A lightweight ConvGRU network for Distracted Driving detection

Pantazis Anagnostou, Nikolaos Mitianoudis
Electrical & Computer Engineering Dep.
Democritus University of Thrace
Xanthi, Greece
{pantanag, nmitiano}@ee.duth.gr

***Abstract*—In this paper, we explore the problem of automatic detection of dangerous or distracted driving using multi-modal cameras. A deep convolutional model with Gated Recurrent Unit (GRU) layers for classification on the Driver Anomaly Detection (DAD) dataset is proposed. The key features are the limited use of 3D convolutions and the replacement of 2D convolutions with depth-wise separable convolutions, which reduce the computational complexity of the model to a small fraction of previous architectures with a small decrease in AUC performance. In addition, the threshold for binary classification between safe and distracted driving is adaptively estimated through the training data. Finally, an ensemble of all multi-modal inputs yields the final classification with favourable performance.**

***Index Terms*—Dangerous Driving, Anomaly Detection**

I. Introduction

Driving is one of the most common activities for humans in their everyday life. The quality of human driving depends on many independent parameters, thus driving can be characterised as a complex action that may unfortunately lead to unwanted consequences. On top of that, dangerous or distracted driving is one of the main reasons behind those accidents, since it leads to four times greater possibility of an accident [2]. To tackle this problem, deep learning provides neural models that can predict abnormal or distracted driving behavior and inform the driver to take the necessary action.

A. Driver Monitoring Datasets

Many datasets for distracted driving identification have been proposed, each representing different aspects of such behavior. The State Farm Distracted Driver challenge [3] offered one of the first of datasets, which contains samples from safe, or normal driving and nine other classes of distracted driving. Similarly, the Distracted Driver Dataset, provided by the American University of Cairo [4], contains videos from 44 drivers and 7 countries. The dataset is labeled into 10 different classes, similarly to the previous dataset. Both these datasets are limited to one camera view and modality (visual). To provide more information to the system, datasets that combine multiple camera views and modalities have emerged. One of these datasets is the Driver Monitoring Dataset (DMD) [5], which provides footage, captured by 3 separate cameras, each having 3 different modalities: RGB, infrared and depth. Similarly to the DMD, the Driver Anomaly Detection (DAD) dataset [1] features clips, captured from two separate cameras, each having two modalities (infrared and depth).

B. Previous approaches to Dangerous Driving Detection

The problem of distracted driving can be described as an action recognition problem. Deep Learning networks have been developed to work on raw data, thus performing feature extraction and classification simultaneously through novel architectures. A number of image classification deep networks have been proposed, including the ResNet [18], ResNeXt [19] and the computationally light MobileNet [20]. These approaches are trained mainly through supervised learning, although some of the latest proposals also involve the use of contrastive learning [1], [16]. In [1], a 3D ResNet18 architecture along with a fully connected classification layer were proposed. In [16], Khan et al. used the same architecture as in [1], but proposed a new loss function and a new projection head during the test phase of contrastive learning. In [17], Tüfekci et al. used a ResNet34 for feature extraction, which were then presented to a LSTM autoencoder, that was trained by minimising the MSE between the ResNet34 features and the output of the autoencoder.

Since this system needs to be placed in a car and detect dangerous driving in real-time, it is essential that the deep learning architecture should be minimal. The main reason is that the car's processing system would probably have more important tasks to fulfill. In this scenario, lightweight architectures should be used for driver distraction detection, without sacrificing much of the detection accuracy, due to its safety importance.

This paper presents a lightweight architecture for dangerous driving detection on the Driver Anomaly Detection (DAD) using supervised learning. Emphasis is given on developing a simple, yet effective, architecture with very low computational cost, in order to ensure real-time implementation.

II. Proposed Methodology

A. Proposed Architecture

This paper aims at introducing a lightweight architecture that manages to maintain a high quality performance. To achieve this, a combination of 3D and 2D convolutions is used to replace the standard 3D convolutions, commonly used in previous approaches [1], [16]. In addition, some of the 2D convolutions are replaced with the much less computationally expensive, depth-wise separable convolutions [10]. While these alterations significantly decrease the size of the architecture, it is crucial that the performance does not decrease as well. To accomplish this, different methods of normalization are used, i.e. GroupNorm [12] and Batch Normalisation, along with a modified 2D convolution that employs the weight standardization [11]. The CNN part of the neural model is responsible for feature extraction. Consequently, a two-layer Gate Recurrent Unit (GRU) captures the temporal correlations behind the image sequence that is presented to the network. Finally, a fully connected network operates as the binary classifier, which will produce the final answer for each different modality/input. The complete architecture is depicted in Fig. 1 and Table I in more detail.

B. Decision Making

The model produces a final score, or logit, which will be compared with the corresponding threshold to infer the existence of distracted driving or not. This threshold can be considered a probability threshold between the two actions, safe and distracted driving. If the score is greater than the threshold then the action is classified as distracted, while in the opposite case, the action is classified as safe. Common values for such thresholds are 0.5, assuming equal probability of detecting a distracted or a safe driving scene. Since this can vary in many applications, in this work, this threshold is determined by the optimal detection threshold between the two cases in the training dataset. More specifically, a Gaussian distribution $\mathcal{N}_1(\mu_1, \sigma_1)$ is fitted to the values, produced by the safe driving examples, and a second Gaussian distribution $\mathcal{N}_2(\mu_2, \sigma_2)$ on the values, produced by the distracted driving examples. The optimal threshold is determined during training at the intersection of the two Gaussians $\mathcal{N}_1$ and $\mathcal{N}_2$, which is estimated numerically. Fig. 2 demonstrates the optimal threshold selection for a specific run, noting that the network's output is presented in logits.

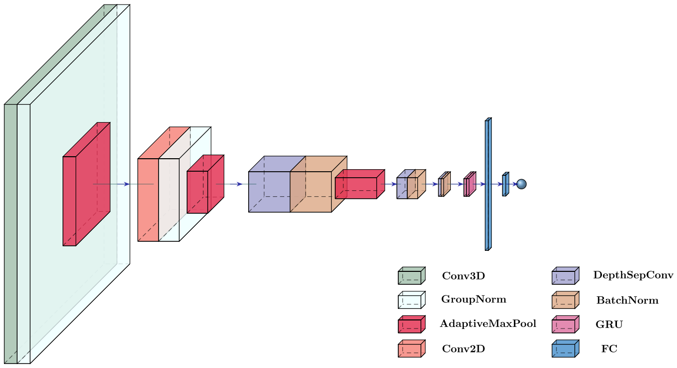

Fig. 1. The proposed architecture, consisting of 3D convolutional layers (Conv3D), Group Normalisation (GroupNorm), Adaptive Max Pooling (AdaptiveMaxPool), 2D convolutional layers (Conv2D), Depth Separation Convolutional Layers (DepthSepConv), Batch Normalization (BatchNorm), Gated Recurrent Unit (GRU) and Fully-Connected (FC) layers.

TABLE I
THE PROPOSED ARCHITECTURE CONVGRU.

Layer/Stride	**Contents**	**Output Size (CxDxHxW)**
Input Clip	-	1x16x160x160
Conv3D	$Conv3D(16, k = 3)$ $Dropout(p = 0.2)$ $LeakyReLU$ $GroupNorm(n = 8)$ $AdaptiveMaxPool3D$	16x8x55x55
Block 1	$Conv2D(128, k = 5)$ $Dropout(p = 0.2)$ $LeakyReLU$ $GroupNorm(n = 8)$ $AdaptiveMaxPool2D$	128x8x26x26
Block 2	$DwSepConv(256, p = 1)$ $Dropout(p = 0.25)$ $LeakyReLU$ $BatchNorm$ $AdaptiveMaxPool2D$	256x8x13x13
Block 3	$DwSepConv(64)$ $Dropout(p = 0.2)$ $LeakyReLU$ $BatchNorm$	64x8x13x13
Block 4	$DwSepConv(16)$ $Dropout(p = 0.25)$ $LeakyReLU$ $BatchNorm$	16x8x11x11
Recurrent Block	GRU(n_layers=2, hidden_size=100)	100x8
FC1	-	800
FC2	-	128
FC3	-	1

C. Ensemble Methods

Ensemble is the technique used to combine the inference of different models or the same model on different modality inputs. In the case that the driver dataset contains multiple camera views or multi-modal inputs, one solution is to apply the proposed architecture to each view of modality independently and consequently use an ensemble method to fuse the

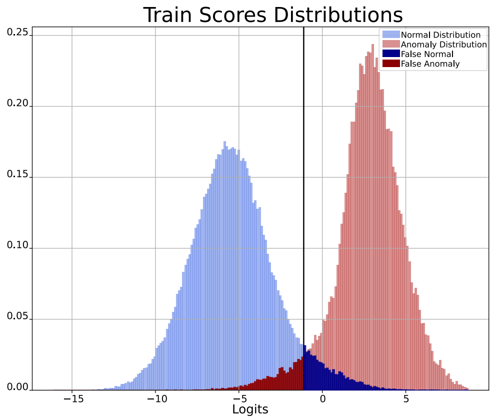

Fig. 2. Example of threshold obtained from intersection point.

inferences. Ensemble can be applied through different methods, but in this paper we focus only on Majority Voting and Decision based on Average. Majority voting reaches a decision through voting and more specifically, by choosing the decision with the greatest number of votes. In the case of a tie, the final decision will be made by the neural model of the mode with the highest confidence during training. On the other hand, the second method uses the average scores or probabilities produced by each model. The estimated thresholds for each modality/view are averaged and the average is used.

III. Dataset and Implementation

A. Dataset Details

The dataset that was used in this series of experiments is the Driver Anomaly Detection (DAD) dataset [1]. The DAD provides an abundance of data, recorded by cameras placed in two different positions, one top and one front, with each camera having two different modalities: infrared and depth. This dataset can provide more information to the neural model, since it examines and analyzes the problem from different aspects. In addition, DAD is split into a training and a testing set, which contain 650 and 133 minutes of data respectively. It should be noted that the cameras record at 45 frames per second, leading to approximately 120GB of data. The training set consists of recordings of normal driving and 8 scenarios of distracted driving, while the testing set contains additional scenarios of distracted driving.

B. Pre-processing

Since the data are highly imbalanced, some pre-processing is required. Firstly, a center crop is applied to remove redundant information, such as black pixels. Each frame is finally resized to 160×160. Secondly, data augmentation is used to tackle the imbalance in the input data. To tackle the problem, data associated with distracted driving are augmented with a higher probability than those from safe driving. From all possible video augmentations, the following 3 spatial and 1 temporal transformations are selected:

- Horizontal Flip: This flips every frame of the input video sequence. This augmentation is rational, since driving may take place on the other side of the car in some countries.
- Gaussian Blur: This applies a Gaussian blur filter to every frame of the input video sequence. In addition, cameras may get out of focus occasionally, which can be simulated by Gaussian blur.
- Salt and Pepper Noise: This augmentation adds salt and pepper noise to the original input video sequence. Similarly to Gaussian blur, it introduces some variety to the data, but noise may be introduced by the sensor or other technical difficulties.
- Temporal Mirroring: this is a temporal transformation that reverses the order of frames in the input video sequence.

All these transformations introduce extra variety to the data, so as to improve the generalization qualities of the network.

C. Class Imbalance

Data augmentation is a useful method to deal with the problem of class imbalance, but only to a certain extent. Many techniques [7], [9] have been developed to tackle this problem. In this experiment, a weighted loss function was used to take into account the distribution of data. The weighted loss function that was chosen is the Weighted Binary Cross Entropy [8], where the weight is determined by the variable pos_weight:

$$pos\_weight = \frac{N_{neg}}{N_{pos}} \tag{1}$$

In (1), N_{neg} refers to the number of samples of the negative distribution (safe driving) while N_{pos} corresponds to the positive distribution (distracted driving). Distracted driving is characterized as the positive action, since it is the action that needs to be detected, hence its importance is higher.

D. Training Details

The proposed model is randomly initialised and trained for 50 epochs using early stopping. Two different optimizers were employed: Stochastic Gradient Descent (SGD) for the frontal view videos and AdaBound [14] for the top view videos. The parameters used in the two optimizers are listed below:

- Stochastic Gradient Descent (SGD): Learning rate is set to 5e-5, weight decay [13] is 1e-6, momentum is 0.9 and nesterov momentum is set to true.
- AdaBound: Learning rate is set to 1e-5 while the final learning rate it uses is set to 5e-5. Finally, weight decay is also 1e-6 while amsbound is set to true.

A batch size of 32 sequences was used, with each input video sequence consisting of 16 frames ($sample\_duration$). The network was implemented in PyTorch[1] on a system with a AMD Ryzen 5 5600X, 16GB DDR4 RAM clocked at 3200MHz and an RTX 3060 with 12GB RAM.

IV. Results

Since the data are highly imbalanced, the use of accuracy as an evaluation metric is not preferred. Therefore, F1-Score and Area Under Curve (AUC) are selected for evaluating the performance of the model. By definition, AUC calculates the area under the Receiver Operating Curve (ROC) curve, which determines the performance of the model through different thresholds. Table II summarises the experimental results for the proposed architecture ConvGRU. While individually each camera does not provide much confidence, the two ensemble methods increase the total performance, especially the AUC. Note that AUC was not available for the majority voting method, since logits or probabilities were unavailable. The best performances are observed using the method Decision based on Average, reaching an F1 Score of 0.7205 and an AUC of 0.886.

The proposed architecture is also compared to Köpüklü et al. [1], Khan et al. [16] and Tüfekci et al. [17] that use

[1] The developed code can be found at https://github.com/pantanag/ConvGRU-Driver-Distraction-Detection.

TABLE II

RESULTS OF THE PROPOSED CONVGRU ARCHITECTURE FOR EACH INDEPENDENT SENSOR AND THE RESULTS USING THE ENSEMBLE MODES.

ConvGRU	front_IR	front_depth	top_IR	top_depth	Majority Voting	Decision On Average
F1-Score	0.6935	0.606	0.6618	0.6647	0.7194	**0.7205**
AUC	0.8482	0.74	0.8191	0.8029	-	**0.8866**
Threshold[a]	-0.9005	-0.7038	-1.3789	-1.0242	-	-1.002975

[a] Threshold values are presented in terms of logits. To convert to probabilities, the sigmoid function must be applied.

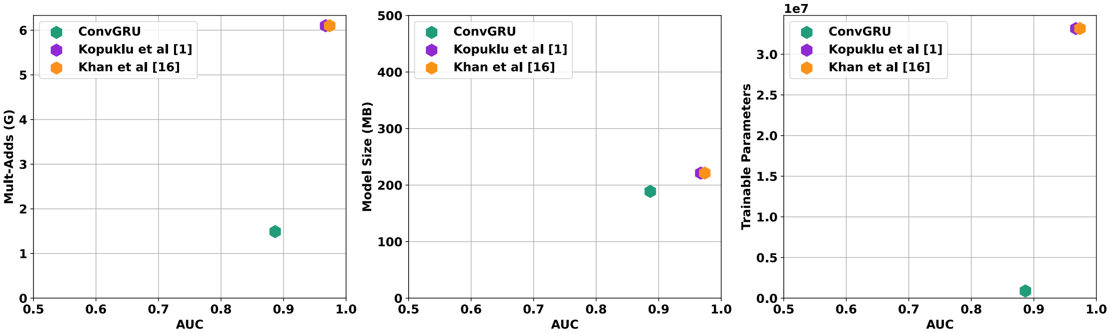

Fig. 3. Comparison of the proposed ConvGRU with two different approaches in terms of complexity, as determined by Mult-Adds, Model Size and Trainable Parameters.

TABLE III

COMPARISON OF APPROACHES IN TERMS OF COMPLEXITY AND PERFORMANCE.

Cost Variable	ConvGRU	Köpüklü et al [1]	Khan et al [16]	Tüfekci et al [17]
Mult-Adds[a]	**1.49**	6.1	6.1	41.39
Model Size[b]	**188.77**	221.20	221.20	699.06
Trainable Parameters[c]	**0.882**	33.16	33.16	63.67
AUC	0.8866	0.9673	**0.9738**	0.8434
ATPB[d]	**0.101536**	0.44307	0.449797	-

[a] Number of multiplications and additions measured in Giga.
[b] Total model size measured in MB.
[c] Number of trainable parameters measured in Mega.
[d] Average time per batch in inference, measured in seconds

the same dataset. The results are depicted in Table III. The proposed architecture may not yield the best performance, but it features other qualities, such as its lightweight nature, compared to other deeper architectures [1], [16], [17]. The works in [1], [16] manage to achieve better performance but with extra computational cost. To be more specific, the 3D version of ResNet18 is used as a based encoder in both cases leading to higher values in model size, number of trainable parameters and number of multiplications/adds[2]. Additionally, both these approaches use $sample\_duration = 16$, but in contrast to ConvGRU, they use dimensions of 112 × 112 for the frames, in order to reduce their model size. Compared to the 3D ResNet18, the proposed architecture has significantly less trainable parameters: 880K instead of 33M, i.e. 2.65% of the original parameters. This applies to the number of multiplications/adds, where ConvGRU has 24.4% of the original operations. Finally, the total model size of ConvGRU is also smaller than that of 3D ResNet18, with 188.77 instead of 221.20 for the other two, i.e. 85% of the original size. The reason behind this is the difference in the input frame dimensions, since ConvGRU uses larger frames (160 × 160) to provide more spatial information to the model. The first two factors are crucial in the model training or possible refinement/retraining stage. Of course, they also determine the response time in a real-time scenario. Specifically, multiplications/adds are performed during both training and testing, while trainable parameters also present an overhead in testing. On the other hand, the requirement of a small size model is equally important, in the case of real-time scenarios. Finally, the average inference time per batch (ATPB) was estimated for the three top methods. The proposed ConvGRU demonstrates its much smaller inference time compared to the higher performing methods.

V. CONCLUSIONS

In this paper, a lightweight ConvGRU architecture, which has not been applied before in this research field, is presented to detect distracted driving. The framework has replaced a 3D ResNet18 with a smaller network with fewer 3D convolutional units and 2D convolutional units using dilated convolution. The proposed architecture may not produce the best performance, offering an AUC of 0.8866, but it has considerable advantages for real-time applications, since it features only 2.65% of the original trainable parameters, 24.4% of the original multiplications and additions and 22.7% of the original inference time. These advantages promote the proposed ConvGRU as a preferable solution for real-time applications.

[2] These metrics were calculated with the help of the torchinfo library.

References

[1] Köpüklü, O., Zheng, J., Xu, H., Rigoll, G. Driver anomaly detection: A dataset and contrastive learning approach. *Proceedings Of The IEEE/CVF Winter Conference On Applications Of Computer Vision*. pp. 91-100 (2021)

[2] Dingus, T.A., Guo, F., Lee, S., Antin, J.F., Perez, M., Buchanan-King, M., Hankey, J. Driver crash risk factors and prevalence evaluation using naturalistic driving data. *Proceedings of the National Academy of Sciences*, 113(10):2636–2641, 2016.

[3] State Farm. State farm distracted driver detection. https://www.kaggle.com/c/state-farm-distracted-driverdetection

[4] Eraqi, H., Abouelnaga, Y., Saad, M., Moustafa, M. "Driver Distraction Identification with an Ensemble of Convolutional Neural Networks", Journal of Advanced Transportation, Machine Learning in Transportation (MLT) Issue, 2019.

[5] Ortega, J., Kose, N., Cañas, P., Chao, M., Unnervik, A., Nieto, M., Otaegui, O., Salgado, L. DMD: A Large-Scale Multi-modal Driver Monitoring Dataset for Attention and Alertness Analysis. *Computer Vision – ECCV 2020 Workshops*. pp. 387-405 (2020)

[6] Köpüklü, O., Kose, N., Gunduz, A., Rigoll, G. Resource efficient 3d convolutional neural networks. *2019 IEEE/CVF International Conference On Computer Vision Workshop (ICCVW)*. pp. 1910-1919 (2019)

[7] Buda, M., Maki, A., Mazurowski, M. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*. **106** pp. 249-259 (2018), https://doi.org/10.1016

[8] Aurelio, Y., De Almeida, G., Castro, C., Braga, A. Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function. *Neural Processing Letters*. **50** (2019,10)

[9] Yu, L., Zhou, N. Survey of Imbalanced Data Methodologies. (arXiv,2021), https://arxiv.org/abs/2104.02240

[10] Chollet, F. Xception: Deep Learning With Depthwise Separable Convolutions. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition (CVPR)*. (2017,7)

[11] Qiao, S., Wang, H., Liu, C., Shen, W., Yuille, A. Micro-Batch Training with Batch-Channel Normalization and Weight Standardization. (arXiv, 2019), https://arxiv.org/abs/1903.10520

[12] Wu, Y., He, K. Group Normalization. (arXiv, 2018), https://arxiv.org/abs/1803.08494

[13] Krogh, A., Hertz, J. A Simple Weight Decay Can Improve Generalization. *Advances In Neural Information Processing Systems*. **4** (1991).

[14] Luo, L., Xiong, Y., Liu, Y., Sun, X. Adaptive Gradient Methods with Dynamic Bound of Learning Rate. *Proceedings Of The 7th International Conference On Learning Representations*. (2019)

[15] Woo, S., Park, J., Lee, J. & Kweon, I. CBAM: Convolutional Block Attention Module. (arXiv, 2018), https://arxiv.org/abs/1807.06521

[16] Khan, S., Shen, Z., Sun, H., Patel, A., Abedi, A. Supervised Contrastive Learning for Detecting Anomalous Driving Behaviours from Multimodal Videos. (ArXiv, 2021), https://arxiv.org/abs/2109.04021

[17] Tüfekci, G., Kayabasi, A., Akagündüz, E., Ulusoy, I. Detecting Driver Drowsiness as an Anomaly Using LSTM Autoencoders. (ArXiv, 2022), https://arxiv.org/abs/2209.05269.

[18] He, K., Zhang, X., Ren, S., Sun, J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778, 2016.

[19] Xie, S., Girshick, R.B., Dollár, P., Tu, Z., He, K. Aggregated Residual Transformations for Deep Neural Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5987-5995, 2017.

[20] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. (ArXiv, 2017) https://arxiv.org/abs/1704.04861