Towards Efficient Driver Distraction Detection with DARTS-Optimized Lightweight Models

Yassamine Lala Bouali[1,2][a], Olfa Ben Ahmed[1][b], Smaine Mazouzi[2][c] and Abbas Bradai[3][d]

[1] *XLIM Research Institute, URM CNRS 7252, University of Poitiers, France*

[2] *Computer Science Dept., University of 20 Aout 1955, Skikda, Algeria*

[3] *University Cote d'Azur LEAT, CNRS UMR 7248, Biot, France*

Keywords: Human-Computer Interaction, Emotional States, Driver Distraction, DAS, Deep Learning, NAS, DARTS.

Abstract: Driver Distraction is, increasingly, one of the major causes of road accidents. Distractions can be caused by activities that may shift the driver's attention and potentially evoke negative emotional states. Recently, there has been notable interest in Driver Assistance Systems (DAS) designed for Driver Distraction Detection (DDD). These systems focus on improving both safety and driver comfort by issuing alerts for potential hazards. Recent advancements in DAS have prominently incorporated deep learning techniques, showcasing a shift towards sophisticated and intelligent approaches for enhanced performance and functionality. However, model architecture design is mainly based on expert knowledge and empirical evaluations, which are time-consuming and resource-intensive. Hence, it is hard to design a model that is both efficient and accurate at the same time. This paper presents a Neural Architecture Search (NAS)-based approach for efficient deep CNN design for DDD. The proposed approach leverages RGB images to train a lightweight model with few parameters and high recognition accuracy. Experimental validation is performed on two driver distraction benchmark datasets, demonstrating that the proposed model outperforms state-of-the-art models in terms of efficiency while maintaining competitive accuracy. We report 99.08% and 93.23% with model parameter numbers equal to 0.10 and 0.14 Million parameters for respectively SFD and AUC datasets. The obtained architectures are both accurate and lightweight for DDD.

1 INTRODUCTION

With the development of smart vehicles, Driver Assistance System (DAS) in human-centered transportation has attracted much attention in recent years (Xing et al., 2021). Using an intuitive Human-Machine interface, such systems aim to enhance driver comfort, ensure safety, and assist drivers. Driver monitoring in terms of emotions, behaviors, and actions is a key application of DAS to control the driver's mood and emotions (McCall and Trivedi, 2006). In this context, Affective Computing is revolutionizing the automotive industry by creating DAS capable of recognizing, interpreting, processing, and responding to human emotions and behaviors (Nareshkumar et al., 2023). Through the integration of sensors, cameras, and AI algorithms, vehicles can detect signs of fatigue, stress, or distraction. This prompts the vehicle to issue alerts or take corrective actions, thereby ensuring driver safety.

Driver distraction is a major cause of road accidents. According to recently published statistics (for Statistics and Analysis, 2023), eight percent of fatal car accidents are due to distraction. Indeed, nowadays, drivers are continually bombarded with potential distractions due to the widespread use of smartphones, infotainment systems, and various other in-car technologies. These circumstances can compromise the driver's attentiveness and disturb their overall mood, thereby impacting their ability to drive safely. Driver distraction can be categorized into three main types (Lee, 2005): 1) visual distraction, such as diverting one's gaze away from the roadway, 2) cognitive distraction, which involves the mind being diverted from the road, and 3) manual distraction, including activities like responding to a ringing cell phone. It is worth noting that distractions caused by

[a] https://orcid.org/0000-0002-2133-6086
[b] https://orcid.org/0000-0002-6942-2493
[c] https://orcid.org/0000-0003-3587-7657
[d] https://orcid.org/0000-0002-6809-4897

Lala Bouali, Y., Ben Ahmed, O., Mazouzi, S. and Bradai, A.
Towards Efficient Driver Distraction Detection with DARTS-Optimized Lightweight Models.
DOI: 10.5220/0012595400003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 1*, pages 480-488
ISBN: 978-989-758-680-4; ISSN: 2184-433X

a driver's activities can potentially lead to a shift in emotional state. Indeed, recent psychological studies have shown that driver emotions can be aroused and activated by driver activities, such as a call phone, texting information, or radio information (Fernández et al., 2016). Concretely, attentive drivers focus on the vehicle, the traffic, and the surroundings enabling them to anticipate to unforeseen dangers. A serious problem can arise when a driver loses attention and becomes concentrated on an extra event (activity) that affects his mental and emotional states. For example, using a phone and having a nervous conversation can transform the affective state of the driver and decrease driving performance and concentration. Such an event or behavior redirects the driver's attention and makes driving difficult and unsafe.

Artificial Intelligence has significantly revolutionized the DAS by the investigation of deep learning techniques. Deep learning, especially Convolutional Neural Networks (CNNs), have been widely used for driver behavior (Shahverdy et al., 2020) and emotions (Zepf et al., 2020) recognition. Existing models are hand-designed, and obtained after several experiments on different architectures and substantial parameters tuning. Despite good model performance, the parameter size poses a significant challenge for real-world applications due the limitations of vehicle-mounted computing equipment.

In this paper, we propose a gradient-based NAS method for automatic deep neural network design in the context of DDD. The proposed method is based on a Differentiable Architecture Search (DARTS). The latter is known for its reduced search cost, compared to non-differentiable NAS, and flexibility for searching for high-performance architectures. We use RGB images to search for light models with few parameters and high recognition accuracy. We conduct experiments on two driver distraction benchmark datasets namely, the State Farm Distracted Driver Dataset (SFD) and the American University in Cairo Distracted Driver Dataset (AUC). To the best of our knowledge, our work is the first to investigate DARTS for a real-world application, namely driver distraction detection. The rest of the paper is organized as follows: Section 2 discusses recent works on deep learning-based methods for DDD. Section 3 describes the proposed method. Section 4 presents experiments and results and finally, section 5 concludes the work and opens new perspectives.

2 RELATED WORK

Driver distraction detection field has been notably influenced by the transformative capacities of deep learning, especially Convolutional Neural Networks (CNNs) (Li et al., 2021). Therefore, a variety of approaches using multiple data types and sensors have been proposed in the literature for the DDD. For instance, some works investigated multi-sensing data (Nidamanuri et al., 2022) (Das et al., 2022) and biological signals (Chen et al., 2022) (Dolezalek et al., 2021). However, fusing data from different sensors is complex and requires the presence of all sensors in the prediction phase. Moreover, leveraging physiological data, to infer and understand the cognitive and emotional states of drivers can be deemed invasive due to their reliance on physiological measurements. Visual data, namely RGB images, has emerged as the most effective and affordable information due to its non-intrusive nature. This practicality makes it suitable for real-world applications (Zeng et al., 2022). In this context, CNNs have been extensively trained on large-scale imaging datasets for the DDD (Koay et al., 2022). For instance, (Ai et al., 2019) proposed an attention-based CNN combined with VGG16 and built an accurate model with 140M parameters. (Dhakate and Dash, 2020) integrated features extracted from RESNET, InceptionV3, Xception, and VGG networks and trained a second-level neural network and achieved an accuracy of 92.20% on the State Farm dataset (SFD) with 25.60 M parameters. Similarly, (Eraqi et al., 2019) utilized a genetic algorithm to assign weights to a CNN ensemble and achieved 94.29% accuracy with 62.00 M parameters on the American University in Cairo dataset (AUC). (Huang and Fu, 2022) proposed a deep 3D residual network with an attention mechanism and encoder-decoder for predicting the true driver's focus of attention. (Wang and Wu, 2023) enhanced the generalization of DDD using multi-scale feature learning and domain adaptation, achieving an accuracy of 96.82% on SFD with 23.67 M parameters.

However, the aforementioned models remain too large. Indeed, the automotive context requires lightweight solutions, and neglecting such constraints may result in models that are accurate but inefficient. Addressing this challenge, recent works have proposed hand-crafted lightweight models such as MobileNetV2-tiny, (Wang et al., 2022b) and MTNet (Zhu et al., 2023). Nevertheless, manually designing CNN is a time-consuming and iterative task that often requires a high level of expert knowledge. Moreover, the iterative nature of the design process implies training models until a satisfactory result leading to

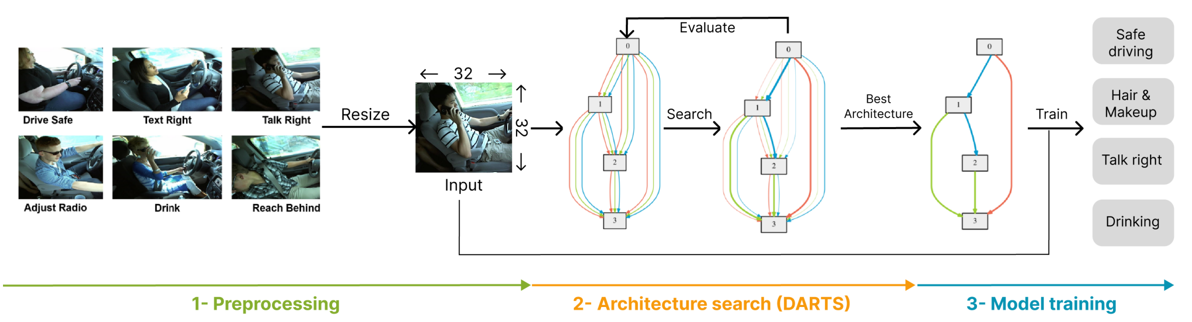


Figure 1: Illustration of the proposed approach.

an excessive consumption of resources.

Recently, Neural Architecture Search (NAS) has emerged as a new paradigm to address this challenge by automating the design of neural architectures. NAS has been widely used for computer vision applications (Kang et al., 2023). However, despite its potential, scarce are the works that have explored NAS for detecting driver distraction. Recently, (Seong et al., 2022) employed reinforcement learning in conjunction with a weight-sharing method for real-time recognition of driver behavior. They gathered their own data and found that their proposed model outperformed hand-crafted models. However, the lack of testing on benchmark datasets makes a comprehensive evaluation impossible. Moreover, (Zaman et al., 2022) integrated an enhanced faster R-CNN with NasNet large CNN to identify driver emotions. They also used a private dataset and compared their model with static emotion recognition datasets. In addition, (Chen et al., 2021) fused data from multiple sources and used NAS to generate a CNN architecture that identify normal driving and distraction states. However, the resulting CNN was large and did not fully meet the specific requirements of its intended use. Lastly, (Liu et al., 2023) presented a NAS-based teacher-student model with knowledge distillation for the same task. This study achieved a lightweight model with 0.42 M parameters. To our knowledge, this is the unique study that has used benchmark public datasets and NAS for DDD, making it the most pertinent reference to our work.

3 PROPOSED APPROACH

Considering the pivotal importance of detecting driver distraction, it is crucial to emphasize the need for a detection model that not only proves effective but is also lightweight enough for practical deployment. In this section, we describe the proposed approach to efficiently detect driver distraction. Figure 1 illustrates the main steps of our approach: 1) Preprocessing, 2) Architecture search, and 3) Model training.

First, we preprocess the input data to ensure it is in a suitable format for our model. Therefore, we resize the images to 32x32 pixels. We believe that this is a crucial step that contributes to achieving a balance between precision and efficiency. Moreover, using a smaller image size during the search process can accelerate the exploration, as it reduces computational requirements. This facilitates a more efficient and faster architecture search process, helping in the discovery of lightweight yet effective model architectures.

Second, we perform a Differentiable Architecture Search (DARTS) (Liu et al., 2019) to look for the best network architecture for our DDD task. We investigate DARTS as a cutting-edge technique that automates the exploration of a diverse space of neural network architectures. Therefore, DARTS facilitates the search for architectures that excel in both efficiency and accuracy.

The architecture search problem is formulated as a bi-level optimization problem. As stated in Eq. 1, in the upper level, DARTS searches for an architecture by minimizing a validation loss using gradient descent. Simultaneously, in the lower level, the algorithm fine-tunes the neural network weights based on the architecture identified in the upper-level optimization.

$$\begin{aligned} &\min_{\alpha} \mathcal{L}_{val}(w^*(\alpha), \alpha) \\ \text{s.t.} \quad & w^*(\alpha) = \arg\min_{w} \mathcal{L}_{\text{train}}(w, \alpha) \end{aligned} \tag{1}$$

DARTS achieves computational efficiency by representing the architecture search space as a directed acyclic graph (DAG) with N nodes. Each directed edge (i, j) in the graph is associated with a set of candidate operations $o^{(i,j)}$ transforming node $x^{(j)}$:

$$x^{(j)} = \sum_{i<j} o^{(i,j)}(x^{(i)}) \tag{2}$$

The continuous distribution of weights or probabilities, modeled using the Softmax function, allows for gradient-based optimization in the search space. Therefore, the architecture gradient is approximated as follows:

$$\nabla_{\alpha} L_{\text{val}}(w - \xi \nabla_{w} L_{\text{train}}(w, \alpha), \alpha) \tag{3}$$

We apply the search space defined in DARTS (Liu et al., 2019), i.e., a supermodel formed by repeatedly stacking normal and reduction cells. Each cell is a collection of nodes. The operations involved in darts are max pooling 3x3, average pooling 3x3, skip connect, separable convolution 3x3 and 5x5, and dilated convolution 3x3 and 5x5.

Third, following the architecture search, we meticulously train the resulting models on two distinct benchmark datasets. This approach allows us to analyze the generalization capabilities of the architectures and provide further insights into the proposed methodology. Finally, we rigorously evaluate the performance of our model on the testing dataset.

4 EXPERIMENTS AND RESULTS

4.1 Distracted Drivers Datasets

We evaluate our approach on two benchmark datasets illustrated in Figure 2, namely, the State Farm Distracted Driver Dataset (SFD) (Anna Montoya, 2016) and the American University in Cairo Distracted Driver Dataset (AUC) (Eraqi et al., 2019).

Figure 2: Sample images from SFD and AUC.

State Farm Distracted Driver Dataset (SFD). is made up of 22,424 images, all of which were taken from video footage recorded with cameras positioned on a car's dashboard. Each image in the collection is tagged with the specific activity the driver is engaged in at the time the image was captured. These activities include safe driving (0), texting on the right (1), talking on the phone-right (2), texting-left (3), talking on the phone-left (4), operating the radio (5), drinking (6), reaching behind (7), hair and makeup (8), and talking with a passenger (9). This dataset has been extensively used in research and has contributed to the creation of a variety of models for detecting driver distraction. We split the dataset into three sets : 60% for training, 10% for validation, 30% for testing.

American University in Cairo Distracted Driver Dataset (AUC). is made up of video footage that captures drivers engaging in various activities. The videos were recorded from two distinct perspectives and each video is approximately 10 minutes in duration. The dataset encompasses a total of 44 participants, with 29 males and 15 females, and includes over 17000 frames. The images are categorized into the following classes: safe driving (0), texting left (1), talking on the phone-left (2), texting right (3), talking on the phone-right (4), adjust the radio (5), drinking (6), reaching behind (7), hair and makeup (8), talking to passenger (9). The dataset is already split into train and test sets by the original authors. In addition, we use 10% of the training data to perform validation.

4.2 Experimental Setups

Hyperparameters and Preprocessing: we employed the Cosine Annealing scheduler to dynamically modify the learning rate with a lower limit of 1e-3. The initial learning rate was fixed to 0.025. Cross-validation and early stopping, with a patience of 10 iterations, were also incorporated during the training phase of the final architecture. The number of epochs varied, as the requirements for training and searching differed for each dataset. Specifically, training was conducted over a maximum of 60 epochs, each consisting of 1900 steps, while the search process lasted for 3 epochs of 1700 iterations each. We used the corss-entropy loss and SGD optimizer.

Architecture Search: to automatically find the best-performing architecture, we conduct a differentiable architecture search (DARTS) on both SFD and AUC datasets. With regards to our specific task, i.e., DDD, we carefully initialize our supermodel with a width of 8 and 8 stacked cells. The model's complexity and the search cost are significantly impacted by two primary hyperparameters: the width and the number of channels. The width, which refers to the number of neurons in a layer, and the number of channels, indicating the depth of the feature maps, are both experimentally set to eight in our supermodel. As a

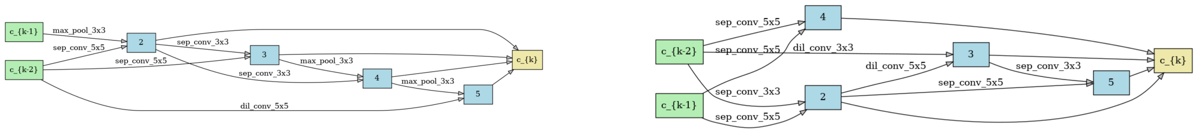


Figure 3: Cells structures of final architecture on SFD.

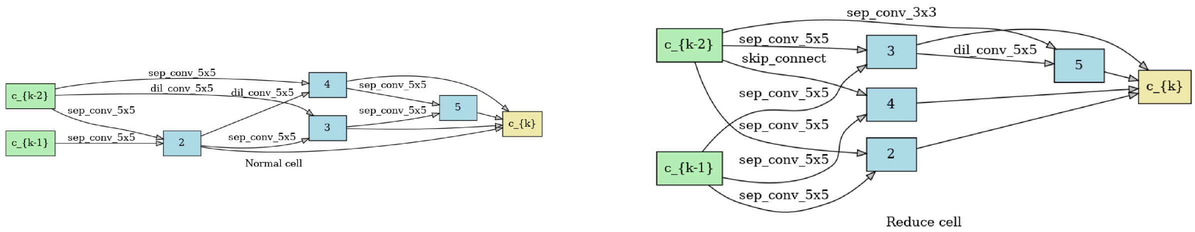


Figure 4: Cells structures of final architecture on AUC.

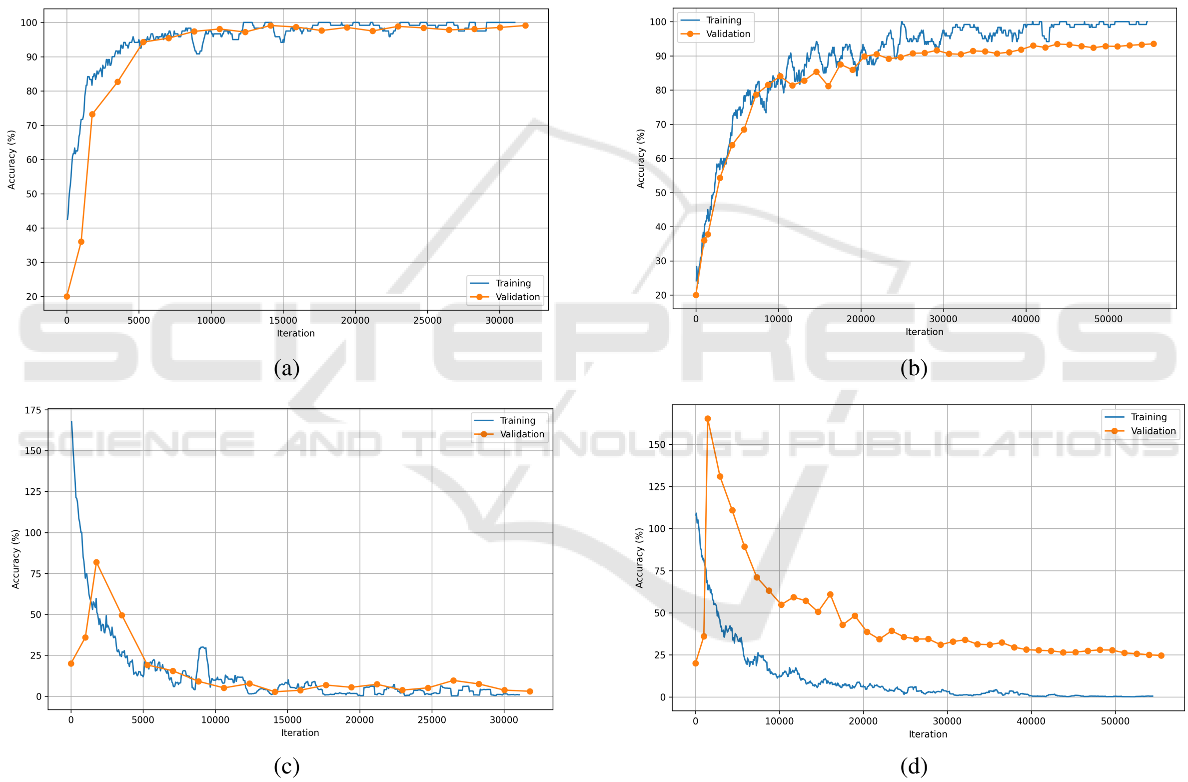


Figure 5: Training performances of both architectures on SFD (left column) and AUC (right column).

result, our supermodel contains approximately 0.96 million parameters.

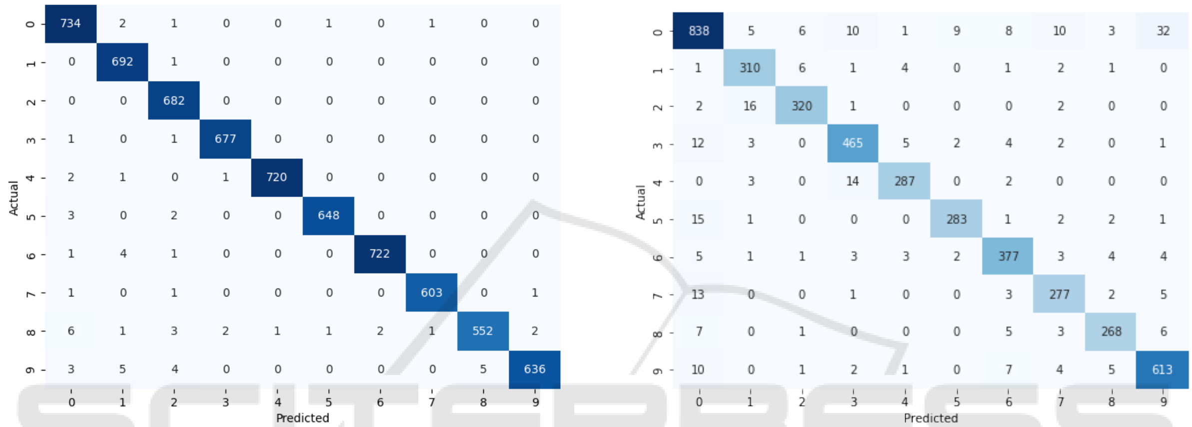
We then run the search for a total of 3 epochs with 1766 steps per epoch. Figure 3 illustrates the cell structures of the final architectures. We notice that, on SFD, the cells mostly contain separable convolution operations as well as a few dilated convolutions and maxpooling. Similar operation choices are observed on AUC dataset in Figure 4. This consistency across different datasets may indicate that the discovered architecture is potentially generalizable and not overfitting to a specific dataset.

4.3 Model Training and Classification Results

Following the architecture search, we then train the resulting architectures on both SFD and AUC separately. We evaluate the classification performances of the models through various metrics including : 1) validation loss and accuracy, 2) test accuracy, 3) preci-

Table 1: Achieved Recall, Precision, and F1 score for each class of SFD / AUC.

	Precision		Recall		F1-score	
Driver Activity	SFD	AUC	SFD	AUC	SFD	AUC
Safe Driving	0.98	0.93	0.99	0.91	0.99	0.92
Texting - Right	0.98	0.94	1.00	0.94	0.99	0.94
Talking on the phone - Right	0.98	0.95	1.00	0.94	0.99	0.95
Texting - Left	1.00	0.91	1.00	0.95	1.00	0.93
Talking on the phone - Left	1.00	0.96	0.99	0.94	1.00	0.95
Operating the radio	1.00	0.96	0.99	0.93	0.99	0.94
Drinking	1.00	0.92	0.99	0.94	0.99	0.93
Reaching behind	1.00	0.91	1.00	0.92	1.00	0.91
Hair and makeup	0.99	0.94	0.97	0.92	0.98	0.93
Talking to passenger	1.00	0.93	0.97	0.95	0.98	0.94



(a) (b)

Figure 6: Confusion matrices of both architectures on : (a) SFD and (b) AUC.

sion, 4) recall and, 5) F1-score. In addition, we provide further insights on the models' efficiency by reporting the number of parameters, inference time and search cost in Table 3.

4.3.1 Performance Evaluation

The learning curves (accuracy and loss) on each dataset are illustrated in Fig. 5 where we observe a faster training on SFD, taking nearly half of the time required for AUC. This can be attributed to the challenging nature of AUC, primarily due to imbalanced data distribution.

Table 2: Classification results.

Dataset	$L_{val} \downarrow$	Acc_{val} (%) $\uparrow$	Acc_{test} (%) $\uparrow$
SFD	0.03	99.11	99.08
AUC	0.25	93.53	93.23

For the same underlying reasons, the classification results, reported in Table 2, demonstrate increased accuracy on SFD. Indeed, we notice a test accuracy of 99.08% for SFD and 91.85% for AUC. Moreover, Fig. 6a shows the confusion matrix of each dataset. We notice that only a few images from SFD are wrongly classified. For instance, six images of "hair and makeup" were classified as "safe driving" likely due to the similarity between these classes. This misclassification may be attributed to the subtle similarity in the head orientation. Hair and makeup involves moving hands while the posture of the head may remain the same, i.e., focused on the road, which can be considered as safe. Similarly, on AUC, up to 32 images from "safe driving" were misclassified as talking to passenger. Table 1 further emphasize these results and presents precision, recall and f1-scores. The table clearly indicates a lower rate of false positives on SFD as well as a better consistency across metrics.

4.3.2 Efficiency Evaluation

We also evaluate the efficiency of the resulting models in terms of number of parameters, inference time and search cost. These metrics are of high importance considering the resource limited environment of DAS. A model with fewer parameters is generally more efficient and easier to implement, making it a crucial characteristic for driver distraction applications. Similarly, low inference time, i.e., fast response, is a desirable characteristic in such an environment. The results are reported in Table 3. It is to note that we implemented our approach using a GPU Nvidia Tesla A100 32G.

Table 3: Model computational metrics.

Dataset	Search cost ↓	Params ↓	Inference ↓
SFD	1h00	0.10 M	9 ms
AUC	1h30	0.14 M	10 ms

Firstly, we notice that the search process is remarkably efficient, completing within a notably brief timeframe of only 55min to 1.5 hours. Secondly, our resulting models are extremely lightweight with 0.10 M and 0.14 M parameters for SFD and AUC respectively. Furthermore, the inference time on the GPU is impressively fast with 9 (ms) on SFD and 10 (ms) on AUC. Similar inference is expected on in-car platforms as studies have shown that lightweight architectures usually perform equal or better on CPUs than GPUs (Li et al., 2023).

4.3.3 Comparison with State-of-the-Art

In evaluating our approach for DDD, we benchmarked against existing state-of-the-art methods. As a reminder, most of the studies using benchmark datasets present hand-crafted approaches. Only a single work by (Liu et al., 2023) uses non-differentiable NAS. Notably, our work stands as the sole contributor exploring DARTS in the context of DDD. We report state-of-the-art on SFD in Table 4 and on AUC in Table 5.

Table 4: Comparison with state of the art on SFD.

Work	Acc (%)	Params (M)
Hand-crafted		
(Dhakate and Dash, 2020)	92.90	25.60
(Baheti et al., 2020)	99.75	2.20
(Qin et al., 2021)	99.82	0.76
(Hossain et al., 2022)	98.12	3.50
(Wang et al., 2022b)	99.88	2.78
(Wang et al., 2022a)	**99.91**	9.02
(Wang and Wu, 2023)	96.82	23.67
(Mittal and Verma, 2023)	99.50	8.50
NAS		
(Liu et al., 2023)	99.87	0.42
Ours (2023)	99.08	**0.10**

Table 5: Comparison with state of the art on AUC.

Work	Acc (%)	Params (M)
Hand-crafted		
(Eraqi et al., 2019)	94.29	62.00
(Ai et al., 2019)	87.74	140
(Baheti et al., 2020)	95.24	2.20
(Qin et al., 2021)	95.64	0.76
(Mittal and Verma, 2023)	95.59	8.50
NAS		
(Liu et al., 2023)	**96.78**	0.42
Ours (2023)	93.23	**0.14**

We first assess hand-crafted methodologies on SFD where traditional CNNs such as VGG16 (Dhakate and Dash, 2020) and Capsule Networks (Mittal and Verma, 2023) are employed. Our approach achieves a comparable accuracy of 99.08% while having x7 times fewer parameters than the best performing study by (Qin et al., 2021) in terms of efficiency. Additionally, the comparison includes well-established architectures like MobileNet (Hossain et al., 2022) which is surpassed by our model both in terms of accuracy and efficiency. Similarly, despite the challenges posed by the AUC dataset, our model showcases competitive performance, achieving an accuracy of 93.23% while reducing the number of parameters to 0.14 M.

In the NAS category, our approach establishes its efficiency by achieving approximately x3 fewer parameters than the model reported by (Liu et al., 2023). This emphasizes not only the accuracy but also the computational efficiency of our method in comparison to the NAS counterpart.

Overall, our approach, guided by the innovative application of DARTS, not only outperforms some hand-crafted methodologies but also demonstrates efficiency gains, thereby contributing significantly to the evolving landscape of DDD methodologies.

5 CONCLUSIONS

In this paper, we proposed an efficient Driver Distraction Detection with DARTS-optimized lightweight models. We perform for the first time Differentiable Architecture Searches to automatically find accurate yet efficient models for a real-world application. We discuss the challenges of hand-designed models and the motivation behind NAS. We demonstrate that the obtained models are extremely lightweight with high classification performance compared to the state-of-the-art. The efficiency gains, evidenced by the reduction in the number of parameters by almost threefold compared to state-of-the-art models, further emphasize the practical viability of our approach. Future works include a broader investigation of multiple hardware as well as a more efficient search strategy.

ACKNOWLEDGEMENTS

This work was supported by the CHIST-ERA grant SAMBAS (CHIST-ERA20-SICT-003), with funding from FWO, ANR, NKFIH, and UKRI.

REFERENCES

Ai, Y., Xia, J., She, K., and Long, Q. (2019). Double attention convolutional neural network for driver action recognition. In *(EITCE)*, pages 1515–1519.

Anna Montoya, Dan Holman, S. T. S. W. K. (2016). State farm distracted driver detection.

Baheti, B., Talbar, S., and Gajre, S. (2020). Towards computationally efficient and realtime distracted driver detection with mobilevgg network. *IEEE Transactions on Intelligent Vehicles*, 5(4):565–574.

Chen, J., Jiang, Y., Huang, Z., Guo, X., Wu, B., Sun, L., and Wu, T. (2021). Fine-grained detection of driver distraction based on neural architecture search. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):5783–5801.

Chen, J., Li, H., Han, L., Wu, J., Azam, A., and Zhang, Z. (2022). Driver vigilance detection for high-speed rail using fusion of multiple physiological signals and deep learning. *Applied Soft Computing*, 123:108982.

Das, K., Papakostas, M., Riani, K., Gasiorowski, A., Abouelenien, M., Burzo, M., and Mihalcea, R. (2022). Detection and recognition of driver distraction using multimodal signals. *ACM Transactions on Interactive Intelligent Systems*, 12(4):1–28.

Dhakate, K. R. and Dash, R. (2020). Distracted driver detection using stacking ensemble. In *(SCEECS)*, pages 1–5. IEEE.

Dolezalek, E., Farnan, M., and Min, C.-H. (2021). Physiological signal monitoring system to analyze driver attentiveness. In *(MWSCAS)*, pages 635–638. IEEE.

Eraqi, H. M., Abouelnaga, Y., Saad, M. H., and Moustafa, M. N. (2019). Driver distraction identification with an ensemble of convolutional neural networks. *Journal of Advanced Transportation*, 2019.

Fernández, A., Usamentiaga, R., Carús, J. L., and Casado, R. (2016). Driver distraction using visual-based sensors and algorithms. *Sensors*, 16(11):1805.

for Statistics, N. C. and Analysis (2023). Distracted driving in 2021. Research Note DOT HS 813 443, National Highway Traffic Safety Administration.

Hossain, M. U., Rahman, M. A., Islam, M. M., Akhter, A., Uddin, M. A., and Paul, B. K. (2022). Automatic driver distraction detection using deep convolutional neural networks. *Intelligent Systems with Applications*, 14:200075.

Huang, T. and Fu, R. (2022). Driver distraction detection based on the true driver's focus of attention. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):19374–19386.

Kang, J.-S., Kang, J., Kim, J.-J., Jeon, K.-W., Chung, H.-J., and Park, B.-H. (2023). Neural architecture search survey: A computer vision perspective. *Sensors*, 23(3):1713.

Koay, H. V., Chuah, J. H., Chow, C.-O., and Chang, Y.-L. (2022). Detecting and recognizing driver distraction through various data modality using machine learning: A review, recent advances, simplified framework and open challenges (2014–2021). *Engineering Applications of Artificial Intelligence*, 115:105309.

Lee, J. D. (2005). Driving safety. *Reviews of human factors and ergonomics*, 1(1):172–218.

Li, H., Wang, Z., Yue, X., Wang, W., Tomiyama, H., and Meng, L. (2023). An architecture-level analysis on deep learning models for low-impact computations. *Artificial Intelligence Review*, 56(3):1971–2010.

Li, W., Huang, J., Xie, G., Karray, F., and Li, R. (2021). A survey on vision-based driver distraction analysis. *Journal of Systems Architecture*, 121:102319.

Liu, D., Yamasaki, T., Wang, Y., Mase, K., and Kato, J. (2023). Toward extremely lightweight distracted driver recognition with distillation-based neural architecture search and knowledge transfer. *IEEE Transactions on Intelligent Transportation Systems*.

Liu, H., Simonyan, K., and Yang, Y. (2019). Darts: Differentiable architecture search.

McCall, J. and Trivedi, M. M. (2006). Driver monitoring for a human-centered driver assistance system. In *Proceedings of the 1st ACM international workshop on Human-centered multimedia*, pages 115–122.

Mittal, H. and Verma, B. (2023). Cat-capsnet: A convolutional and attention based capsule network to detect the driver's distraction. *IEEE Transactions on Intelligent Transportation Systems*.

Nareshkumar, R., Suseela, G., Nimala, K., and Niranjana, G. (2023). Feasibility and necessity of affective computing in emotion sensing of drivers for improved road safety. In *Principles and Applications of Socio-Cognitive and Affective Computing*, pages 94–115. IGI Global.

Nidamanuri, J., Mukherjee, P., Assfalg, R., and Venkataraman, H. (2022). Dual-v-sense-net (dvn): Multisensor recommendation engine for distraction analysis and chaotic driving conditions. *IEEE Sensors Journal*, 22(15):15353–15364.

Qin, B., Qian, J., Xin, Y., Liu, B., and Dong, Y. (2021). Distracted driver detection based on a cnn with decreasing filter size. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6922–6933.

Seong, J., Lee, C., and Han, D. S. (2022). Neural architecture search for real-time driver behavior recognition. In *(ICAIIC)*, pages 104–108. IEEE.

Shahverdy, M., Fathy, M., Berangi, R., and Sabokrou, M. (2020). Driver behavior detection and classification using deep convolutional neural networks. *Expert Systems with Applications*, 149:113240.

Wang, H., Chen, J., Huang, Z., Li, B., Lv, J., Xi, J., Wu, B., Zhang, J., and Wu, Z. (2022a). Fpt: fine-grained detection of driver distraction based on the feature pyramid vision transformer. *IEEE transactions on intelligent transportation systems*, 24(2):1594–1608.

Wang, J. and Wu, Z. (2023). Driver distraction detection via multi-scale domain adaptation network. *IET Intelligent Transport Systems*.

Wang, J., Wu, Z., et al. (2022b). Model lightweighting for real-time distraction detection on resource-limited devices. *Computational Intelligence and Neuroscience*, 2022.

Xing, Y., Lv, C., Cao, D., and Hang, P. (2021). Toward human-vehicle collaboration: Review and perspectives on human-centered collaborative automated driving. *Transportation research part C: emerging technologies*, 128:103199.

Zaman, K., Sun, Z., Shah, S. M., Shoaib, M., Pei, L., and Hussain, A. (2022). Driver emotions recognition based on improved faster r-cnn and neural architectural search network. *Symmetry*, 14(4):687.

Zeng, X., Wang, F., Wang, B., Wu, C., Liu, K. J. R., and Au, O. C. (2022). In-vehicle sensing for smart cars. *IEEE Open Journal of Vehicular Technology*, 3:221–242.

Zepf, S., Hernandez, J., Schmitt, A., Minker, W., and Picard, R. W. (2020). Driver emotion recognition for intelligent vehicles: A survey. *ACM Computing Surveys (CSUR)*, 53(3):1–30.

Zhu, Z., Wang, S., Gu, S., Li, Y., Li, J., Shuai, L., and Qi, G. (2023). Driver distraction detection based on lightweight networks and tiny object detection. *Mathematical Biosciences and Engineering*, 20(10):18248–18266.