Deep Learning on Image Classification: Hybrid Model for Distracted Driver Detection

Lok Ping Joanna Wang, Mo Zhou

University of Colorado Boulder

Abstract. Numerous car accidents are attributed to driver distractions such as texting every year. This study endeavors to enhance the accuracy of image classification in distracted driver detection using different model architectures, aiming to alert distracted drivers and mitigate accidents. Prior research has utilized pre-trained CNNs and it demonstrates satisfactory outcomes. Transformers have shown proficiency in image classification tasks, with recent studies incorporating hybrid models (ResNet and Vit) surpassing VGG-19. This paper will leverage pre-trained CNN models as the baseline and explore whether alternative hybrid models of pre-trained CNN and transformers can have superior results. The findings indicate that pre-trained CNNs perform the best, followed by a hybrid model with ResNet and Vit. This showcases the effectiveness of CNN in capturing local features when classifying images. Another finding is that ResNet50 and ResNet50+ViT models exhibit confusion between the label safe driving and the label talking to passengers. Future research could modify the transition layer of the hybrid model and involve augmenting data samples for classes 0 and 9 to enhance accuracy

Keywords: Vision Transformer; Swin Transformer; Hybrid Model; Distracted Driver Detection

1 Introduction

According to the National Highway Traffic Safety Administration (2024), 3,308 people were killed and 289,310 people injured in motor vehicle crashes because of distracted drivers in 2022. The estimated economic cost of distracted-driving traffic crashes is $98 billion, including emergency medical services and property damage. Distractions are activities that make the driver not focus on driving, for example talking to passengers and texting the phone. By detecting and alerting drivers when they are distracted, technology can help prevent accidents, saving lives and reducing injuries. Therefore, it is important to eliminate risky behaviors on the road to mitigate casualties and economic loss.

Distracted driver detection can make use of image classification to identify the distraction. Convolutional Neural Networks (CNNs) perform well in image classification, which can be used for distracted driver detection [1]. Previous research has used pre-trained models on deep convolutional neural networks (CNNs) for distracted driver detection. For instance, Faster-RCNN [2], VGG-19

[3], AlexNet, VGG-16 and ResNet152 [2]. By transfer learning and fine-tuning the pre-trained model on our dataset, we can make use of knowledge learned or features extracted from a dataset and minimize computational resources. Recently, transformers are another state-of-art of neural network architecture that has gained significant popularity and success in computer vision. The attention mechanism can capture long-range dependencies and contextual information more effectively compared to CNN. Vision Transformer (ViT) and Swin Transformers are two popular transformers for computer vision, but using only transformers does not outperform CNNs [4]. Li et al.[5] proposed to combine ViT and CNN on distracted driving detection as it can capture both local and global contexts. ViT requires large computational resources and the Swin Transformer is a variation of the original Vision Transformer (ViT), so Koay et al.[4] have studied the potential of the Swin Transformer for distracted driving detection. This project is going to fill the gap by exploring the performance of combining Swin Transformer and CNN on distracted driving detection and to compare the performance of different models with slightly different architectures.

This project proposes to combine CNN with different transformer architectures by adding a transformer layer on top of a CNN pre-trained model serving as the feature extractor. The project will compare the novel architecture to previous solutions, such as comparing hybrid models composed of transformers and CNN to pre-trained CNN models including ResNet and VGGNet, and transformers including ViT and Swin, as the baseline.

2 Related Work

CNNs have been proven to perform well on image classification tasks, particularly on the topic of distracted driver detection. Yan et al.[1] used CNN with sparse filtering on the Southeast University Driving-posture Dataset (SEU dataset) that was built in 2012 and performed an accuracy of 99.47%. However, the dataset focusing on hand position only, includes postures of correct driving position with hands on wheel, operating the shift gear, eating or smoking, and responding to a cell phone. Splitting the similar action “eat” and “smoke” could better suit the need to classify dangerous behaviors. Hoang et al.[2] proposed the Multiple Scale Faster-RCNN (MSFRCNN) approach on another dataset from the Strategic Highway Research Program (SHRP-2) and the accuracy is 94.2%. However, the model focuses on the phone usage detection based on face and hands segmentation, so other distracting behaviors like talking to other passengers are not included. In contrast, our project would use a dataset that includes more distracting behaviors.

The StateFarm dataset(SFDDD), that we selected in this project, is more comprehensive as it can recognize more distracting behaviors including using a cell phone for texting or calling, drinking, operating car accessories, talking to passengers, reaching behind, and makeup. The action of texting and talking on the phone are even further split based on using right or left hands, so there are totally 10 classes within this dataset. Hssayeni et al.[6] used AlexNet, VGG-16,

and ResNet-152, which are pre-trained on ImageNet, and fine-tuned the last two FC layers on this dataset(SFDDD). The work showed ResNet-152 performs the best and the highest accuracy on the test data is 85%. It also suggests that fine-tuned models are still under the risk of overfitting. Koesdwiady et al.[3] did fine-tuning on pre-trained VGG-19 to fit on this dataset and reached an overall accuracy of 95% and an average accuracy of 80% per class on the test set. The above work demonstrated that pre-trained CNN models have a satisfying result on distracted driver detection and we would use them as the baseline.

Transformers have also been proven to perform well on image classification tasks, after their success in natural language processing tasks[7]. Koay et al.[4] trained and compared a total of 17 CNN models and 16 Vision Transformers on another detecting driving distraction dataset(AUC-DDD dataset). The paper shows the result that ResNet and VGG-19 perform the best among 17 CNN models, and Vit and Swin transformer perform the best among 16 Vision Transformers. It also shows pre-trained CNN models still outperform Vision Transformers, especially when dealing with a low amount of data. Li et al.[5] proposed a ViT-Conv module that combines CNN with ViT and has shown it outperformed VGG-19[3] on the SFDDD by capturing both the local and global features. It demonstrates the reason and potential of combining CNN and transformers. Therefore, this paper will dive more deeply into studying if combining other transformers like Swin Transformer with pre-trained CNN models could produce a better result.

3 Methods

3.1 Dataset

The dataset we used is State Farm Distracted Driver Detection from Kaggle. It includes driver images taken in a car with a driver doing something in the car (texting, eating, talking on the phone, makeup, reaching behind, etc). The goal is to predict the likelihood of what the driver is doing in each picture. The classes include:

TABLE.1. DRIVER ACTIONS AND TOTAL IMAGES IN THE CLASS.

Classes	Driver actions	Images
C0	safe driving	2489
C1	texting - right	2267
C2	talking on the phone - right	2317
C3	texting - left	2346
C4	talking on the phone - left	2326
C5	operating the radio	2312
C6	drinking	2325
C7	reaching behind	2002
C8	hair and makeup	1911
C9	talking to passenger	2129
Total		22424

Fig. 1. Dataset Samples for Each Class

3.2 Model

In this experiment, we test the performance of different models on this image classification task, including CNN-based models and transformers. We also combine both CNN and transformers together to see if it can result in a better performance. The reason for the combination is because CNN models capture local patterns of an image better, and are efficient at processing raw pixel data and extracting low-level features like edges, and textures, while transformers utilize self-attention mechanisms to capture global dependencies within the image. The CNN-based models we used are ResNet50 and VGG16. The transformer architectures we used are ViT and Swin.

For CNN-based models, ResNet-50 consists of a series of convolutional layers followed by residual blocks, also named skip connections. This allows the network to bypass one or more layers, facilitating the training of very deep networks. VGG-16 (Visual Geometry Group) consists of a series of convolutional layers, each followed by a max-pooling layer, and finally several fully connected layers.

For transformers, Vision Transformer (ViT) divides the image into fixed-size patches, which are then flattened into sequences and fed into the transformer encoder. Swin Transformer divides the input image into non-overlapping patches and applies self-attention within these windows, which are then shifted in subsequent layers for broader receptive fields without dramatically increasing computational cost. After the shifting mechanism where local windows are repeatedly shifted and aggregated, the embeddings are then fed into the transformer encoder. Swin_t indicates swin_tiny architecture, which is more lightweight and can reduce computational cost.

For better performance in limited time and resources, we used pre-trained weights on ImageNet dataset. The input size is 224x224. The output contains one thousand classes. In order to fit on our task, we changed the final fully-connected layer to predict ten classes.

3.3 Framework

We first tried to use TensorFlow as the framework, since it is convenient to make changes on provided models and combine different layers. Along with the project going, the training process gradually takes longer and it requires more computation resources, which makes it necessary to do training with GPUs. When setting up a GPU-based training environment, we found it extremely hard due to its compatibility with CUDA on the Windows system and other necessary libraries as well. Therefore, we ended up choosing Pytorch as the deep learning framework.

3.4 Implementation

Data Preprocessing The size of our images is different from ones in ImageNet, so to fully exploit pre-trained weights. We first reshaped the input images and then did normalization according to values from ImageNet. Doing training on all the samples will be costly, so we evenly extracted 1500 samples from every class and separated them into training(0.5), validation(0.2), and testing(0.3) sets. The data loading time is about 108 seconds.

Training and Hyperparameters The loss function we used for each model is cross-entropy, which is a good choice for multi-label classification. The optimizer we used includes SGD with momentum and Adam. For faster training, SGD allows us to use a bigger batch size and add more epochs, and introducing momentum is more likely to find the global optimal. However, it is still hard to converge after training, and the performance always falls behind using Adam. After trading off, we ended up using Adam with a batch size of 16. Since Adam can adjust the learning rate according to the current gradient, the initial learning rate is not as important as in SGD. To keep the ability learned from ImageNet, we set the number of epochs to be lower than 5, and the experiment turns out that the model performs best when it is set to be 3.

Table 2. Hyperparameter of all model

learning rate	0.0001
batch size	16
number of epochs	3
image resize	(224, 224)
test ratio	0.3
val ratio	0.2

Hybrid Models To prepare features for compatibility with transformers, adjustments are needed to ensure the CNN output can be organized into patches.

This involves removing the final fully connected layers and utilizing the feature maps as input images for transformers. The transition preceding the Swin transformer consists of a CNN layer, followed by the merging of multiple input feature map channels into three channels, a batch normalization layer to enhance training stability and speed, and a ReLU layer to maintain the gradient of data.

The transition preceding the Vision transformer differs slightly. Here, the ViT takes the feature map size (7x7x2048) as the input size and segments 1x1 patches. This approach allows ViT to treat each pixel of the feature map as an embedding, corresponding to a local portion of the raw image, and then compute global relationships among all pixels. Differences in architectures result in different performance outcomes, which will be explained further in subsequent sections.

4 Experimental Design

4.1 Experiment

The purpose of our project is to figure out the potential of transformers and hybrid models in computer vision. For specific task image classification here, we want to know which transformer works best, if it is better than CNN-based models, and if utilizing CNN as a feature extractor will increase its accuracy. Therefore, the experiment is composed of the following steps:

1. Do fine-tuning and training on ResNet50 and VGG16; Compare their performance on this task and choose the better one as the feature extractor in step 3.

2. Train Vision transformer and Swin transformer on this dataset; Evaluate their performances;

3. Construct hybrid models with different transformers and CNN-based model chosen on step 1; Explore and analyze the possible improvement and deterioration.

4.2 Evaluation

Each image will be labeled with one true class and calculated a set of predicted possibilities.The evaluation metric will be simply classification accuracy and also follow the one given by the dataset which is multi-class logarithmic loss:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} log(p_{ij}) \tag{1}$$

where N is the number of images in the test set, M is the number of image class labels, log is the natural logarithm, y_{ij} is 1 if observation i belongs to class j and 0 otherwise, and p_{ij} is the predicted probability that observation i belongs to class j. Logarithmic loss, however, penalizes incorrect predictions based on

their probability scores, providing a more nuanced view of model performance across all classes.

The reason of using this is because it penalizes incorrect predictions that are made with high confidence more heavily. This will not only capture whether the predictions are right, but introduce the influence of confidence. Besides this, accuracy can also be misleading when predicting imbalanced data, as the model can predict the majority class most of the time and still achieve high accuracy.

Moreover, we also utilized confusion matrix to assess the performance of image classification. It provides a detailed breakdown of the model's predictions compared to the actual labels across different classes. It helps to understand how well the model distinguishes among different classes of images, and also enables us to identify specific classes that are challenging for the model so that it can provide insights for potential enhancements.

4.3 Results and Findings

Table 3. Metrics of Models

Model	Train Acc	Test Acc	Logloss	Size	Training Time/s
ResNet50	99.33	98.89	0.040	23528522	335.87
VGG16	98.08	98.23	0.077	134301514	1003.80
ViT	97.84	95.97	0.137	85806346	909.81
Swin	98.04	97.45	0.100	27527044	468.07
ResNet+ViT	98.53	98.03	0.071	50063946	392.00
ResNet+Swin	94.99	95.97	0.183	51810229	376.86

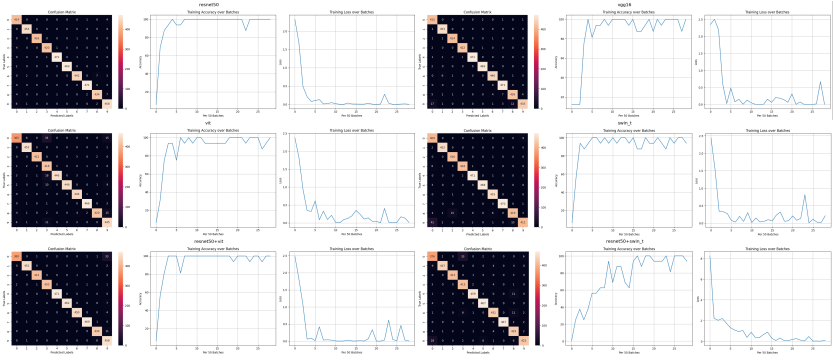

Fig. 2. Confusion Matrix (Left) and Learning Curve (Middle and Right). It shows the confusion matrix, training accuracy and loss over batches for six models.

From Table 3, ResNet50 and VGG16 performed the best, followed by ResNet50+vit. So transformers and hybrid models do not always perform better than pre-trained CNN. This suggests CNN models are good at capturing local features, which makes them more effective for classifying important visual cues such as the position of the hands on the steering wheel. Transformers, on the other hand, are particularly effective at capturing long-range dependencies and global context across the entire image because of the self-attention mechanism.

Table 3 also shows that ResNet50 with Vit performs better than ResNet 50 with Swin_t. As mentioned in the method section, the transition between ResNet and Swin transformer is to simply merge multiple channels to three corresponding to RGB input requirements of Swin transformer, while the transition between ResNet and Vision transformer is to treat each feature map as an input image and set the input patch size for vision transformer to be 1x1. This difference suggests the way of combining CNN-based models and transformers in the transition layer will significantly influence the performance. The feature extracted by CNN models will be reserved in each pixel of the final feature map. To utilize that in transformer architectures, we need to do pixel-based operations. Future work on the transition layer of Swin_t should be done for better comparison. Another reason could be that Swin_tiny implies the model has fewer parameters because of fewer layers and smaller hidden dimensions, causing the combination of Swin_t did not perform better.

Considering ResNet50, which has the best performance among the pre-trained model, and ResNet50+vit, which is the better hybrid model, their confusion matrix shows that both ResNet50 and ResNet50+vit are confused with class 0 (safe driving) and class 9 (talking to passengers). This may be due to the training dataset that classes 0 and 9 have a very similar hand position. Future work should also include more data samples for classes 0 and 9 to improve the accuracy.

4.4 Limitation and Future Work

Due to the limited time and shifted-window mechanism, the transition between ResNet and Swin transformer is not efficient and sensible enough. To fully compare the performance of hybrid models and explore their potential in this task, the transition action should be reasonable and similar on all the transformer architectures. Moreover, our model fails on pictures taken on our friends. This indicates a bad generalizability among conditions in real life. This could be because different size of pictures or the overall hue. Also, as discussed in the above section, the dataset should include more data samples for classes 0 and 9 to improve the accuracy of identifying the 2 classes.

Therefore, future work would mainly focus on constructing a generalized and reasonable transition layer for all kinds of hybrid models, and gathering more training samples taken under various conditions. It would also be feasible to utilize adversarial learning to improve the robustness.

5 Conclusion

In conclusion, pre-train CNN performs the best in classifying distracted drivers because it can capture local features like the hand position on a steel wheel. ResNet50+vit ranked third and this suggests the hybrid model still has the potential in the application of distracted driver classification. Future work should be done on combining ResNet and Swin with the transition that treats each feature map as an input image of the transformer for a fair comparison with other hybrid models. Also, since both ResNet50 and ResNet50+vit are confused with class 0 (safe driving) and class 9 (talking to passengers), future work should include more data samples of these 2 classes for training.

Finally, one concern of this task is that the majority of individuals are hesitant to install cameras for recording and detecting their driving behavior due to privacy issues. Besides this, a set of related policies are also required to work with this system, so that it can use the information and reduce the car accidents. However, it could be a durable problem, since the punishment is hard to made according to this machine learning method. It might not be appropriate to punish people due to what does not happen yet, and the model could also make mistakes.

References

1. Yan, C., Coenen, F., Zhang, B.: Driving posture recognition by convolutional neural networks. IET Computer Vision **10**(2) (2016) 103–114
2. Hoang Ngan Le, T., Zheng, Y., Zhu, C., Luu, K., Savvides, M.: Multiple scale faster-rcnn approach to driver's cell-phone usage and hands on steering wheel detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. (2016) 46–53
3. Koesdwiady, A., Bedawi, S.M., Ou, C., Karray, F.: End-to-end deep learning for driver distraction recognition. In: Image Analysis and Recognition: 14th International Conference, ICIAR 2017, Montreal, QC, Canada, July 5–7, 2017, Proceedings 14, Springer (2017) 11–18
4. Koay, H.V., Chuah, J.H., Chow, C.O.: Convolutional neural network or vision transformer? benchmarking various machine learning models for distracted driver detection. In: TENCON 2021-2021 IEEE Region 10 Conference (TENCON), IEEE (2021) 417–422
5. Li, Y., Wang, L., Mi, W., Xu, H., Hu, J., Li, H.: Distracted driving detection by combining vit and cnn. In: 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), IEEE (2022) 908–913
6. Hssayeni, M.D., Saxena, S., Ptucha, R., Savakis, A.: Distracted driver detection: Deep learning vs handcrafted features. Electronic Imaging **29** (2017) 20–26
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)