from existing data
- Exposes more aspects of same data
- Supports random on-the-fly transformation

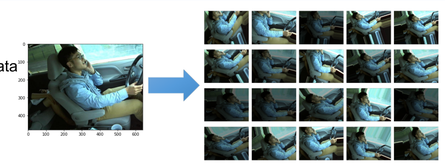
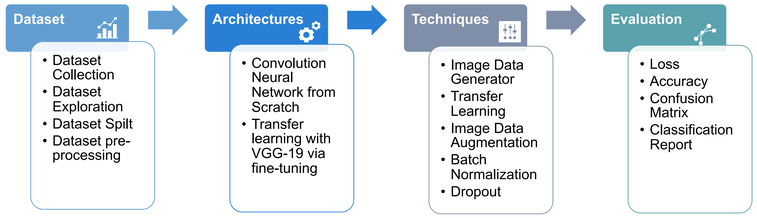


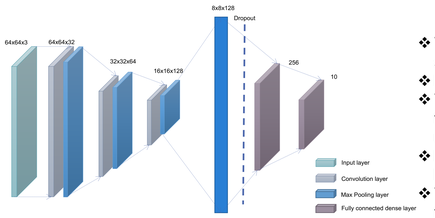
Image Resizing

- Reduces memory load by reducing the image size
- Allows the batch size to be increased to speed up model training time
- Keeps the original aspect ratio and preserves the feature information

IMPLEMENTATION



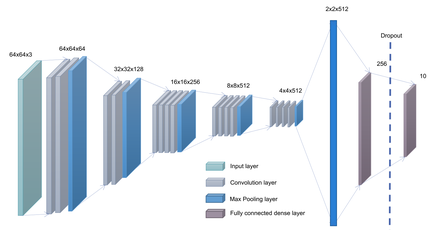
ARCHITECTURES



Convolution Neural Network from Scratch

- Three common layers are used, convolution, pooling, and fully connected
- Input layers accepts 64x64x3 input (RGB images)
- Three convolution layers, with various number of filters. E.g. 32, 64, and 128; All convolution layers use *relu* activation function and 5x5 stride.
- Each convolution layer has a following max pooling layer, which uses 2x2 pooling window.
- Two fully connected dense layers are following flattening out, the last layer has 10 as number of output, which corresponds to number of classes to classify.

- VGG network architecture includes two variants, VGG-16 and VGG-19.
- Original VGG-19 takes 224x224x3 input shape
- Includes 16 convolution layers(3x3 stride), 5 max pooling layers(2x2 pooling window), and 3 fully connected layers(first two layers have 4096 outputs, while the last has 1000 outputs)
- Replace three fully connected dense layers with two, which use 256 and 10 as output
- Keep convolution and pooling layers as convnet base, but later layers will be kept trainable.

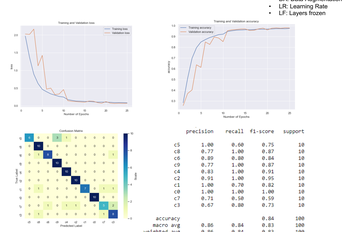


Fine-tuned VGG-19 Network

EVALUATION

Convolution Neural Network from Scratch

	Training		Validation		Test	
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
Base	0.022	99.33%	0.085	97.83%	2.69	66.00%
Base + D(0.3)	0.009	99.72%	0.022	99.46%	2.85	62.99%
Base + D(0.3) + BN	0.051	99.81%	0.041	99.62%	0.903	75.00%
Base + D(0.3) + BN + DA	0.207	94.98%	0.422	87.03%	0.664	81.00%
Base + D(0.2) + BN + DA	0.148	96.50%	0.221	94.11%	0.858	75.00%
Base + D(0.4) + BN + DA	0.343	91.52%	0.487	85.69%	0.762	74.00%
Base + D(0.3) + BN + DA + LR(0.01)	**0.087**	**97.55%**	**0.074**	**97.88%**	**0.683**	**84.00%**
Base + D(0.3) + BN + DA + LR(0.001)	0.199	95.12%	0.32	90.02%	0.804	75.00%
Base + D(0.3) + BN + DA + LR(0.0001)	0.59	88.65%	0.55	89.49%	0.96	70.00%
Base + BN + DA + LR(0.01)	0.0626	98.32%	0.078	97.63%	0.792	78.00%



- D: Dropout
- BN: Batch Normalization
- DA: Data Augmentation
- LR: Learning Rate
- LF: Layers frozen

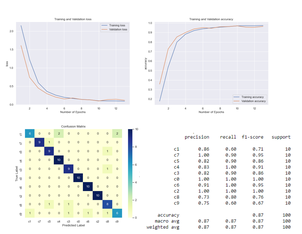
```
              precision    recall  f1-score   support

           c5       1.00      0.60      0.75        10
           c8       0.77      1.00      0.87        10
           c6       0.89      0.80      0.84        10
           c9       0.77      1.00      0.87        10
           c4       0.83      1.00      0.91        10
           c2       0.91      1.00      0.95        10
           c1       1.00      0.70      0.82        10
           c0       1.00      1.00      1.00        10
           c7       0.71      0.50      0.59        10
           c3       0.67      0.80      0.73        10

    accuracy                           0.84       100
   macro avg       0.86      0.84      0.83       100
weighted avg       0.86      0.84      0.83       100
```

Fine-tuned VGG-19 Network

	Training		Validation		Test	
	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy
Base	0.013	99.88%	0.083	97.46%	2.42	54.00%
Base + D(0.3)	0.092	97.11%	0.071	98.12%	2.11	46.99%
Base + D(0.3) + DA	1.23	56.95%	1.072	63.59%	1.536	46.00%
Base + D(0.2) + DA	1.186	58.52%	1.073	63.39%	1.735	39.00%
Base + D(0.4) + DA	1.378	51.39%	1.207	59.29%	1.546	48.00%
Base + D(0.3) + DA + LR(0.01)	1.807	33.25%	1.745	35.07%	1.678	30.00%
Base + D(0.3) + DA + LR(0.001)	1.303	53.69%	1.13	61.92%	1.66	44.00%
Base + D(0.3) + DA + LR(0.0001)	1.613	43.54%	1.514	47.92%	1.467	47.00%
Base + D(0.3) + DA + LR(0.0001) + LF(15)	0.123	96.03%	0.157	95.02%	0.635	82.00%
Base + D(0.3) + DA + LR(0.00001) + LF(15)	0.177	94.33%	0.207	93.06%	0.9843	70.00%
Base + D(0.3) + DA + LR(0.0001) + LF(10)	**0.092**	**97.32%**	**0.115**	**96.43%**	**0.623**	**87.00%**



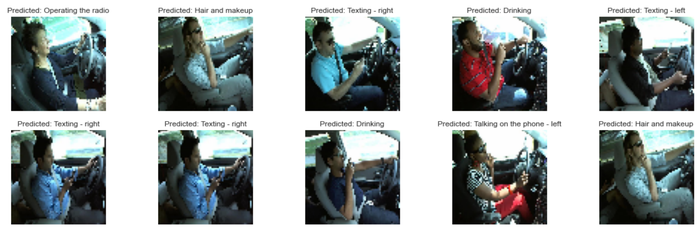

```
              precision    recall  f1-score   support

           c1       0.86      0.60      0.71        10
           c7       1.00      0.90      0.95        10
           c5       0.82      0.90      0.86        10
           c4       0.83      1.00      0.91        10
           c2       1.00      1.00      1.00        10
           c8       1.00      1.00      1.00        10
           c6       0.91      1.00      0.95        10
           c3       1.00      1.00      1.00        10
           c0       0.73      0.80      0.76        10
           c9       0.75      0.60      0.67        10

    accuracy                           0.87       100
   macro avg       0.87      0.87      0.87       100
weighted avg       0.87      0.87      0.87       100
```

CONCLUSION & LESSONS LEARNED

CONCLUSION: After evaluated the results for all trainings and performance of trained models, it is clearly that a model had been trained and is capable to detect distracted drivers from a dash camera image. Figure below shows the prediction result from a group of sample images. Therefore, we can conclude convolution neural network is an efficient approach for image classification problem. The results show fine-tuned VGG-19 network outperforms the convnet from scratch under the same maximum training epochs, batch size, and same input eventually. But this is not always true. Without re-train the convnet base, the scratch network always beats the fine-tuned and it takes less time to train, so scratch network seems to be a better choice from cost performance respective. Just like other neural networks, overfitting issue is a challenge for this project due to limited data, but the combination of image data augmentation, batch normalization and dropout minimized the impact of overfitting.



LESSONS LEARNED: Throughout this project, there are mistakes were made which leads to un-expect result. For example, *horizontal_flip* was set to True while using *ImageDataGenerator*, which misleads the neural network since there is separation among left and right for distraction classes. I also learned that fine-tuning a pre-trained model needs to be look at case by case, good result is not guaranteed without re-training the convnet base, even though it may work very well for object detection tasks. In this project, partial layers from the convnet base must be set for trainable, and experiments are needed for determining number of layers to be re-trained. Besides, I also learned the Dropout rate does not show linear impact on the performance. Among 0.2, 0.3, and 0.4, 0.3 shows the best result. Lastly, I learned that there is no best learning rate for all networks. For example, with the same number of training epochs, 0.01 works very well for the scratch version convnet, while 0.001 is a better choice for fine-tuned VGG-19 convnet.

FUTURE WORK

Based on the outcome of this project, the initial objectives have been met. However, there are still areas can be explored. With regarding to Image augmentation, this project only used basic image manipulations, which can be extended to use Deep learning approaches, such as adversarial training, neural style transfer, and GAN (Generative adversarial networks) Data augmentation [7]. Another area can be improved is the training platform, instead of relying on the CPU of personally MacBook Pro, it can use the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. Specifically, it used the Bridges-2 system, which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC). [5, 6] Further, trained model can be deployed to real environment, input images captured from cameras, for real-time detection, only as experimental study since the model was trained on a competition dataset.

ACKNOWLEDGEMENT

This project was inspired by driving education classes from Greg Driving School. I also would like to express sincere thanks to Dr. Liu, Xinlian for his great guidance given through the semester under the pandemic situation. In addition, I would like to give thanks to Omar Aboul-Enein for sharing his project with the class and providing tips that helps me to accomplish my project. Last not the least, I would like to express my deepest thanks my wife, my whole family and colleagues for their love, understanding, and support.

REFERENCE

[1] Centers Disease Control and Prevention (CDC) Transportation Safety. *Distracted Driving.* https://www.cdc.gov/transportationsafety/Distracted_Driving/index.html

[2] State Farm Distracted Driver Detection. https://www.kaggle.com/c/state-farm-distracted-driver-detection/data

[3] National Highway Traffic Safety Administration. (April 2020) Traffic Safety Facts Research Note: Distracted Driving 2018external icon. Department of Transportation, Washington, DC: NHTSA. Accessed 18 August 202

[4] Global status report on road safety. World Health Organization. https://www.who.int/violence_injury_prevention/road_safety_status/report/en

[5] Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G.D., Roskies, R., Scott, J.R. and Wilkens-Diehr, N. 2014. XSEDE: Accelerating Scientific Discovery. Computing in Science & Engineering. 16(5):62-74. http://doi.ieeecomputersociety.org/10.1109/MCSE.2014.80.

[6] Nystrom, N. A., Levine, M. J., Roskies, R. Z., and Scott, J. R. 2015. Bridges: A Uniquely Flexible HPC Resource for New Communities and Data Analytics. In Proceedings of the 2015 Annual Conference on Extreme Science and Engineering Discovery Environment (St. Louis, MO, July 26-30, 2015). XSEDE15. ACM, New York, NY, USA. http://dx.doi.org/10.1145/2792745.2792775.

[7] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1). https://doi.org/10.1186/s40537-019-0197-0