Journal of Computer Science and Cybernetics, V.41, N.2 (2025), 181–196
DOI no. 10.15625/1813-9663/20349

EFFECTIVE OF CONTRASTIVE LEARNING FRAMEWORK IN DRIVER BEHAVIOR ANALYSIS

THANH-HA DO[1*], VU MINH HUNG[2], NGUYEN TRUNG KIEN[3]

[1]*Posts and Telecommunications Institute of Technology, Km10, Nguyen Trai Street, Ha Dong District, Ha Noi, Viet Nam*
[2]*AWL Vietnam LLC, 23 Phan Chu Trinh Street, Hoan Kiem District, Ha Noi, Viet Nam*
[3]*Hanoi University of Science and Technology, 01 Dai Co Viet Street, Hai Ba Trung District, Ha Noi, Viet Nam*

Abstract. The demand for advanced driver behavior analysis systems to support car drivers has arisen, leading to a reduction in accidents. The solutions have been researched and developed for a long time, but the results have recently been acknowledged since some deep learning methods have been published widely. Our paper proposes several modifications to one of the effective deep learning models, Contrastive Learning Framework (CLF), to improve understanding and overall impact. However, it met a lot of challenges such as data imbalance and real-time predicting problems. In more detail, we propose the CENCE loss function for computing comparable visual features both positive and negative, and the Cross Stage Partial Technique (CSPNet and CSPResnet) to improve the outcome in the base encoder. Our approach is evaluated on published datasets, and the obtained results represent some positive performance in the analysis of driver behavior.

Keywords. Contrastive learning, driver behavior, deep learning, combined loss functions.

1. INTRODUCTION

In recent years, with the emergence of new technologies and the rapid growth of hardware computing, driver behavior analysis systems have gradually become important components inside cars, playing a critical role in preventing accidents and ensuring road safety. In these systems, accurately tracking and identifying the driver's distracted behavior in real-time to immediately alert the driver is a must-have feature for preventing potential accidents. The reason is that if the prediction is inaccurate, any error could lead to devastating outcomes, and if the inference speed is not fast enough, the system's alert may come too late.

To address the driver behavior analysis problem effectively, many classical machine learning approaches and deep learning approaches were introduced and achieved remarkable results (details in Section 2). However, these approaches only focused on a handful number of driver's features including face, driving style, or vehicle conditions such as speed, steering wheel, etc. While lacking much information from other features in the context as a whole, hence, they have not worked well in real scenarios. Recently, a deep learning method,

*Corresponding author.
E-mail addresses: dothanhha@ptit.edu.vn (T.H. Do); hung.ttkt1@gmail.com (V.M.Hung); kien.nt200303@sis.hust.edu.vn (N.T.Kien).

namely Contrastive Learning Framework (CLF) [1] achieved prominent results on the Driver Anomaly Detection (DAD) [1] dataset, at above 0.96 in the AUC metric.

Inspired by the success of the CLF, the paper researched and proposed some modifications to CLF to better understand the impact of every component on the overall result and to try to improve its performance. The final objective is to develop a better variant of CLF with more accurate predictions while keeping the real-time inferencing speed, which is vital for driver behavior analysis systems. To develop a reliable deep-learning model for analyzing driver behavior, several challenges are addressed. The first one is data imbalance. Data imbalance is a common issue in deep learning and machine learning, where one class of data significantly outnumbers another. This can lead to models that perform well in the majority class but poorly in the minority class. In driving scenarios, abnormal behaviors such as drowsiness, eating, reading, etc, are rare compared with normal driving behaviors. We did some experiments on imbalanced data. For example, in the DAD dataset [1], the number of normal data recorded was 550 minutes, whereas the number of abnormal data was only one-fifth of that at 100 minutes, although these abnormal behaviors may be more important and must be focused on. Another imbalance problem is the unequal distribution of recorded data between participants. For instance, the MPIIGaze dataset comprises over 213 thousand images from 15 participants, but the data distribution is skewed to only some participants. A participant has only about 1.4 thousand images in the dataset, whereas there are over 35 thousand images from another participant. These problems require efficient data processing methods and sophisticated training techniques to address. Table 1 indicates the data imbalance problems on several driver monitoring datasets.

Table 1: Some published driver monitoring datasets

Dataset	Year	Size	Imbalanced description
DAD	2020	95 GB	550 minutes are recorded for normal driving and 100 minutes for abnormal driving in the training set.
DriverMVT	2022	5.119 million frames	There are a total number of 5.119 million frames, but only a small number of those are the critical events (e.g., Fatigue is nearly 1 thousand frames, cellphone use is several dozen frames, etc.)
DriveandAct	2019	9.6 million images	Only 2% of the data is taking-off sunglasses class, whereas the watch video class accounts for 23%.
MPIIGaze	2017	213.000 images	Images collected by participants varied from over 1.400 to nearly 35.000.

The second one is real-time prediction. In general, the average time the human brain processes and responds to a stimulus is nearly 0.3 seconds. If a driver drives at a speed of 100km/h, it means that his car can move a distance of nearly 28m per second. Assuming that the monitoring system can only detect the abnormal state of the driver and alert him after 1 second, that means the driver can only be aware of the situation after 1.3 seconds or

after a distance of $1.3 \times 28 = 36.4$m. In many cases, it may be too late for the driver to handle if something unexpected happens. That is why real-time predicting of driver behaviors is crucial for the proactive prevention of accidents, enabling vehicles and drivers to respond dynamically to changing conditions. This requires developing an efficient model with low latency while ensuring its predictions are reliable.

This paper proposed some improvements to the CLF model for the driver monitoring problem. The main contributions of the paper include:

1. The paper presents another loss function, the combination of noise contrastive estimation (NCE) loss and cross-entropy (CE) loss. In addition, in cross-entropy loss, we propose a separate projection head dedicated to it. The experiment results showed that the combination of the two losses gave a significant improvement.

2. The paper proposed a modification to the base encoder in the CLF model by upgrading the residual block to a Cross Stage partial residual block, which not only strengthened the learning ability of the model but also reduced computational and memory costs.

3. The paper proposes a method to deal with the imbalanced dataset problem. We evaluated the effectiveness of the data augmentation approach, and the obtained results showed that a data sampling strategy focusing on abnormal data can alleviate the data imbalance problem and improve the capabilities of the system.

The structure of the paper is organized as follows. Section 2 presents the related works for driver behavior analysis. Section 3 impresses the CLF model and the contributions of the paper. Section 4 evaluates the obtained experimental results. Conclusion and future works are indicated in Section 5.

2. RELATED WORKS FOR DRIVER BEHAVIOR ANALYSIS

When analyzing a driver's activities while driving, many aspects can be taken into account. It may include the driver's distracted states, such as eating or drinking, talking to another person or on the phone, or being drowsy. Another aspect is the driver's health condition, such as brain activity, heart rate, and blood pressure. Besides, the working conditions of the vehicle, such as speed, brake, and steering wheel movements, are also worth considering. This information can be captured by cameras, from sensors placed in the cabin, or attached to the driver. Generally, they can be categorized into two feature groups: *non-visual feature* group and *visual feature* group. This section briefly introduces classical machine learning and deep learning approaches for driver behavior analysis corresponding to each group.

2.1. Classical machine learning approaches

In the classical machine learning methods [2, 3, 4], the authors frequently focus on a feature or a limited set of related features of drivers or vehicles and then train classical machine learning classifiers to predict the drivers' states. The common characteristics of these methods are that the number of features is generally small (e.g. *only use vehicle speed, throttle, and acceleration signals from sensors for prediction*) or local (e.g. *only use mouth region from the camera for prediction*). The classifiers usually are SVM [5], Logistic Regression [6], or Random Forest [7].

There are two *non-visual* features that are usually used for driver behavior analysis, driver health condition, and vehicle monitoring. The driver's health condition is analyzed by measuring brain activity using an electroencephalogram (*EEG*). Some researchers exploited EEG information and achieved positive results. In [2], the EEG signals go through a multi-stage system to extract low-dimensional features, and then these features are fed into an extra trees classifier with the output of whether the driver is drowsy or not. The authors conducted experiments on two EEG datasets, Physionet [8] and SVDD [9], reaching the F_1 scores of 93.16% and 75.69%, respectively. However, this approach has some drawbacks. First, personal factors such as age and gender may affect the accuracy of this method. Another problem is the cumbersome nature of this technique. Since this method requires the electrode sensor to come into contact with many points on the driver's head, it may make driving less safe.

Other approaches use vehicle conditions to monitor driver behavior. For example, by using information from the steering wheel, the authors may indirectly predict the concentration level of the driver, whether the driver is distracted or focused on the road. A steering wheel movement with an unexpectedly great angle of correction may be an indicator of the distraction. For instance, the authors in [3] categorized steering wheel features into three subsets: time domain, frequency domain, and state space features. Then, every feature subset was fed into an ensemble classifier which is a combination of five sophisticated machine learning methods: linear SVM, radial kernel SVM, nearest neighbor, decision tree, and logistic regression. The final result is the averaged and dichotomized value from three classification outputs of these subsets. Although this method got promising results (93.3% *specificity on sleepy driver detection on a private dataset*), it and other vehicle features-based methods are based heavily on the environment's condition, such as road surfaces or crosswinds, to be able to work properly.

Recently, the researchers tend to favor *visual feature*-based approaches over non-visual ones since by analyzing visual data such as images or videos, the researchers can gain insights into various aspects of driver behavior, including facial expressions, gaze patterns, head movements, eye movements, and distracted actions such as using phones, eating, talking with passengers, and so on. These approaches enable researchers to identify potential risks, evaluate driver performance, and develop effective methods to improve road safety. Some of the most remarkable studies were included in [10, 11, 4]. In [4], the authors used eye features and suggested a drowsiness detection system comprising two main stages: face/eye detection and tracking, and eye classification. The obtained accuracy of 96.9% over the private dataset. However, the system's performance depends on the light conditions, camera quality, and whether you are wearing glasses.

Another *visual feature* is used, being mouth features. In [12], by using the differences between two successive images, the driver's face region can be extracted, and then the mouth area would be determined by the region between nostrils and chin, if the mouth area is greater than a predefined threshold, then the driver may be yawning, hence drowsy or in other word, low level of concentration. Another method [13] used a multiple-stage system. Firstly, the driver's face is detected using a gravity-center template. The mouth corners are detected and their features are extracted using Gabor wavelets. Finally, the yawning state of the driver can be detected by applying the linear discriminant analysis classifier on the mouth's features. The accuracy of a multiple-stage system is 94.7% over their private dataset. One

more worth mentioning method is [14], in which the authors also developed a lightweight multi-stage system to predict if the driver is yawning. This system will detect the mouth and try to segment it from the face using the color difference between face and lips (mouth); then, based on the number of pixel differences between yawning mouth and mouth at normal state as well as its location on the face, the system can predict yawning state and send the alert. Although these mouth features-based methods may get promising results to some extent, yawning may not be too far from talking or laughing in terms of mouth shape. As a result, it makes the system less reliable in real-life scenarios when the driver may interact with other people in the car.

2.2. Deep learning approaches

Given the fact that classical approaches were only able to focus on a limited group of visual features in a region such as the mouth or face, it may lead to unsatisfactory results due to false positives. Deep learning emerged as a potential direction that can address these issues. By collecting information from features from many regions of the context instead of focusing on one by one region, it can learn complex structures and combination features under the hood that classical approaches could not do.

Similar to classical machine learning approaches, *non-visual features* such as vehicle and driving features also affect directly deep learning approaches. Vehicle and driving features are used in [15], in which authors used multiple signals from various aspects of vehicle driving, including speed, throttle, acceleration, gravity, and Revolutions Per Minute (RPM); and transformed them into image signals using overlapped time windows and recurrence plot technique. Then these image signals were fed into a deep CNN-based classifier with its output as one of the five driving styles: normal (safe), aggressive, distracted, drowsy, and drunk driving. The authors evaluated the performance of the proposed approach over the dataset that comprised over 21 thousand samples, and the obtained result is high with an accuracy of 99.99% and low computational cost.

On the other side, *visual features* such as emotion prediction are used in [16]. In [16], firstly, a mixture of trees with a shared pool of parts model [17] is used to detect and extract the facial Region of Interest (ROI) as well as the facial landmark points from the frame. Next, the first VGG16 model used facial ROI to extract high-level features from it, and the second VGG16 network used facial landmark points to extract corresponding high-level features. Then, a weighted summation combines this information and predicts the driver's final emotional state. The system outperformed all previous state-of-the-art works with 97.3% and 89.7% accuracy on the JAFFE dataset and the MMI dataset, respectively.

The position of the driver's hands is used in [18]. In this method, a CNN model underwent initial pre-training using an unsupervised feature learning technique known as sparse filtering, followed by fine-tuning through a classification process to discern discriminative features directly from raw image data. The effectiveness of this approach was assessed using the Southeast University driving posture dataset, encompassing video clips illustrating four distinct driving postures: normal driving, responding to a cell phone call, eating, and smoking. Comparative analysis against alternative approaches employing diverse image descriptors and classification methods revealed that this method achieved superior performance, boasting an impressive overall accuracy of 99.78%.

In [19], the authors employ a CNN classifier to identify distraction categories. Given

that the risk associated with different types of distractions varies, tailored interventions are needed. To achieve this, an image captured from the vehicle dashboard is input into the neural network, facilitating the recognition of the specific distraction category. To reduce training time, the authors explored the utilization of pre-trained weights derived from training CNN models on the ImageNet dataset. This strategic use of pre-trained weights enhances efficiency while maintaining the classifier's precision. By adopting this strategy, using the VGG19 architecture, an accuracy of 98.98% can be reached on the StateFarm dataset.

To accurately differentiate complex distraction behaviors such as talking on the phone, drinking, reading, etc., from normal driving behavior, all information inside the cabin must be considered. They include not only facial features such as mouth and eyes, or body features such as hands, shoulders, and body posture, but viewing these features from different angles and channels is also essential. However, aggregating information from many sources may be challenging because large variances may have an adverse effect on the system's performance. Besides, it may make the system increasingly complex. To address these problems, the Contrastive Learning Framework along with its fusion strategy was introduced [1]. In this work, Okan *et al.* used many modalities and views from multiple cameras at once, and a multi-3D-CNN-model system was adopted to aggregate these signals. Then, the novel deep contrastive learning strategy was used to train the system on the robust DAD dataset. Based on that, the system could differentiate between normal driving behavior and anomalous driving patterns at a high AUC of 0.96%.

From the obtained results of the state-of-the-art driver behavior analysis, we can see that by analyzing the whole context of the camera's field of view instead of focusing on a single region as introduced in the classical machine learning method, deep learning-based methods can perform very well in many datasets and scenarios in terms of accuracy. Moreover, adopting the deep learning approach on visual features also makes the pipeline more efficient since the approach mostly only requires a single analyzing stage between input data (video/image) and output (e.g., driving state, driver behavior) instead of the multi-stage analyzing approach employed in the classical methods. The contrastive learning approach is the most prominent among these deep learning methods due to its novel architecture and good performance. Therefore, we decided to use it as a baseline model for further studies in the paper.

3. CONTRASTIVE LEARNING FRAMEWORK AND ITS IMPROVEMENTS

This section details the main components of the contrastive learning framework (CLF), analyzes its advantages and disadvantages, and proposes several improvements.

3.1. Contrastive learning framework

Contrastive learning framework consists of two stages: training and testing. In the training stage, CLF consists of three main components: the 3D CNN Base Encoder, the Projection Head, and the noise contrastive Estimation loss as shown in Figure 1.

The 3D CNN Base Encoder in CLF is a ResNet18 model [20], a robust architecture commonly employed in many studies as it stabilizes the training phase and yields better results. When using the 3D variant of ResNet18, the data fed into the model are video clips of drivers

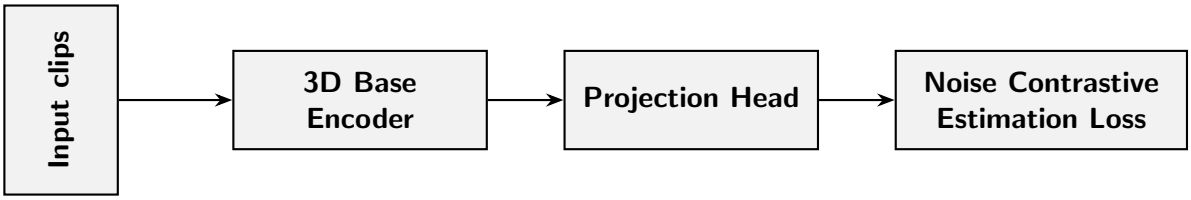

Figure 1: Main components of CLF in training time

in normal and abnormal scenarios from a view-modality camera combination. These clips are small windows with the duration of $d = 16$ consecutive frames with resolution 112×112, sliced from larger videos. Each window clip is called a sample, and every sample is labeled as the normal or abnormal class. A downsampling method called temporal transformation was applied to reduce the informational redundancy of consecutive frames in clips. Basically, given a skip coefficient k, frames in a large video are counted sequentially from 0, then only valid frames satisfying FrameNumber mod $k = 0$ for sample data are kept, while all other frames are excluded. By default, the authors set $k = 2$ for normal and abnormal data creation in training time.

After being created using the above technique, these samples are fed into the 3D-ResNet18 Base Encoder model, where they are transformed into 512-dimensional vectors before continuing to the Projection Head.

The projection head is a multilayer perceptron (MLP) [21], which maps the 512-dimensional vectors from the output of the 3D-ResNet18 Base Encoder into an intermediate representation with 128 dimensions. The Algorithm 1 described every layer inside the projection head. The projection head has 1 hidden layer and *ReLU* activation. The output 128-dimensional vector is then L_2-normalized before going into the loss function.

Algorithm 1 Multilayer Perceptron in the Projection Head for NCE loss

Input: *vector* $x \in R^{512}$
Result: *vector* $v \in R^{128}$
$hidden \leftarrow$ **Linear**$(input \leftarrow 512, output \leftarrow 256)$
$output \leftarrow$ **Linear**$(input \leftarrow 256, output \leftarrow 128)$
$normalize \leftarrow$ **Normalize**$(p \leftarrow 2, dim \leftarrow 1)$
$x \leftarrow hidden(x)$
$x \leftarrow$ **ReLU**(x)
$x \leftarrow output(x)$
$v \leftarrow normalize(x)$
return v

Contrastive representation learning aims to minimize the distances between similar samples and maximize the distances between dissimilar ones. Following works from [22], the authors in [1] also used Noise Contrastive Estimation (NCE) to approximate the contrastive loss in CLF since NCE reduces the computational cost while keeping comparable performance, which is suitable for adapting into a training model. The NCE treats positive pairs between normal samples as data and negative pairs between normal and abnormal samples as noise and tries to distance data from noise. The Equation (1) shows the formula of NCE.

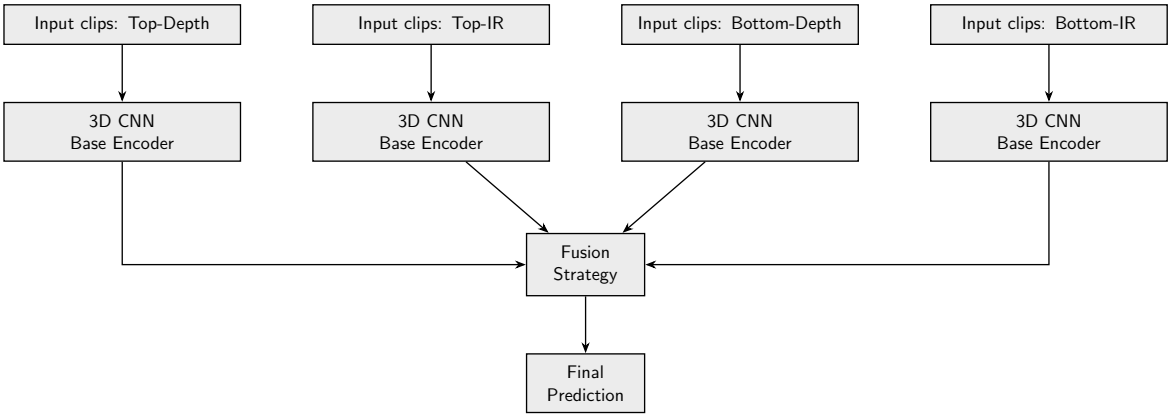

Figure 2: The architecture of the CLF system in testing time

$$\mathcal{L}_{NCE} = -E_{P_n}[log(h(i, v_n))] - mE_{P_a}[log(1 - h(i, v_a))], \tag{1}$$

here, E_{P_n} and E_{P_a} denote the expected values of the log-posterior distribution of data and noises, respectively, and m is the number of negative pairs, $h(i, v_n)$ is the posterior probability of the sample i with feature v_n being in the distribution of data whereas $h(i, v_a)$ is the posterior probability of the sample i being in the distribution of noise

$$h(i, v) = \frac{P_n(i|v)}{P_n(i|v) + mP_a(i)}. \tag{2}$$

In Equation (2), $P_n(i|v_n)$ is the probability distribution of similarity between normal samples, and P_a is the probability of similarity between normal and abnormal samples.

In the testing stage, a fusion strategy combines multiple camera view-modality, as indicated in Figure 2.

The high-level architecture of the CLF system in the testing stage consists of four 3D CNN Base Encoder branches dedicated to four different camera view-modality combinations. The four CNN branches use the same architecture but are trained and evaluated separately on every view-modality input. From every CNN branch k, clip i belonging to that camera view-modality combination will be classified as normal or abnormal by calculating the cosine similarity between its output embedding vector $f_k(x_{ki})$ and the normal representation vector v_k as shown in Equation (3)

$$\text{similarity}_{ki} = v_k^T \frac{f_k(x_{ki})}{\|f_k(x_{ki})\|_2}. \tag{3}$$

After getting all similarity scores from testing data corresponding to every base encoder branch, the output results on all four view-modality data will be aggregated together through a fusion strategy to yield the final prediction. Basically, it is calculated by averaging the similarity score of clips i^{th} across all four view-modality combinations. If the similarity score is greater than a γ classification threshold, the sample will be classified as a normal class; otherwise, it will be classified as an abnormal class.

3.2. Some proposed improvements on the contrastive learning framework

3.2.1. A proposed CENCE loss NCF loss in weighted combination with cross entropy loss

Multi-loss training is a popular strategy used in cases where multiple objectives need to be considered or to generalize the model better. Especially in re-ID problems, researchers usually use multi-loss training strategies [23, 24] such as a combination of Cross Entropy (CE) with Triplet or ArcFace loss to improve the performance of the encoder. The CLF adopted a similar approach to the re-ID problem as the base encoder is treated as a Siamese neural network to compute comparable compressed visual features from negative and positive input images. Based on that, the paper proposes the same approach with the weighted combination of NCE loss and CE loss, which we call CENCE loss. The proposed CENCE loss function has the form as in Equation (4)

$$\mathcal{L}_{CENCE} = \beta\mathcal{L}_{NCE} + (1-\beta)\mathcal{L}_{CE}\ ,\ \text{with}\ 0 \leq \beta \leq 1. \tag{4}$$

Regarding the CE loss component, every sample only belongs to one of the two classes: normal and abnormal, and the objective is to minimize the negative log-likelihood over the training set as Equation (5) follows, in which $f(z_i)$ is the softmax function

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N} y_i \log\left(\frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)}\right). \tag{5}$$

Besides the current projection head for NCE loss (see Algorithm 1), the paper proposes to add a new head dedicated to CE loss to reduce the potential bottleneck problem in the single projection head architecture. Detail of the projection head for CE loss is indicated in Algorithm 2.

Algorithm 2 Proposed dedicated Projection Head for CE loss

Input: *512D vector* x
Result: *2D vector* v
$hidden1 \leftarrow$ **Linear**$(input \leftarrow 512, output \leftarrow 256)$
$hidden2 \leftarrow$ **Linear**$(input \leftarrow 256, output \leftarrow 64)$
$output \leftarrow$ **Linear**$(input \leftarrow 64, output \leftarrow 2)$
$normalize \leftarrow$ **Normalize**$(p \leftarrow 2, dim \leftarrow 1)$
$x \leftarrow hidden1(x)$
$x \leftarrow$ **ReLU**(x)
$x \leftarrow hidden2(x)$
$x \leftarrow$ **ReLU**(x)
$x \leftarrow output(x)$
$v \leftarrow normalize(x)$
return v

The CE projection head maps the 512-dimensional vector space into a 2-dimensional vector space as input of the CE loss. To avoid squeezing into 2D space too quickly, it has two hidden layers instead of only one, as seen in the NCE projection head.

3.2.2. Enhanced the 3D CNN base encoder with cross stage partial technique

Much research has been done to improve the capability of CNNs architecture. For example, ResNeXt [25] introduced aggregated transformations to the residual block called cardinalities, which can effectively increase accuracy while decreasing the complexity of the original ResNet. Another work is CSPNet [26] in which researchers proposed partitioning the feature map of a base layer into two parts, one part goes through traditional CNN blocks such as Residual, ResNext, or Dense block, and then fused with the other part through a cross-stage hierarchy. In this paper, we also adopt that idea into the ResNet backbone to improve accuracy and speed simultaneously, instead of trading off accuracy for speed and vice versa.

Figure 3 demonstrates three types of fusion strategies used in CSPNet. In *d type: Fusion last*, the second gradient flow went through a residual block and a transition layer before being fused with the first gradient flow in the later stage. In *c type: Fusion first*, two gradient flows were concatenated together before going through a later transition layer. Finally, in *b type: CSPResnet* strategy, a hybrid flow between *c type* and *d type* was adopted. We will sequentially conduct experiments with these three fusion strategies in the experiment section to better understand each strategy.

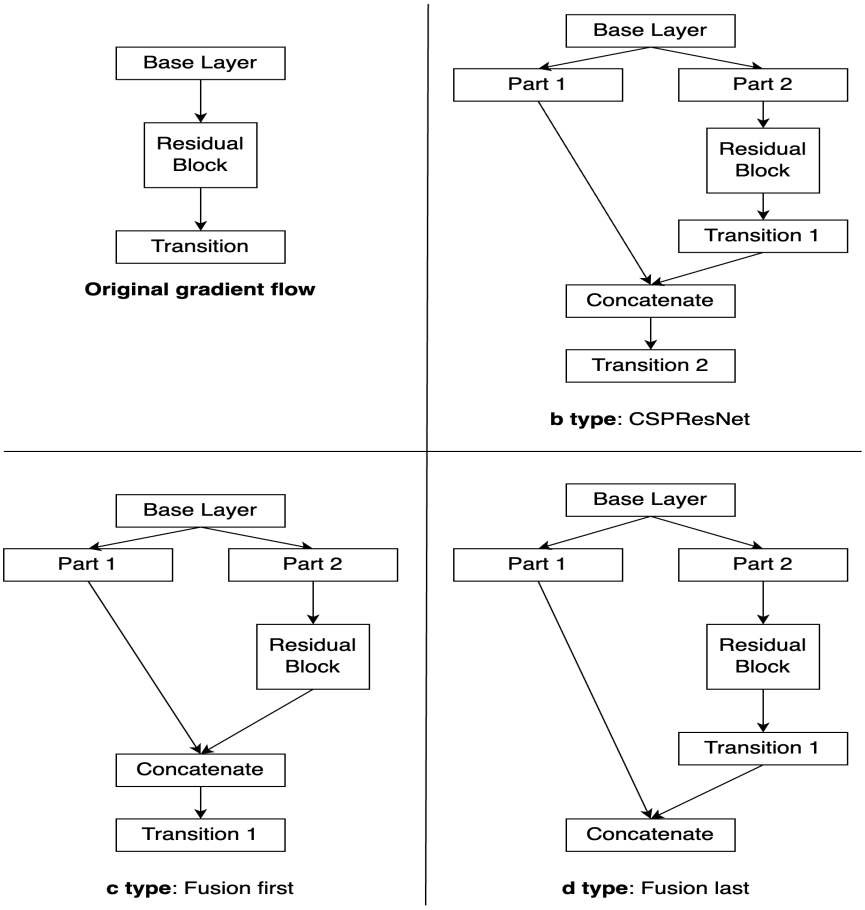

Figure 3: Three fusion strategies for the Cross Stage Partial technique

3.2.3. Data imbalancing alleviation

There is a great imbalance between normal and abnormal labels in the dataset, with the number of normal data nearly four times the number of abnormal data. Therefore, alleviating this problem may positively impact the model's performance.

Besides many spatial data augmentation techniques introduced in [1], such as rotating, horizontal flipping, salty noise, and scaling. We focused more on the temporal transformation of the data, which is a way to reduce redundancy in data clips. In the original setting, normal and abnormal data were created using the same step size $k = 2$. To make the dataset more balanced, we increase the step size for normal data by 1, and keep the step size for abnormal data in the original setting as shown in Figure 4.

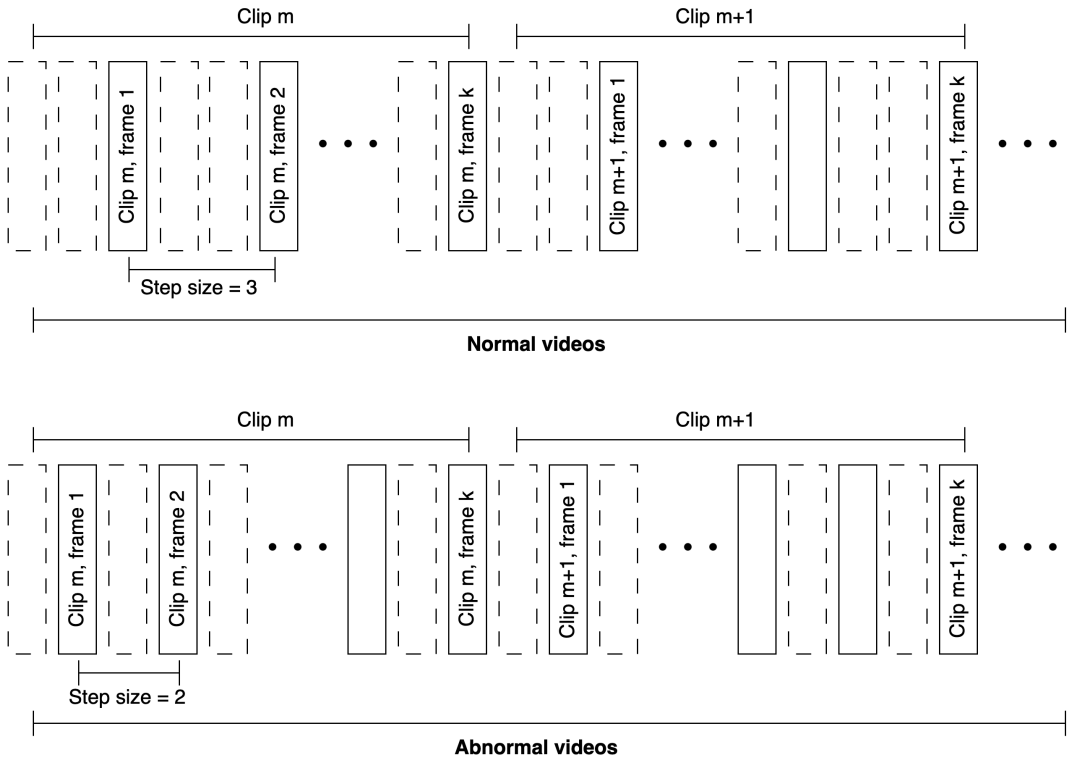

Figure 4: Improve temporal transformation for the data by using two different step sizes for normal and abnormal data.

We assumed that in a normal driving state, consecutive frames usually do not contain too much valuable information. Hence, reducing the number of normal clips by increasing the step size will not harm the data's quality. Instead, the model's performance may benefit due to a more balanced ratio between normal and abnormal clips.

4. EXPERIMENTS AND EVALUATION

4.1. Dataset

In this section, we conducted some experiments for our proposals on a subset of the Driver Anomaly Detection dataset. The reason we chose the sub-DAD dataset instead of the whole dataset is that we do not have many resources to conduct experiments on the whole DAD dataset. As a result, we only took a part of this DAD dataset, which we called the sub-DAD dataset, including 35GiB/95GiB videos on the DAD dataset chosen randomly and with the same distribution as the DAD dataset. In more detail, the statistics of video duration between normal and abnormal classes in the DAD and sub-DAD datasets indicated that the number of normal data is nearly four times that of abnormal data. Furthermore, in the training data, the normal data is nearly five times the abnormal data.

4.2. Measures

The paper uses the Area Under the ROC curve (AUC) metric for all experiment and evaluation. The AUC serves as a metric gauging the binary classifier's proficiency in discriminating between classes, providing a concise summary of the Receiver Operator Characteristic (ROC) curve. A higher AUC value indicates a better model.

4.3. Experimental settings

We use the following experimental strategy to reduce the number of experiments while still sufficiently evaluating the impact of every modification on the model's performance.

1. The model is trained with the best settings suggested in [1] and used as the baseline model. In the baseline model, ResNet18 model is the encoder because of the lightweight and robust characteristics, the number of epochs being 100, and the minibatch gradient descent optimizer is used with a batch size of 160 (*150 for abnormal clips, and 10 for normal clips*), a momentum of 0.9, and a learning rate of 0.01 for the first 50 epochs and 0.001 for the later 50 epochs. This came from several trials initially with 250 epochs, but they exceeded 24 hours of training on the Colab environment, which had NVIDIA Tesla T4 16GB GPU.

2. Next, we developed the proposed improvement directly to the baseline model. The purpose is to compare the model performance before and after applying the new technique and compare various hyper-parameter settings within the proposed technique. Then, we will have the best settings using every proposed technique that will become our candidates for the final model.

3. At the end, after getting numerous combinations of proposed techniques from the previous steps, we assessed them one by one till the end to find out the most prominent.

To see the impact of the additional CE loss, the paper started with a dominant value of NCE loss ($\beta = 0.9$ *in proportion to the loss combination, and then gradually increased the* β *value*). Results corresponding to every β value are shown in Table 2, $\beta = 1$ indicates that it is the baseline model (*there is only NCE loss being used*).

Table 2: AUC results with various β value settings in CENCE loss function

Value of β	0.4	0.5	0.6	0.9	1
AUC	0.92	**0.94**	0.93	0.89	0.92

Interestingly, an equal contribution of the CE and NCE functions to the combined loss function yielded the best results, while a small proportion of CE in the combined loss degraded the model's performance. An explanation for this is that adding a small amount of signal from another source may act as noise, and confuse the model, therefore it harms the model's capability. However, when the amount of this signal is more significant, its signal becomes more comprehensive, which benefits the model's performance.

Furthermore, we studied the influence of the cross-stage partial (CSP) technique by experimenting sequentially with all three strategies mentioned in [26] to see how they perform, and the results are shown in Table 3.

Table 3: Results of three CSP fusion strategies on the Base encoder

Fusion Strategy	Evaluating time (s)	AUC
No fusion (*Original gradient flow*)	2132	0.921
Mix fusion (*b type*)	2110	**0.934**
Fusion first (*c type*)	2045	0.909
Fusion last (*d type*)	2148	0.887

In three fusion strategies, CSPResnet demonstrated a more remarkable ability to increase the model's accuracy in the AUC metric and outperformed the baseline by a wide margin of over 0.012 points. In contrast, fusion first and fusion last fell short after it. These results were expected, as observed in [26], although the computational cost advantage represented by evaluation time was still unclear in all of these fusion strategies.

The next experiment will evaluate the technique used to deal with imbalanced data. Given that the number of normal clips several times exceeds the number of abnormal clips and the larger the stepsize value is, the fewer clips are created, we proposed using the different stepsize values for normal and abnormal data separately. We tried to increase the stepsize value for normal clips and keep or decrease the stepsize value for abnormal clips to alleviate the imbalance problem in the dataset. Table 4 illustrated the AUC results with various stepsize values for creating normal and abnormal data. There was clear evidence that downsampling normal data by increasing the stepsize value benefited the model's performance.

Table 4: Results of data sampling strategies on the model

Sampling Strategy	AUC
The Baseline (*Normal stepsize = Abnormal stepsize = 2*)	0.921
Normal stepsize = 3, Abnormal stepsize = 2	**0.924**
Normal stepsize = 3, Abnormal stepsize = 1	0.906

After conducting separate experiments for each proposal, the paper combined all pro-

posals with the best settings into a single model to evaluate its capability on the DAD sub-dataset. The result is shown in Table 5.

Table 5: Results of combining all proposed techniques in baseline model

Model	AUC
The Baseline	0.921
Best CENCE loss setting ($\beta = 0.5$)	**0.940**
Best CSP setting (Table 3)	0.934
Best sampling setting (Table 4)	0.924
Combination of CENCE + CSP	0.938
Combination of CENCE + proposed data sampling	0.920
Combination of CENCE + CSP + proposed data sampling	**0.940**

Interestingly, although both CENCE loss and CSP technique got better AUC results compared with the baseline, combining these two only made the new combined model better than the baseline at 0.938 but still fell short in the model with only CENCE loss at 0.940. Combining all three techniques, including CENCE loss, CSP and proposed data sampling only improved the result slightly from 0.938 to 0.940 but it still does not overcome the result of the model with only CENCE loss.

5. CONCLUSION

The paper proposed several improvements to the Contrastive Learning Framework and analyzed the impact of these proposals through extensive experiments. These experiments focus on three main factors of the CLF framework: the objective function, the base encoder, and data imbalance. On the objective function, the paper proposed a weighted combination between the CE function and the NCE function. By applying weighting to these two loss functions, the CLF's capacity improved significantly. On the base encoder, the paper proposed a gradient flow splitting technique called cross-stage partial in which the feature map is partitioned into two parts, one of which goes through the Residual block, then they are fused into a new feature map through a cross-stage hierarchy manner. By implementing the CLF model along with three fusion strategies at the base encoder stage, the paper indicated that the *b type* variant strengthened the model's learning ability by a gain of over 1% on the AUC metric whereas, in the other two variants, the model's performance was degraded significantly. The final factor was the data imbalance problem, which was also dealt with in this paper by applying temporal transformation using different step sizes for normal and abnormal data, in which the step size for normal data was greater. The experimental results showed that all three proposed techniques helped to improve marginal performance in driver behavior analysis. In the future, we aim to conduct further studies in this field to evaluate performance in terms of accuracy and resource utilization. Additionally, data quality and relevance warrant more attention. The most critical features in driver monitoring are found in the driver's body and gestures, while the background information inside the car generally contains fewer valuable features.

REFERENCES

[1] O. Kopuklu, J. Zheng, H. Xu, and G. Rigoll, "Driver anomaly detection: A dataset and contrastive learning approach," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 91–100.

[2] V. P. B and S. Chinara, "Automatic classification methods for detecting drowsiness using wavelet packet transform extracted time-domain features from single-channel eeg signal," *Journal of Neuroscience Methods*, vol. 347, p. 108927, 2021.

[3] J. Krajewski, D. Sommer, U. Trutschel, D. Edwards, and M. Golz, "Steering wheel behavior based estimation of fatigue," in *Fifth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, 01 2009, pp. 118–124.

[4] B. Cyganek and S. Gruszczyński, "Hybrid computer vision system for drivers' eye recognition and fatigue monitoring," *Neurocomputing*, vol. 126, p. 78–94, 02 2014.

[5] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[6] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, 1958.

[7] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282 vol.1.

[8] G. Moody, R. Mark, and A. Goldberger, "Physionet: a research resource for studies of complex physiologic and biomedical signals," *Computers in Cardiology*, vol. 27, pp. 179–82, 02 2000.

[9] W.-L. Zheng and B.-L. Lu, "A multimodal approach to estimating vigilance using eeg and forehead eog," *Journal of Neural Engineering*, vol. 14, 11 2016.

[10] M. Sabet, R. A. Zoroofi, K. Sadeghniiat-Haghighi, and M. Sabbaghian, "A new system for driver drowsiness and distraction detection," in *20th Iranian Conference on Electrical Engineering (ICEE2012)*, 2012, pp. 1247–1251.

[11] J. Jo, S. Lee, K. Park, I.-J. Kim, and J. Kim, "Detecting driver drowsiness using feature-level fusion and user-specific classification," *Expert Systems with Applications*, vol. 41, p. 1139–1152, 03 2014.

[12] Y. Lu and Z. Wang, "Detecting driver yawning in successive images," in *2007 1st International Conference on Bioinformatics and Biomedical Engineering*, 2007, pp. 581–583.

[13] X. Fan, B.-C. Yin, and Y.-F. Sun, "Yawning detection for monitoring driver fatigue," in *2007 International Conference on Machine Learning and Cybernetics*, vol. 2, 2007, pp. 664–668.

[14] S. Abtahi, B. Hariri, and S. Shirmohammadi, "Driver drowsiness monitoring based on yawning detection," in *2011 IEEE International Instrumentation and Measurement Technology Conference*, 2011, pp. 1–4.

[15] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Systems with Applications*, vol. 149, p. 113240, 07 2020.

[16] B. Verma and A. Choudhary, "Deep learning based real-time driver emotion monitoring," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2018, pp. 1–6.

[17] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2879–2886.

[18] C. Yan, B. Zhang, and F. Coenen, "Driving posture recognition by convolutional neural networks," *IET Computer Vision*, vol. 10, 10 2015.

[19] S. Masood, A. Rai, A. Aggarwal, M. Doja, and M. Ahmad, "Detecting distraction of drivers using convolutional neural network," *Pattern Recognition Letters*, vol. 139, pp. 79–85, 2020.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[21] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1994.

[22] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, no. 2018, 2018, pp. 3733–3742.

[23] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3701–3711.

[24] Z. Dai, M. Chen, X. Gu, S. Zhu, and P. Tan, "Batch dropblock network for person re-identification and beyond," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3690–3700.

[25] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5987–5995.

[26] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1571–1580.

Received on March 15, 2024
Accepted on May 8, 2025