Do You Act Like You Talk? Exploring Pose-based Driver Action Classification with Speech Recognition Networks

Pablo Pardo-Decimavilla[1] Luis M. Bergasa[1] Santiago Montiel-Marín[1] Miguel Antunes[1] Ángel Llamazares[1]

***Abstract*— Recognizing distractions on the road is crucial to reduce traffic accidents. Video-based networks are typically used, but are limited by their computational cost and are vulnerable to viewpoint changes. In this paper, we propose a novel approach for pose-based driver action classification using speech recognition networks, which is lighter and more viewpoint invariant that video-based one. We leverage the similarity in the encoding of information between audio and pose data, representing poses as key points over time. Our architecture is based on Squeezeformer, an efficient attention-based speech recognition network. We introduce a selection of data augmentation techniques to enhance generalization. Experiments on the Drive&Act dataset demonstrate superior performance compared to state-of-the-art methods. Additionally, we explore the integration of object information and the impact of viewpoint changes. Our results highlight the effectiveness and robustness of speech recognition networks in pose-based action classification.**

I. Introduction

Every day in the United States, nine people lose their lives in accidents that are reported to involve a distracted driver [1]. In Europe, between 5-25% of all crashes are due to lack of attention while driving [2]. Actions such as talking on the phone, eating or operating the multimedia screen distract the driver and slow down his reaction time. The implementation of systems that identify and notify these actions has the potential to reduce the occurrence of accidents.

Driver action classification networks are derived from Human Action Recognition (HAR) architectures. Both can be broadly classified into those based on RGB images and those based on human pose. Other references in the literature combine both modalities [3]. Image-based networks are able to capture fine details of the video and extract relevant features from each class [4] [5]. Pose-based architectures extract the 2D or 3D pose of the driver for further processing with a neural network [6]. These last are computationally lighter than vision-based ones and they are agnostic to viewpoint, background or lighting changes, resulting in interesting networks for automation purposes [7].

This paper explores the use of speech recognition networks for pose-based driver distractions classification. The relationship between these two concepts arises from the similarity in the encoding of information. As seen in the Figure 1, audio is encoded in a spectrogram as a number of features (frequencies) sampled over time, while the pose can be represented in the same way, with those features being keypoints of the driver. Our interest lies in audio recognition encoders that can identify temporal patterns and relationships through complex sequences.

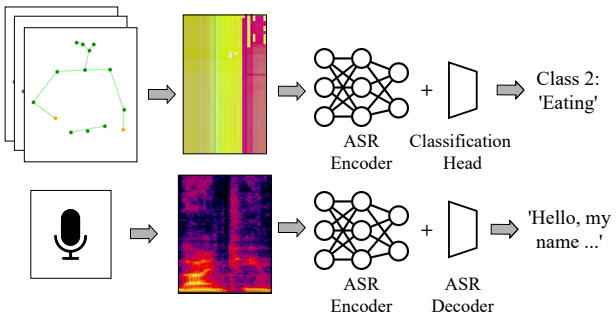

Fig. 1. Graphical abstract.

Our architecture comprises three sections. First, a stem layer adjusts the size of the input to be processed by the encoder. We use a Transformer-based encoder inspired by Squeezeformer [8], a novel and efficient architecture imported from the audio recognition domain. In contrast to speech networks, we aim to classify actions, therefore we use a classification head as decoder.

We propose a selection of data augmentation techniques to enhance the network's generalisation and avoid overfitting during training. Three sets of techniques are proposed, one focusing on the temporal dimension, one on the spatial dimension and one combining both. We train on Drive&Act [9], a challenging dataset for driver distraction classification. Our experiments show superior performance compared to other state-of-the-art techniques. The main contributions are:

- Development of an architecture based on speech recognition networks for pose-based driver activity classification.
- Development of data augmentation techniques focusing on spatial and temporal features. Through an ablation study, we demonstrate the contribution of each group of techniques.
- Validation of the network in the Drive&Act dataset over-performing the state-of-the-art.
- Inclusion experiment of information about objects of interest in the scene, which resulted in significant im-

*This work has been supported by the Spanish PID2021-126623OB-I00 project, funded by MICIN/AEI and FEDER, TED2021-130131A-I00, PDC2022-133470-I00 projects from MICIN/AEI and the European Union NextGenerationEU/PRTR, PLEC2023-010343 project (INARTRANS 4.0) from MCIN/AEI/10.13039/501100011033, and ELLIS Unit Madrid funded by Autonomous Community of Madrid.

[1]P. Pardo-Decimavilla, L.M. Bergasa, S. Montiel-Marín, M. Antunes and Ángel Llamazares are with the Electronics Departament, University of Alcalá (UAH), Spain.{pablo.pardod, luism.bergasa, santiago.montiel, miguel.antunes, angel.llamazares}@uah.es

provements.
- Robustness study of our network to viewpoint changes compared to video-based networks.

In summary, this paper proposes an architecture based on speech recognition networks that, together with a selection of data augmentations, provides results that outperform the state of the art. Our code is publicly available[1].

II. Related Works

A. Pose-based Driver Action Recognition

Over the past few years, there has been a growing interest in pose-based action recognition, particularly in its application to autonomous vehicles. The aim of this task is to classify driver videos into predetermined categories. The actions of interest can occur while performing manual driving of the vehicle or while being a passenger in an automated environment. Some studies focus on analysing facial pose to classify the driver's gaze direction [10]. Others focus on examining the complete posture of the driver. Convolutional neural networks are utilized to classify actions based on pose input [11]. Mingyan Wu et al. [12] combine spatial features obtained with a CNN with geometric features to predict the corresponding driver action. The two-stream RNN architecture proposed by [13] models both temporal and spatial dynamics. A third stream is introduced in [14] with information about the driver's environment. Advanced networks unify the study by processing the 3D skeleton in the spatial and temporal domain in one stream. Graph neural networks [15] are able to learn spatial and temporal patterns from data simultaneously. Martin et al. [16] use this approach to integrate additional input modalities like interior elements and objects. Hong Yan et al. [17] build an asymmetric graph-based encoder-decoder integrated with a spatial-temporal representation learning module. With the advent of transformers, an efficient alternative called MLP-Mixers, inspired by CNNs and transformers, is proposed. St-MLP [6] demonstrates competitive results in pose-based action classification by adapting this architecture.

B. Speech Recognition Networks

End-to-end automatic speech recognition (ASR) models typically comprise an encoder that processes a speech signal (a sequence of speech frames) and extracts high-level acoustic features. The encoder is complemented by a decoder that converts the extracted features into a text sequence. The encoder's architecture is critical in determining the model's representational capacity and its ability to extract acoustic features from input signals. Our study will focus exclusively on ASR encoders.

Convolutional neural networks (CNN) are often used as backbones [18]. The main weakness of these solutions is the inability to capture the global context. With the advent of Transformers, new models adopted this architecture to address this issue [19]. Conformer [20] combines Transformers with convolutions to take advantage of the benefits of each. Recently, Squeezeformer [8] has been introduced, which makes a deep study and solves some design problems of the Conformer resulting in improved efficiency and performance.

[1] https://github.com/pablopardod/dyalyt

III. Method

A. Problem Formulation

The input data is represented as a three-dimensional matrix. Let $X \in \mathbb{R}^{3\times F\times K}$, where the first dimension encompasses the spatial coordinates (x, y, z) for each keypoint, F denotes the number of frames and K represents the count of keypoints in the pose. Let C be the class label associated with the i-th video segment, where i denotes the segment index within the video. Formally, $C = \{C_1, C_2, \ldots, C_M\}$, where M is the total number of distinct classes. The dataset D is a collection of multiple videos, $D = \{V_1, V_2, \ldots, V_N\}$ where N represents the total number of videos. Each video V_j is partitioned into segments X_i of duration F and the associated class labels C.

Our objective is to formulate a mapping function $f : \mathbb{R}^{3\times F\times K} \rightarrow \mathbb{N}$. Given a tensor X_i, our goal is to predict the corresponding class C with the following architecture.

B. Our Approach

The aim of this study is to investigate the application of automatic speech recognition in pose-based action recognition. The encoder is based on the Squeezeformer [8] block. It is a hybrid attention-convolution architecture that builds on the Conformer architecture [20] to improve performance, making an important FLOPS reduction by solving some design problems. Considering the current state of the art and the design paradigms adopted by other solutions in the field, we propose the architecture shown in Figure 2.

The initial layer is a convolutional block, designed to conform the input data to the specifications of the encoder. The layer transforms the input's three (x,y,z) channels into the number of tokens per channels required by the encoder. The utilized kernel selectively operates on the spatial dimensions of the input data without temporal integration. This means that we will squeeze the input tensor without mixing the features in the temporal dimension. This operation guarantees that the number of frames matches the token length. Following the convolutional layer, a subsequent normalization block is incorporated, followed by the application of the activation function (SiLU). We have switched from an image representation to a token like that can be consumed by a transformer. Compared to the audio data, ours have similar characteristics but in smaller dimensions. Due to this it is not necessary to temporarily reduce the input or apply feature extraction techniques. As summary, this first section transforms the input tensor $\mathbb{R}^{3\times F\times K}$ into $\mathbb{R}^{F\times E}$, where F is the token length (number of frames) and E is the channels per token size.

Next, we continue with the Transformer encoder with multiple Squeezeformer blocks, which are repeated N times. The input and output sizes are identical ($\mathbb{R}^{F\times E}$), allowing for seamless chaining of the blocks without any intermediate

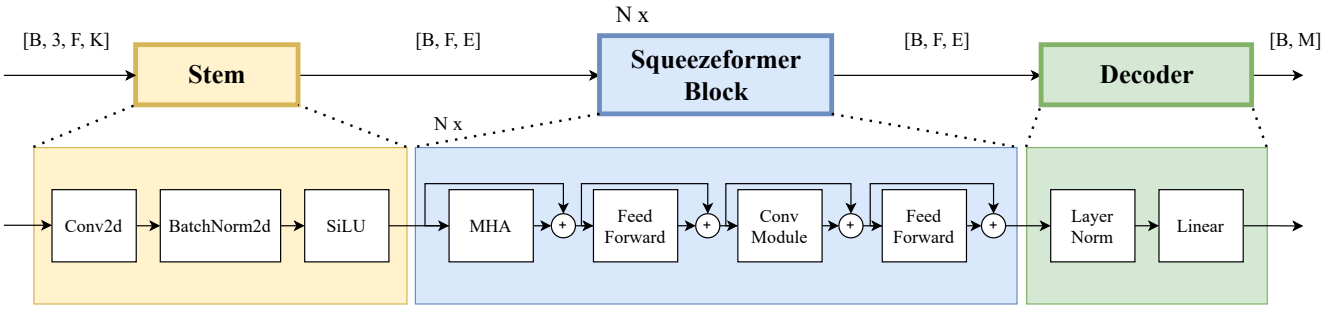

Fig. 2. Architecture overview. **B:** Batch size; **F:** Number of frames; **K:** Number of keypoints; **E:** Channels per token; **M:** Number of classes; **N:** Number of encoder blocks.

adaptation. Each Squeezeformer block comprises four modules. Following each block, the residual sum is calculated by adding the output of a module to its input. This technique prevents the problem of gradient fading and enables the network to learn identity functions. Additionally, after each sum, a post-layer normalization (PostLN) is applied to normalize the output of the modules. The first module is a multi-head attention module (MHA). This module enables the model to simultaneously focus on various parts of the input sequence, which is crucial for capturing intricate dependencies between key points. The next module is the Feed Forward Module, which applies transformations to the input data processed by the attention to enhance the model's ability to represent relationships in the data. The next module is the Convolution Module, which captures local features and helps the model understand the temporal structure of the data. The block concludes with a Feed Forward Module.

Finally, the decoder is presented. It consists of a normalization layer followed by a linear layer. This block transforms the output of the last Squeezeformer block into a vector of size $\mathbb{R}^M$, which indicates the confidence of belonging to each class.

C. Data Augmentation

Data augmentation plays a crucial role in training deep learning models. This technique involves generating diverse variations of the existing dataset by applying transformations. The primary objective of data augmentation is to enhance the model's generalization capability by exposing it to a wider range of scenarios and variations in the input data. We propose the selection and adaptation of techniques that can be divided into three categories:

- **Temporal augmentations:** Resampling functions have been implemented to adjust the sequence length by either interpolating frames or repeating them to modify the sampling frequency. A random timeshift is also applied to the sequence, along with an augmentation that randomly removes the beginning and end of the sequence. Additionally, time masks are randomly applied to hide certain time periods in the sequences.
- **Skeleton augmentations:** Based on the data augmentation methods proposed in [21], three modifications to the global driver skeleton are suggested. The height can be modified by stretching or shrinking the skeleton, the width by enlarging or narrowing it, and the rotation of the skeleton about the back axis can also be adjusted. These modifications enable the generation of diverse bodies.
- **CutMix-based augmentations:** CutMix [22] is a data augmentation technique that involves cutting and pasting square patches from different images during training to create new training samples. This technique will be applied by mixing a contiguous set of keypoints between two sequences for a certain time to form a square. Additionally, as shown in Figure 3, we propose two modifications that are adapted to our type of data:
  - **Spatial mixing:** A random set of contiguous keypoints are selected and spread over the entire sequence.
  - **Temporal mixing:** During a random time duration all keypoints will be mixed.

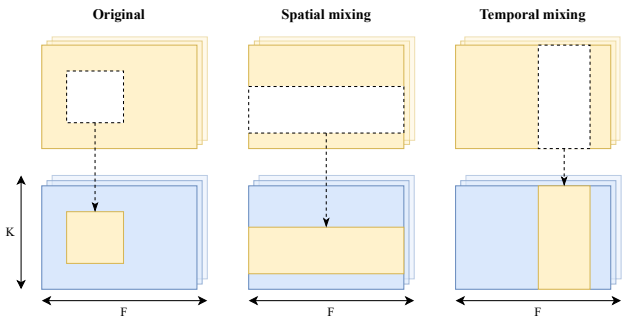

Fig. 3. Original CutMix and proposed modifications. Blue and yellow represent two different sequences that belong to the same batch. **F**: number of frames. **K**: number of keypoints.

The labels of the pasted patches are mixed based on the area ratios of the patches, providing a smooth combination of multiple classes in the augmented training samples.

IV. Experiments

A. Dataset

Drive&Act [9] is an action recognition dataset for autonomous vehicles in which 15 different people perform

actions while being recorded from 6 different points of view. It provides common distractions such as *eating* or *drinking*, but also include actions more typical of autonomous cars such as *reading a newspaper* or *working with a laptop*. It also provides the 3D skeleton of the driver that has been extracted with OpenPose [23] and triangulated with the 2D poses from three different views. Figure 4 displays four class examples along with the driver's pose. The skeleton is formed by 13 landmarks corresponding to the driver's upper body. A 3D model of the car interior is also provided.

Three simultaneous annotations are defined for all sequences:

- **Coarse scenarios/tasks**: Representing the highest level of abstraction, these encompass 12 general tasks that actors must complete during recording.
- **Fine-grained activities**: At the second level of hierarchy, these activities further break down into 34 classes. Each one is performed uniquely by drivers based on individual preferences. These concise classes distinguish actions like opening and closing a water bottle.
- **Atomic action units**: Describing the most precise level of detail, this level encompasses 5 action types, 17 object categories, and 14 car locations. This results in 372 distinct combinations or classes, explicitly associated with object interactions.

Fig. 4. Example of four frames, including the driver's pose and corresponding class.

As we are mainly concerned with the pose, the most appropriate annotation is the coarse. Then, coarse scenarios will be used for training and validation in the results section. The other annotations refer to specific object-related classes. The predefined splits, for training, validation, and testing, will be utilized and our chosen metric will be the mean per-class accuracy (macro-accuracy), established as the standard criterion. Our task will be to classify the entire sequence, already defined by the dataset, with the label corresponding to the class. Sequences are sampled at a fixed frequency for a maximum amount of time. If there are not enough samples, they are padded to zero.

B. Implementations Details

Our architecture has been implemented with PyTorch [24] and PyTorch Lightning [25]. The model has been trained for 100 epochs with a balanced batch (since the dataset is unbalanced) of 256 sequences. AdamW was used for optimization and cosine annealing was employed as the learning rate scheduler.

In the pursuit of optimizing model performance, a systematic exploration of hyperparameters was conducted employing Wandb Sweeps [26]. The network size was investigated by varying the number of encoder blocks and token channels to identify an optimal configuration. Various dropout values for the blocks were also included in the exploration. Parameters pertinent to the learning rate scheduler, such as learning rate (lr), warmup steps, and weight decay, were incorporated into the search. Detailed information regarding these parameter values can be found in the code repository. The search is performed using bayesian optimization seeking to minimize the validation loss function (cross entropy loss). Results of the best model are shown in the following section.

C. Results

We evaluate our architecture in the Drive&Act dataset [9] for the coarse modality. Fig. 5 shows the accuracy per class in the test set. Table I displays the macro-accuracy results comparing with other state-of-the-art architectures. The data column indicates the input used in the network. 'P' represents the driver pose, 'I' represents the 3D model of the vehicle interior and 'O' the objects. Our encoder consists of two Squeezeformer blocks with 120 channels per token.

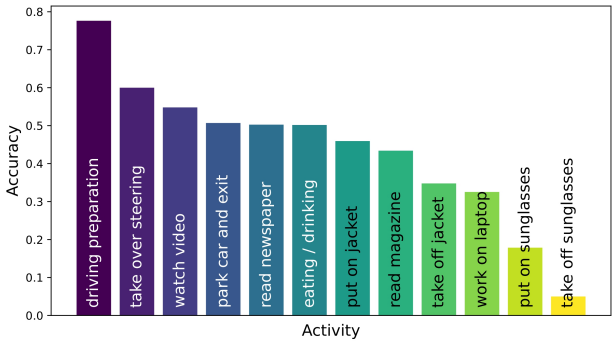

Fig. 5. Per class accuracy of the test set of the Drive&Act dataset.

TABLE I

EVALUATION OF COARSE SCENARIOS/TASKS ON THE DRIVE&ACT DATASET USING MACRO-ACCURACY. **P:** POSE; **I:** INTERIOR; **O:** OBJECT

Data	Method	Validation	Test
-	Random	8.33	8.33
P + I	Interior [9]	35.76	29.75
P + I	Three-Stream [14]	41.70	35.45
P + I + O	GNN [16]	42.82	37.84
P	Pose [9]	37.18	32.96
P	Two-Stream [13]	39.37	34.81
P	st-MLP [6]	40.56	34.61
P	**ours**	**44.60** (+4.04)	**43.59** (+8.98)

The architecture demonstrates superior performance compared to other proposals, achieving better results in both validation and test. Performance is superior even when compared to networks that use extra data (I,O). The major improvement lies in the test suite. The values obtained are more balanced, demonstrating a higher generalisation capability. When compared to st-MLP [6], a solution also based on attention, we improved results by 4.04 in validation and 8.98 in the test set. The number of parameters in our network is 538k, while st-MLP has 588k, resulting in a reduction of 50k parameters.

D. Ablation Study

1) Data augmentation: Data augmentation helps to improve the generalisation and robustness of data, particularly when the dataset is small. This leads to a smaller difference between the results of the validation and test sets. An ablation study has been proposed to test the contribution of these techniques to the final result. The study uses the configuration outlined in the result section and is conducted with the four data augmentation groups described above. The individual contribution of each group of used techniques are depicted in Table II.

TABLE II
ABLATION STUDY IN DATA AUGMENTATION. **O:** ORIGINAL; **S:** SPATIAL; **T:** TEMPORAL

Data augmentation techniques						
Temporal	**Skeleton**	**CutMix-based**			**Val**	**Test**
		O	**S**	**T**		
-	-	-	-	-	33.44	30.88
✓	-	-	-	-	38.17	33.25
-	✓	-	-	-	36.17	28.62
-	-	✓	-	-	34.61	32.68
-	-	-	✓	-	35.21	35.28
-	-	-	-	✓	35.70	34.79
-	-	✓	✓	✓	39.42	37.11
✓	✓	✓	✓	✓	**44.60**	**43.59**

We can conclude that the utilization and combination of data augmentation techniques improves the performance of our network. The test results benefit the most, with a greater improvement compared to the model without data augmentation. Among all the techniques, the CutMix-based technique stands out. Each contributes to enhancing the baseline, but integrating them provides a more significant improvement.

2) Objects implementation: We observed that classes that involve the interaction with specific objects, particularly sunglasses, exhibit a poorer performance. Previous studies [27] demonstrated that combining these objects with the pose improves the results of related classes. Martin et al. [16] combine the pose information with manually annotated objects, resulting in a 7.54 improvement over the pose-only model. In order to validate this hypothesis we propose including the objects as a keypoints. This means that dimension K will be increased by O, which represents the number of detectable objects in the scene.

The dataset provides annotations for the objects that the driver interacts with. This information is used to conduct our experiment. Information regarding the object is encoded as previously described. Consequently, the network is trained to classify actions based on the pose and objects information. The experiment results for fine-grained annotations using the macro-accuracy metric are presented in Table III.

TABLE III
EVALUATION FOR FINE-GRAINED ACTIVITIES ON THE DRIVE&ACT DATASET USING MACRO-ACCURACY.

Type	**Method**	**Validation**	**Test**
Video	P3D ResNet [28]	55.04	45.32
Video	I3D Net [29]	**69.57**	63.64
Pose + Objects	Graphs [16]	65.87	58.8
Pose + Objects	ours	66.21	**64.83**

Significant improvement is achieved through the inclusion of objects, reaching comparable results to state-of-the-art video networks based on transformers. We require additional processing for pose extraction and object detection. It is important to note that results are based on perfect annotations. Overall, this experiment shows the potential of our proposed architecture including additional information, such as objects, with pose to improve classification performance at the cost of introducing a higher computational load, similar to the vision-based approaches.

3) Robustness in viewpoint change: Achieving a robust network that handles changes in viewpoint minimise data collection efforts and facilitate data reuse. This issue is important in the automotive field since multiple cameras have already been introduced in modern cars, but their mounting position can change between vehicle models. Manuel Martin et al. [7] measured the behaviour of their network in response to changes in viewpoints. We replicate their experiment to assess the robustness of our architecture. The 3D pose is extracted from the triangulation of three frontal views (right-top, front-top, left-top). The dataset includes a Kinect camera with depth estimation located at the top-left position. The 3D pose is extracted combining OpenPose 2D pose along with the depth image. A cross-view test is conducted by training in one modality and evaluating in the opposite one.

Results are compared with I3D, a video-based end-to-end model. Evaluation is conducted using both the central mirror and the Kinect camera. Results of I3D are taken from [7]. Table IV displays the cross-view experiment results for both I3D and our pose-based architecture in the fine-grained activities of the Drive&Act dataset.

The performance of the video-based network is significantly impacted by changes in camera perspective, resulting in a 90% reduction. In contrast, our proposal exhibits greater robustness to sensor changes, with an average reduction of 22.79%.

TABLE IV

CROSS-VIEW EVALUTION FOR I3D AND THE PROPOSED ARCHITECTURE ON THE VALIDATION SET OF FINE-GRAINED ACTIVITES OF DRIVE&ACT. **CM:** CENTER MIRROR; **KIR:** KINECT IR

Train	**Test**			
	I3D Net		**Ours**	
	CM	KIR	CM	KIR
CM	69.6	6.8	52.88	35.58
KIR	6.7	72.9	43.77	49.67
Variation	-90.37%	-90.67%	-17.22%	-28.36%

V. CONCLUSION

This paper presents a novel approach to driver action classification based on 3D pose information, leveraging insights from the field of speech recognition. The proposed architecture combines a Squeezeformer-based encoder with data augmentation techniques to enhance the model's generalization. The experiments conducted on the Drive&Act dataset demonstrate the effectiveness of the proposed architecture, outperforming existing approaches.

An ablation study is conducted to measure the importance of each group of data augmentation techniques. Based on previous studies an experiment explores the impact of incorporating object information. Results indicate that our architecture is able to establish relationships and improve the baseline. Finally, an experiment demonstrates the robustness of our architecture to changes in viewpoint when compared to video-based networks.

In future work we plan to further explore the object fusion incorporating a suitable object detector, as well as to incorporate more information such as the driver's gaze or the car interior.

REFERENCES

[1] C. for Disease Control and Prevention, "Distracted driving," https://www.cdc.gov/transportationsafety/distracted_driving/index.html, (Accessed on 01/17/2024).

[2] D. G. for Transport, *Road safety thematic report – Driver distraction*. European Commission, 2022, vol. European Road Safety Observatory.

[3] F. Guo, T. Jin, S. Zhu, X. Xi, W. Wang, Q. Meng, W. Song, and J. Zhu, "B2c-afm: Bi-directional co-temporal and cross-spatial attention fusion model for human action recognition," *IEEE Transactions on Image Processing*, vol. 32, pp. 4989–5003, 2023.

[4] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," 2019.

[5] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," 2021.

[6] A. Holzbock, A. Tsaregorodtsev, Y. Dawoud, K. Dietmayer, and V. Belagiannis, "A spatio-temporal multilayer perceptron for gesture recognition," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1109/IV51971.2022.9827054

[7] M. Martin, D. Lerch, and M. Voit, "Viewpoint invariant 3d driver body pose-based activity recognition," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–6.

[8] S. Kim, A. Gholami, A. Shaw, N. Lee, K. Mangalam, J. Malik, M. W. Mahoney, and K. Keutzer, "Squeezeformer: An efficient transformer for automatic speech recognition," 2022.

[9] M. Martin, A. Roitberg, M. Haurilet, M. Horne, S. Reiß, M. Voit, and R. Stiefelhagen, "Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2801–2810, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:201624289

[10] C. Hari and P. Sankaran, "Driver distraction analysis using face pose cues," *Expert Systems with Applications*, vol. 179, p. 115036, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421004772

[11] B. Li, Y. Dai, X. Cheng, H. Chen, Y. Lin, and M. He, "Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn," in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2017, pp. 601–604.

[12] M. Wu, X. Zhang, L. Shen, and H. Yu, "Pose-aware multi-feature fusion network for driver distraction recognition," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 1228–1235.

[13] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," 2017.

[14] M. Martin, J. Popp, M. Anneken, M. Voit, and R. Stiefelhagen, "Body pose and context information for driver secondary task detection," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 2015–2021.

[15] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," 2018.

[16] M. Martin, M. Voit, and R. Stiefelhagen, "Dynamic interaction graphs for driver activity recognition," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7.

[17] H. Yan, Y. Liu, Y. Wei, Z. Li, G. Li, and L. Lin, "Skeletonmae: Graph-based masked autoencoder for skeleton sequence pre-training," 2023.

[18] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," 2019.

[19] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, Dec. 2019. [Online]. Available: http://dx.doi.org/10.1109/ASRU46091.2019.9003750

[20] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020.

[21] J. Chen, W. Yang, C. Liu, and L. Yao, "A data augmentation method for skeleton-based action recognition with relative features," *Applied Sciences*, vol. 11, no. 23, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/23/11481

[22] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," 2019.

[23] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," 2019.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[25] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," Mar. 2019. [Online]. Available: https://github.com/Lightning-AI/lightning

[26] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/

[27] P. Weyers, D. Schiebener, and A. Kummert, "Action and object interaction recognition for driver activity classification," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 4336–4341.

[28] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," 2017.

[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," 2015.