IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 August 2022, accepted 19 September 2022, date of publication 30 September 2022, date of current version 7 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3210973

RESEARCH ARTICLE

Toward Vehicle Occupant-Invariant Models for Activity Characterization

LEONARDO CAPOZZI[1,2], VÍTOR BARBOSA[1,2], CAROLINA PINTO[3], JOÃO RIBEIRO PINTO[1,2], (Student Member, IEEE), AMÉRICO PEREIRA[1,2], PEDRO M. CARVALHO[1,4], (Senior Member, IEEE), AND JAIME S. CARDOSO[1,2], (Senior Member, IEEE)

[1]Centre for Telecommunications and Multimedia, INESC TEC, 4200-465 Porto, Portugal
[2]Faculdade de Engenharia, Universidade do Porto, 4099-002 Porto, Portugal
[3]Bosch Car Multimedia, 31139 Hildesheim, Portugal
[4]School of Engineering, Polytechnic of Porto, 4149 Porto, Portugal

Corresponding author: Leonardo Capozzi (leonardo.g.capozzi@inesctec.pt)

This work was supported in part by the National Funds through the Portuguese Funding Agency through FCT—Fundação para a Ciência e a Tecnologia under Project UIDB/50014/2020; and in part by Ph.D. under Grant 2021.06945.BD, Grant SFRH/BD/146400/2019, and Grant SFRH/BD/137720/2018.

ABSTRACT With the advent of self-driving cars and the push by large companies into fully driverless transportation services, monitoring passenger behaviour in vehicles is becoming increasingly important for several reasons, such as ensuring safety and comfort. Although several human action recognition (HAR) methods have been proposed, developing a true HAR system remains a very challenging task. If the dataset used to train a model contains a small number of actors, the model can become biased towards these actors and their unique characteristics. This can cause the model to generalise poorly when confronted with new actors performing the same actions. This limitation is particularly acute when developing models to characterise the activities of vehicle occupants, for which data sets are short and scarce. In this study, we describe and evaluate three different methods that aim to address this actor bias and assess their performance in detecting in-vehicle violence. These methods work by removing specific information about the actor from the model's features during training or by using data that is independent of the actor, such as information about body posture. The experimental results show improvements over the baseline model when evaluated with real data. On the Hanau03 Vito dataset, the accuracy improved from 65.33% to 69.41%. On the Sunnyvale dataset, the accuracy improved from 82.81% to 86.62%.

INDEX TERMS Autonomous vehicles, computer vision, deep learning, domain generalization.

I. INTRODUCTION

Successfully capturing passenger activity has far-reaching implications for both the user experience and safety features in autonomous vehicles (AVs). Without a driver responsible for the safety and integrity of the vehicle and its occupants, it is incumbent upon automated detection systems to monitor occupant well-being and actions and to detect potentially harmful behaviour or even violence. However, the multitude of possible actions that can be represented, the variability in how different individuals represent the same actions, the heterogeneity of sensors and types of information collected, and the influence of external factors still pose significant challenges to this task [1]. The current state of the art in action recognition is based on deep models, but their use for vehicle occupant action recognition is not without problems.

Training deep learning models requires significant amounts of data, which escalates with the complexity of the models to avoid overfitting. Moreover, the available datasets are usually split, with one part used for training and another part used for subsequent testing. Dataset availability and size can be critical when dealing with individuals engaged in different activities, both for legal reasons and because of the effort involved in preparation. If the dataset used to train a

The associate editor coordinating the review of this manuscript and approving it for publication was Khoa Luu.

model contains only a small number of actors, the model can be biased toward them, i.e., to specific individuals and their unique characteristics [2]. This causes the model to perform poorly when confronted with a different set of actors performing the same activities. In the context of this study, an actor is defined as a specific group of people in the vehicle (rather than a specific person). This means that different actors may have some individuals in common. In this work, we focus on the detection of violence and non-violence in the vehicle and describe and evaluate three methods that aim to counteract actor bias by removing actor-specific information from the features of the model, or by providing the model with data that is independent of the actors, such as information about posture, which contributes to a more robust model.

The presented research focuses on the monitoring of occupants of autonomous vehicles, more specifically on the detection of violence and non-violence. We believe that this scenario is particularly relevant since: (1) there is a strong focus on people with high heterogeneity; (2) to our knowledge, there are no available datasets for this scenario, leading to a high dependence of the models on the actors. Our main contributions are:

- Using a dataset that is domain-specific to the target scenario;
- The application of domain generalization methodologies;
- The application of regularization techniques for the specific scenario of the vehicle interior;
- The use of body posture to diversify data sources and consequently reduce actor bias.

In addition to the introduction, this paper contains 5 other sections. Section II presents the current state of the art for this problem. In section III the methods used in this study are reviewed. In section IV the data and experimental settings used in this work are presented. In section V we analyse the results. Section VI gives the conclusions from this work and presents ideas for future work.

II. RELATED WORK

A. VIDEO ACTION RECOGNITION

Action classification requires models that focus on modelling spatio-temporal information. Early attempts at action recognition used compact video descriptors to extract handcrafted spatio-temporal features [3], [4], [5]. In recent years, the use of these techniques has declined, and deep learning-based methods have pushed the boundaries of the state-of-the-art in action recognition. However, despite their success, these models have raised concerns about their bias towards specific domain characteristics. This problem arises because deep learning methods rely on training data. This dependency leads to the developed models being biased towards, for example, actor features when the dataset used lacks diversity.

One of the most successful deep learning approaches is the two-stream model [6]. In these models, there are two separate deep Convolutional Networks (ConvNets) for processing RGB frames and optical flow, which are then combined by late fusion [7]. Based on this model, several approaches to action recognition have been proposed [8], [9], [10], [11]. Another approach that is proving successful is the use of 3D Convolution Neural Networks (CNNs). These networks behave similarly to 2D CNNs, but use 3D convolutions to extract features in spatial and temporal dimensions [12]. The C3D [13] network is an example of this type of approach, where the spatio-temporal features are learned to use 3D ConvNets trained on large-scale supervised video datasets. The use of large-scale video datasets was possible because C3D can handle video frames as input without any preprocessing. Other variants based on 3D CNNs have been proposed [14], [15]. One worth mentioning is an architecture that combines the two-stream model with 3D ConvNet, called I3D [15], which served as the basis for the model created in this work and is still one of the best performing architectures for action recognition. One of the problems introduced by the use of 3D convolutions is the high computational cost. To solve this problem, several works have proposed to treat the spatial and temporal dimensions differently. Some proposed the decomposition of 3D convolutions into 2D spatial convolutions and 1D temporal convolutions, such as S3D [16], P3D [17], R(2+1)D [18]. Furthermore, SlowFast [19] shows that space and time should not be treated symmetrically. Therefore, it introduces a two-path structure to handle slow and fast motion separately.

The models mentioned so far use random spatial crops of video frames as inputs during training. Since they do not explicitly focus on the human body, it is easy to overfit the scenes and objects in the video. Therefore, skeleton data was used to focus the action recognition on the human body. This has the advantage of being lightweight and free of scene cues. As for pose-based methods for action recognition, the main differences are the use of CNNs or Graph Convolution Networks (GCN). CNN-based [20], [21], [22], [23] methods represent the skeleton with a pseudo-image and thus recognize actions in the same way as image classification. Nevertheless, skeleton data is essentially a graph in non-Euclidean space with skeleton joints as vertices and bones as edges. Therefore, GCN-based [24], [25], [26] methods were proposed to capture joint interactions on the skeleton graphs, explicitly considering the adjacent relationship between joints in the non-Euclidean space.

B. REPRESENTATION BIASES

Representation bias is a problem in various image and video classification problems, and studies have been conducted to analyse and mitigate it. Li *et al.* mitigated scene, object and people biases by re-sampling the original video datasets [27], [28]. The re-sampling approach reduces representation bias, but it also reduces the number of training data, which is not desirable for deep learning methods. Another proposed method was to use adversarial losses for different scene types to mitigate scene biases [29]. Similarly, adversarial learning procedures allowed learning signer-invariant latent representations to be highly discriminating for sign recognition [2].

The primary source of representation bias is the dataset used for model training. Some studies have shown that video datasets for action recognition exhibit biases toward objects, scenes, and people [27], [28], [29], [30]. Video datasets for action recognition are mainly divided into generic and fine-grained action recognition datasets. Generic action recognition datasets provide the generic action recognition task, which attempts to classify various action categories in various domains. Such datasets include videos from different domains, such as daily activities, sports, and entertainment. Due to the wider diversity of domains in these datasets, the models trained on them can recognise actions indirectly, i.e., by the presence of an object or a person. The use of such biased models leads to degenerate transferability and incorrectly recognizes novel actions in the same static cues. Popular datasets for generic action recognition are Kinetics [31], Moments in Time [32], ActivityNet [33], UCF-101 [34], and HMDB-51 [35]. Fine-grained action recognition datasets have been recently released, providing videos in a specific domain. Something-Something [36] includes the videos of fine-grained actions of human-object interactions. Jester [37] is a dataset for hand gesture recognition. Diving48 [27] is a video dataset in sports. These datasets overcome some problems associated with representation biases since the same domain's static cues, such as objects and scenes, are similar. Therefore, it is not easy to recognise actions based on only static cues. However, these datasets might face biases in domain static cues related to people. This happens especially in datasets with a small set of actors. In this study, the datasets used are fine-grained action recognition datasets. Nevertheless, the number of actors present is quite small. Therefore, several techniques were implemented to improve the generalisation capability regarding the actors' characteristics.

C. IN-VEHICLE OCCUPANT ACTIVITY RECOGNITION

In-vehicle activity recognition is still relatively unexplored. Some proposals addressing this topic demonstrated that activity recognition might be a possible approach for monitoring the vehicle's occupants. Some proposals focused on violence detection to monitor occupants through anomaly detection of occupant's behaviour [38] or recognition of occupant's interactions [39]. Beyond detecting violence, the recognition of occupant's activity using different modalities has also been explored, specifically using audiovisual features [1]. Audio features appeared from applying this type of feature for the classification of group emotion [40]. This work focused primarily on the available hardware and energy consumption constraints associated with implementing an activity recognition system in a vehicle. Using an audio module and a cascading strategy demonstrated reductions in memory requirements and computational demands. This reduction was most significant when the audio module was the first processing block. This research paper continues these previous studies, focusing on a hypothesis previously identified of the possibility of bias in the results obtained so far. This hypothesis arose because the past studies used small datasets with a small set of actors.

III. METHODOLOGIES

In this section, the baseline method used in this work and the methods adapted to promote actor generalization are reviewed. The goal of the latter is to accurately predict violence and non-violence in videos of vehicle occupants, regardless of the actors present. Each of these methods aims to achieve this goal in different ways.

The Adversarial Learning method [2] uses an additional network that learns to classify the different actors in the training set and later uses that knowledge to remove actor information from the features of the network, making it less biased toward the actors.

The Bilevel Learning methodology [41] assigns a different weight to each mini-batch of the training set based on how much its gradients agree with the gradients of a validation mini-batch. This reduces the actor bias by giving more weight to training mini-batches that have similar gradients to validation mini-batches, minimizing the error on the validation set and leading to better generalization capabilities.

The use of pose information is also studied, both as a substitute and as a complement to the RGB information. We calculate the pose of each person in the frame and generate the keypoint map that is then given to the network [42]. Since the keypoint map does not contain any direct information about the actors, it helps to reduce actor bias.

A. BASELINE

The baseline model used in this work is based on the model proposed in [40], presented in Figure 1. It uses RGB data for recognizing actions, and it has achieved results comparable to state-of-the-art methodologies. This model is a 3D CNN, which allows the network to take advantage and use the temporal information more efficiently [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]. The output of 3D Convolutions

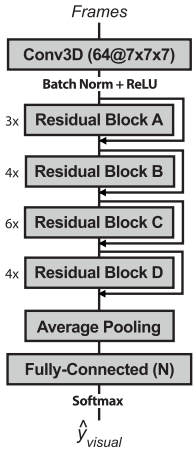


FIGURE 1. Schema of the network used for activity recognition (adapted from [40]).

is a video volume that retains the temporal information of the input, making them superior for video classification than standard 2D Convolutions, which lack the extra dimension used for the temporal component.

The model is composed of the 3D ResNet50's convolutional encoder blocks, an average pooling layer, a dropout layer, and a fully-connected layer with 2048 inputs and 2 outputs (predicting violence or non-violence). The convolutional encoder has one 3D convolutional layer with 64 filters of size $7 \times 7 \times 7$, a batch normalization layer and ReLU activation function. The following layers are three residual blocks of type A, four blocks of type B, six blocks of type C, and four blocks of type D. Residual blocks contain three convolutional layers, with a filter size of $1 \times 1 \times 1$, $3 \times 3 \times 3$, and $1 \times 1 \times 1$, respectively. The number of filters of the first two convolutional layers depends on the type of block. Type A contains 64 filters, type B contains 128 filters, type C contains 256 filters, and type D contains 512 filters. The first two convolutional layers are followed by a batch normalisation layer. The last convolutional layer of each block contains four times the number of filters of the first two layers and is followed by a batch normalisation layer and ReLU activation function. More details on the network architecture can be seen on the original ResNet paper [53].

The used network was pre-trained with videos of the Moments in Time dataset. During training, the layers of the network were frozen, except for the last 2 layers. Class weights were added to the cross-entropy loss function to combat class imbalances. The Adam optimizer was used with a learning rate of 1×10^{-4}. The work was developed using PyTorch. In all the experiments, we used the balanced accuracy to combat any class imbalances in the datasets.

B. ADVERSARIAL LEARNING

To accurately predict violence and non-violence in video, independently of the actors that are present, the model should be trained in a way where the latent representations learned by the model preserve the information relative to the action, and discard the information relative to the actors, which may negatively impact the classification. To accomplish this, the methodology proposed in [2] was adapted. It presents a network architecture and an adversarial training objective, which addresses the signer-independent problem (actor-independent problem, in our case). The model consists of a feature extractor, which maps input videos to latent representations, and two classifiers. In our case, the network architecture is the same as our baseline model. The feature extractor is the inflated ResNet-50 network, and the two classifiers are dense layers. The action-classifier predicts violence or non-violence, and the actor-classifier predicts the actor present in the video.

During the learning process, the feature extractor is simultaneously trained to help the action-classifier while trying to fool the actor-classifier. Figure 2 shows an overview of the network architecture and the loss functions.

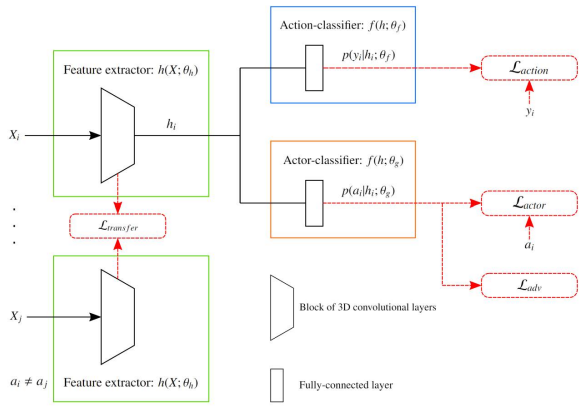


FIGURE 2. Architecture of the adversarial learning methodology (adapted from [2]).

Let $X = \{X_i, y_i, a_i\}_{i=1}^{N}$ represent a labeled dataset with N samples, where X_i represents the i-th video sample (a set of 8 concatenated RGB frames), y_i represents the action label (violence or non-violence), a_i represents the actor label and A represents the set of actor labels.

The feature extractor learns an encoding function $h(X;\theta_h)$, which using the parameters θ_h encodes an input video sample X into a latent representation h.

The action-classifier receives the latent representation h and learns a function $f(h;\theta_f)$, parameterized by θ_f, that gives the predicted probabilities $p(y|h;\theta_f)$ for each action class.

The actor-classifier learns a function $f(h;\theta_g)$, which using the parameters θ_g maps the latent representation h to the predicted probabilities $p(a|h;\theta_g)$ for each actor.

The actor-classifier is trained to minimize the negative log-likelihood of correct actor predictions:

$$\min_{\theta_g} \mathcal{L}_{actor}(\theta_h, \theta_g) = -\frac{1}{N}\sum_{i=1}^{N} \log p(a_i|h(X_i;\theta_h);\theta_g) \quad (1)$$

The feature extractor and the action-classifier are trained to minimize the negative log-likelihood of correct action predictions:

$$\min_{\theta_h,\theta_f} \mathcal{L}_{action}(\theta_h, \theta_f) = -\frac{1}{N}\sum_{i=1}^{N} \log p(y_i|h(X_i;\theta_h);\theta_f) \quad (2)$$

Additionally, the predictions of the actor-classifier should be close to uniform, meaning that it is not capable of doing better than random guessing the actor identity. The following loss (eq. 3) adjusts the weights of the feature extractor to make the predictions of the actor-classifier close to uniform, and it is an adversarial loss with respect to the actor classification loss $\mathcal{L}_{actor}$:

$$\min_{\theta_h} \mathcal{L}_{adv}(\theta_h, \theta_g) = -\frac{1}{N|A|}\sum_{i=1}^{N}\sum_{a\in A} \log p(a|h(X_i;\theta_h);\theta_g) \quad (3)$$

In order to further encourage the actor invariance properties of the latent representation h, another loss $\mathcal{L}_{transfer}$ was added. It minimizes the distance between the hidden latent representations of different actors at each layer of the feature extraction network. The distance $D^{(m)}$ between the latent representations $h^{(m)}(\bullet;\theta_h)$ of actors a and t, at the m-th layer is calculated as follows:

$$D^{(m)}(a,t;\theta_h) = \left\| \frac{1}{N_a}\sum_{i:a_i=a} h^{(m)}(X_i;\theta_h) - \frac{1}{N_t}\sum_{j:a_j=t} h^{(m)}(X_j;\theta_h) \right\|_2^2, \quad (4)$$

where $||\bullet||_2$ is the $l2$-norm, and N_a and N_t represent the number of training examples of actors a and t, respectively. This assumes that the dataset is balanced in respect to the action labels for each actor. If this is not the case, each mini-batch used during training could be designed to fulfil this requirement.

To calculate the actor transfer loss at the m-th layer, the pairwise distances between all actors are summed:

$$\mathcal{L}^{(m)}_{transfer}(\theta_h) = \sum_{a\in A}\sum_{t\in A, t\neq a} D^{(m)}(a,t,\theta_h) \quad (5)$$

The final loss $\mathcal{L}_{transfer}$ is a weighted sum of the loss calculated at each layer of the feature extraction network, where $\beta^{(m)}$ is the weight attributed to the layer m, which controls the importance of that layer:

$$\mathcal{L}_{transfer}(\theta_h) = \sum_{m=1}^{M} \beta^{(m)}\mathcal{L}^{(m)}_{transfer}(\theta_h) \quad (6)$$

Therefore, the final objective can be written as:

$$\min_{\theta_h,\theta_f} \mathcal{L}(\theta_h,\theta_f,\theta_g) = \mathcal{L}_{action}(\theta_h,\theta_f) + \lambda\mathcal{L}_{adv}(\theta_h,\theta_g) + \gamma\mathcal{L}_{transfer}(\theta_h) \quad (7)$$

where λ and γ are weights attributed to each loss component to control their relative importance.

C. DEEP BILEVEL LEARNING

In this study, the Deep Bilevel Learning [41] methodology is explored as another strategy to achieve actor generalization. Deep Bilevel Learning improves the training process by giving different weights to each mini-batch in the training set. These mini-batch weights favour the batches whose gradients match the gradients of the validation mini-batches, minimizing the error on the validation set and resulting in a model with better generalization capabilities. A batch size of 16 was used, and for each weight update, 4 training mini-batches and 1 validation mini-batch with the same class distributions were selected. Figure 3 shows an overview of the methodology.

The weights are calculated using the following function:

$$\forall i \in T^t, \omega_i \leftarrow \sum_{j\in V^t} \frac{\nabla l_j(\theta^t)^T \nabla l_i(\theta^t)}{|\nabla l_i(\theta^t)|^2/\hat{\lambda}+\hat{\mu}}, \hat{\omega} = \frac{\omega}{|\omega|_1}, \quad (8)$$

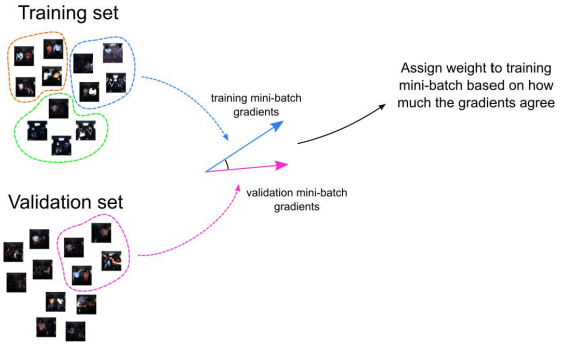


FIGURE 3. Overview of the Deep Bilevel Learning methodology (adapted from [41]).

where T^t represents the collection of training mini-batches used at the t-th training iteration, V^t is the collection of validation mini-batches used at the t-th training iteration, θ^t are the model parameters at the t-th training iteration, $\nabla l_i(\theta^t)$ are the gradients of the i-th mini-batch in the training set, $\nabla l_j(\theta^t)^T$ are the gradients of the j-th mini-batch in the validation set, ω_i is the weight attributed to the i-th mini-batch in the training set, $\hat{\lambda}$ is an adjustable hyperparameter, and $\hat{\mu}$ is a term added to avoid divisions by zero.

A new gradient descent step can be calculated as follows:

$$\theta(\omega) = \theta^t - \epsilon\sum_{i\in T^t}\hat{\omega}_i\nabla l_i(\theta^t), \quad (9)$$

where $\epsilon\hat{\omega}_i$ can be interpreted as a learning rate specific to each training mini-batch. When the gradients of a mini-batch in the training set $\nabla l_i(\theta^t)$ point in the same direction as the gradients of a mini-batch in the validation set $\nabla l_j(\theta^t)$, then their inner product is $\nabla l_j(\theta^t)^T\nabla l_i(\theta^t) > 0$; when the gradients point in different directions (do not agree) the inner product is $\nabla l_j(\theta^t)^T\nabla l_i(\theta^t) \leq 0$ which gives a weight with a negative value or zero.

D. USING POSE INFORMATION

The baseline model uses RGB data for action recognition and has achieved results comparable to state-of-the-art methodologies.

A series of experiments were conducted to measure the effects of using pose information as a complement or alternative to RGB data [42]. Using pose information as an input can have some advantages, as the model receives data that is simpler and easier to interpret, and can also help improve robustness to the actors. When using the raw RGB data, however, the model needs to extract and interpret more complex features, which can make the classification process more complex.

To calculate the key points of each person in a frame, a Keypoint R-CNN model with a ResNet-50-FPN backbone [54] is used. The model calculates the position of 17 key points, and whether they are visible or occluded. The key points calculated by the model are presented in Table 1 along with the corresponding assigned label (colour) and further illustrated in Figure 4.

TABLE 1. Key points calculated by the pose estimation model and their corresponding colour.

Keypoint	Color (RGB)
nose	(128, 0, 0)
left eye	(0, 128, 0)
right eye	(128, 128, 0)
left ear	(0, 0, 128)
right ear	(128, 0, 128)
left shoulder	(0, 128, 128)
right shoulder	(128, 128, 128)
left elbow	(64, 0, 0)
right elbow	(192, 0, 0)
left wrist	(64, 128, 0)
right wrist	(192, 128, 0)
left hip	(64, 0, 128)
right hip	(192, 0, 128)
left knee	(64, 128, 128)
right knee	(192, 128, 128)
left ankle	(0, 64, 0)
right ankle	(128, 64, 0)

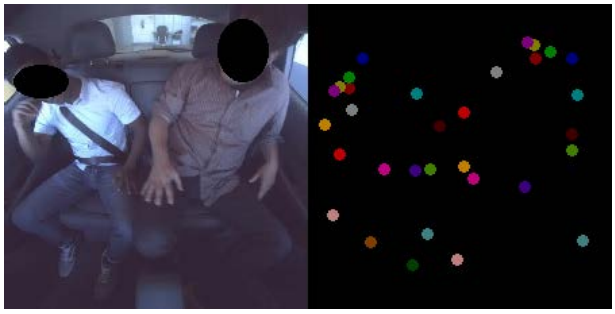


FIGURE 4. Key points calculated by the pose estimation model (Sunnyvale dataset from Bosch Car Multimedia).

To calculate the pose information for a certain video, we start by extracting each frame of the video. The pose information is extracted for each frame, consisting of the coordinates of the key points and the bounding box of each person in the frame.

For each frame, we generate an image with a black background and circles of a certain colour with a radius of 4 pixels, in the position of each key point. We added a different colour to each key point to label them, to make it possible for the model to distinguish each body part. Table 1 shows the key points and their corresponding colour. Figure 4 shows an example of an image and the corresponding key point map.

IV. DATA AND EXPERIMENTAL SETTINGS

As previously stated, the proposed methodologies are tested in a shared autonomous vehicle scenario. Given the lack of freely available appropriate data, two datasets from Bosch Car Multimedia were used, which will be designated Hanau03 Vito and Sunnyvale. Each of the datasets contains videos of people performing different actions inside the vehicle.

The Hanau03 Vito dataset contains 74 actors, and the videos were captured from above with a fisheye lens. Figure 5 shows a frame extracted from one of the videos.

FIGURE 5. Frame extracted from the Hanau03 Vito dataset from bosch car multimedia.

The Sunnyvale dataset has 9 actors, and the videos were recorded from the front without a fisheye lens, making them more similar to the videos in the MMIT dataset, which was used to pre-train the video processing sub-module. Figure 6 shows a frame extracted from one of the videos.

FIGURE 6. Frame extracted from the Sunnyvale dataset from bosch car multimedia.

In these experiments, we focus on detecting violence and non-violence. Each annotated video segment of the dataset was divided into sub-segments that will be referred to as samples. These samples are then used to train and test the model. Each sample has a duration of 1 second, and a frame-rate of 8 frames per second (fps), which means that for each prediction, the model receives 8 concatenated frames with a resolution of $224 \times 224 \times 3$ pixels. The number of samples extracted were 32739 for the Hanau03 Vito dataset, and 46838 for the Sunnyvale dataset.

Table 2 contains the number of samples that were extracted from the datasets, the number of actors, and the average number of samples per actor.

As previously mentioned, an actor is defined as a specific group of people inside the vehicle (and not a specific person). This means that different actors could have some individuals in common.

TABLE 2. Number of samples and actors in the Hanau03 Vito and Sunnyvale datasets.

	No. Samples	No. Actors	No. Samples/Actor (mean $\pm$ 2 $\times$ std)
Hanau03 Vito	32739	74	442 $\pm$ 968
Sunnyvale	46838	9	5211 $\pm$ 6828

When splitting the dataset into train set, validation set and test set, it was ensured that each specific person was not in different sets simultaneously, meaning that each of the train, validation, and test sets were completely independent of each other in terms of actors. The datasets were split in a way that kept the number of samples in each class relatively balanced. Table 3 and Table 4 show the dataset splits.

TABLE 3. Hanau03 Vito dataset train-validation-test split.

	Non-violence samples	Violence samples	No. Samples	No. Actors
Training Set	9752	11744	21496	49
Validation Set	3255	3372	6627	10
Test Set	1762	2854	4616	15
Total	14769	17970	32739	74

TABLE 4. Sunnyvale dataset train-validation-test split.

	Non-violence samples	Violence samples	No. Samples	No. Actors
Training Set	16351	16416	32767	5
Validation Set	5137	4864	10001	3
Test Set	3093	977	4070	1
Total	24581	22257	46838	9

Since the focus of this work is the detection of violence/non-violence the original classes of the datasets were grouped into those two categories. Table 5 and Table 6 show the original classes and their classification as violence or non-violence.

In addition to the Hanau03 Vito and Sunnyvale datasets three widely used and publicly available datasets are used: Moments in Time [32]; HMDB51 [35]; Hollywood [55]. When analysing the pose keypoint data it was noticed that the pose keypoints calculated for the Hanau03 Vito dataset were inaccurate in some situations due to the position of the camera (top view). Therefore, to further test the methodology that used the pose information, we decided to add these datasets. The advantages of these datasets are that they are all publicly available, and they contain actions that could be performed inside the vehicle.

The Moments in Time dataset is a large-scale action dataset. It contains one million 3-second videos and 339 classes. The HMDB51 dataset contains 6849 video clips from 51 action classes obtained from movies and web videos. The Hollywood dataset is a human action dataset which contains video samples obtained from 32 movies. Each sample has one or more labels corresponding to 8 action classes.

For each of the publicly available datasets, a subset of action classes was selected, based on the relevance for this study and for the context of in-vehicle action recognition.

TABLE 5. Action grouping as violence or non-violence for the Hanau03 Vito dataset.

Class	Violence
entering	No
leaving	No
blucke on	No
turning head	No
lay down	No
sleeping	No
stretching	No
changing seating position	No
changing clothes	No
reading	No
use mobile phone	No
making a call	No
posing	No
waving hand	No
drinking	No
eating	No
singing	No
pick up item	No
come closer	No
handshaking	No
talking	No
dancing	No
finger pointing	No
leaning forward	No
tickling	No
hugging	No
kissing	No
elbowing	Yes
provocative	Yes
pushing	Yes
protecting oneself	Yes
stealing	Yes
screaming	Yes
pulling	Yes
arguing/abuse	Yes
grabbing	Yes
touching	Yes
slapping	Yes
pushing	Yes
strangling	Yes
fighting	Yes
threatening weapon	Yes

TABLE 6. Action grouping as violence or non-violence for the Sunnyvale dataset.

Class	Violence
handshaking	No
hugging	No
talkRight	No
talkLeft	No
talking	No
arguing	Yes
touching	Yes
grabbing	Yes
pushing	Yes
kicking	Yes
slapping	Yes
elbowing	Yes
punching	Yes
fighting	Yes

V. RESULTS

A. ADVERSARIAL LEARNING

After training on the Hanau03 Vito dataset the baseline had an accuracy of 65.33% and the adversarial learning

methodology had an accuracy of 62.42%. The actor classification accuracy during training was 2.04%. The actor accuracy should be close to random, as this indicates that the methodology is correctly removing the actor information from the features of the model. Table 7 shows the results.

TABLE 7. Model accuracy after training on the Hanau03 Vito dataset using the baseline and the adversarial learning methodology.

	Baseline Accuracy (%)	Adversarial Learning Accuracy (%)
Training Set	98.75	94.11
Validation Set	57.18	54.25
Test Set	65.33	62.42

After training on the Sunnyvale dataset the baseline had an accuracy of 82.81% and the adversarial learning methodology had an accuracy of 58.11%. The actor classification accuracy during training was 35.96%. Table 8 shows the results.

TABLE 8. Model accuracy after training on the Sunnyvale dataset using the baseline and the adversarial learning methodology.

	Baseline Accuracy (%)	Adversarial Learning Accuracy (%)
Training Set	99.25	98.84
Validation Set	89.30	77.62
Test Set	82.81	58.11

The obtained results indicate that the presented methodology did not improve the results of the baseline model. The hyperparameters used for the experiments were the ones proposed in the paper [2], and given that this problem uses a different dataset and a different network architecture, some tuning could be needed. Additionally, the datasets used are very small and contain a low number of samples per actor, which further increases the difficulty of the problem by making the network unable to correctly learn the distribution of each actor, making this methodology less effective. An interesting experiment would be to train the model from scratch, to be able to remove actor information from the first layers of the network, but more data would be required, since training the model from scratch with these datasets would result in an overfit model that would perform poorly when presented with new data.

B. BILEVEL LEARNING

After training on the Hanau03 Vito dataset, the model achieved an accuracy of 65.33% on the baseline and 68.19% on the bilevel learning methodology. Table 9 show the results.

After training on the Sunnyvale dataset the model achieved an accuracy of 82.81% on the baseline and 86.62% on the bilevel learning methodology. Table 10 show the results.

Although the gradients of the validation set are not used directly to train the model, there may be some data leakage during the training process, since more weight is given to the training mini-batches that have gradients similar to the gradients of the validation mini-batches. This is also a reason why the data is split into training, validation and testing.

TABLE 9. Accuracy after training on the Hanau03 Vito dataset using the baseline and the bilevel learning methodology.

	Baseline Accuracy (%)	Bilevel Learning Accuracy (%)
Training Set	98.75	91.70
Validation Set	57.18	69.77
Test Set	65.33	68.19

TABLE 10. Accuracy after training on the Sunnyvale dataset using the baseline and the bilevel learning methodology.

	Baseline Accuracy (%)	Bilevel Learning Accuracy (%)
Training Set	99.25	92.15
Validation Set	89.30	89.96
Test Set	82.81	86.62

On both datasets, the training set accuracy decreased when using bilevel learning, which means that the model is not overfitting as much to the training data. There is also an increase in accuracy in both the validation set and the test set.

C. USING POSE INFORMATION

Previously, the input of the model was the RGB data of the video frames. This section presents two experiments that aim to assess if inputting pose information into the model has any impact on the performance.

The first experiment consists of training the model using only pose information, where instead of giving the model the RGB frames, the model is given the generated key point map. The second experiment consists of training the model with the pose information as a complement to the RGB frames. To accomplish this, the RGB frames are concatenated with the key point maps before giving them to the model, which results in a total of 6 channels (labels in the key point map are colours expressed in RGB). To accommodate for this change, the number of input channels of the network is duplicated and the pre-trained weights are copied to the new channels. The first layer of the network is also unfrozen, as 3 more channels were added to the filters of this layer.

Table 11, Table 12 and Table 13 show the results on the Moments in Time, HMDB51 and Hollywood datasets, respectively.

Results show that using the pose information as an alternative to the RGB data does not translate into an improvement in performance. The only information that is given to the model is the position of the body parts in each frame, which makes the problem more difficult as there is less available information.

The results on the model trained with the RGB frames and the pose information suggest that there is an improvement for most classes. This could mean that there is an advantage to giving the model the pose information, as it can more easily extract relevant information about movement or the position

TABLE 11. Results after training the model using the Moments in Time dataset, with RGB, pose only, and RGB + pose.

Class	RGB (%)	Pose (%)	RGB + Pose (%)
fighting	**60.00**	20.00	55.00
punching	72.97	8.11	**78.38**
pushing	38.46	30.77	**61.54**
sitting	23.33	3.33	**33.33**
sleeping	49.47	3.16	**53.68**
coughing	14.28	0.00	**28.57**
singing	**72.42**	58.88	68.69
speaking	**50.00**	45.24	32.14
discussing	26.38	11.11	**31.94**
pulling	36.36	9.09	**40.91**
slapping	29.03	9.68	**48.39**
hugging	**82.35**	5.88	**82.35**
kissing	**18.18**	0.00	**18.18**
reading	14.28	0.00	**28.57**
telephoning	**45.45**	0.00	**45.45**
studying	58.82	0.00	**76.47**
socializing	10.00	**15.00**	10.00
resting	46.66	13.33	**53.33**
celebrating	**66.66**	25.00	62.50
laughing	**20.00**	0.00	**20.00**
eating	**71.42**	0.00	**71.42**
all classes	43.17	12.31	**47.66**

TABLE 12. Results after training the model using the HMDB51 dataset, with RGB, pose only, and RGB + pose.

Class	RGB (%)	Pose (%)	RGB + Pose (%)
punching	67.31	43.46	**74.81**
pushing	91.48	67.78	**92.59**
speaking	63.44	40.47	**69.38**
hugging	**95.24**	72.62	89.76
kissing	**100.00**	42.67	90.67
laughing	83.75	72.29	**92.08**
eating	**78.33**	51.11	73.61
all classes	82.79	55.77	**83.27**

TABLE 13. Results after training the model using the Hollywood dataset, with RGB, pose only, and RGB + pose.

Class	RGB (%)	Pose (%)	RGB + Pose (%)
hugging	9.09	6.82	**20.45**
kissing	**91.67**	68.33	83.54
telephoning	57.14	36.07	**62.50**
all classes	52.63	37.07	**55.50**

of people in a given frame, improving the accuracy of the network.

There seems to be no advantage in using the pose information for the "hugging", "kissing" and "eating" classes, when comparing the classes in common between the Moments in Time and HMDB51 datasets. The Hollywood dataset also shared the same results, showing no improvement when using the pose information for the class "kissing". This might mean that some classes benefit more from using the pose information than others.

Table 14 and Table 15 show the results on the Hanau03 Vito and Sunnyvale datasets, respectively. Since there was the need to unfreeze the first layer of the network for the RGB + Pose experiment, we wanted to test if unfreezing the first layer on the other experiments would have any significant impact on the performance. Therefore, the tables present the results with the first layer frozen and unfrozen.

TABLE 14. Results after training the model using the Hanau03 Vito dataset, with RGB, pose only, and RGB + pose.

Class	RGB unfrozen (%)	RGB	Pose unfrozen (%)	Pose	RGB + Pose (%)
non-violence	65.83	**66.00**	59.93	62.59	45.51
violence	72.98	64.64	53.95	60.72	**74.38**
all classes	**69.41**	65.33	56.94	61.66	59.95

TABLE 15. Results after training the model using the Sunnyvale dataset, with RGB, pose only, and RGB + pose.

Class	RGB unfrozen (%)	RGB	Pose unfrozen (%)	Pose	RGB + Pose (%)
non-violence	21.11	74.62	**78.14**	70.22	37.31
violence	**96.11**	90.99	90.17	89.55	91.09
all classes	58.61	82.81	**84.16**	79.89	64.20

Once again, the model achieves reasonable results when using the pose information as a complement to the traditional RGB frames. It also seems that due to the more controlled scenario found in the Sunnyvale dataset, using only the pose information was enough to achieve results superior or comparable to the baseline (RGB column). The results on the Hanau03 Vito dataset did not show any improvements over the baseline model when using the pose information. This could be caused by the camera angle of the dataset (top view), since it makes it difficult to detect the keypoints, making them inaccurate and thus affecting the results.

Since the pose information only contains the position of the actor and its body parts, the model is less likely to become biased towards a certain actor. The downside is that the performance of the model is dependent on the accuracy of the keypoints.

VI. CONCLUSION AND FUTURE WORK

This paper focused on methodologies that counteract actor bias, by removing specific actor information from the features of the model, or by using data that is actor independent, such as pose information.

The first methodology used an adversarial approach to remove actor-related information from the feature vectors of the model. Although the results did not show an improvement, some fine-tuning of the hyperparameters could give better results.

The second methodology assigned a weight to training mini-batches based on how much their gradients "agreed" with the gradients of the validation mini-batches. Results show an improvement over the baseline, and since the gradients of the validation set are taken into account (which contain different actors), this methodology is giving a preference to the training mini-batches that give the model better generalization capabilities.

The third methodology studied the impact of giving the model the pose information of the actors. Since the keypoints do not contain much information regarding the actor, we wanted to test if this was a viable strategy to reduce actor bias. Results show some improvements over the baseline on public datasets, and comparable results on the Hanau03 Vito and Sunnyvale datasets.

For future work, it would be interesting to explore the use of these methodologies together and see if there are any significant improvements. Since the use of these methodologies had results aligned or superior to our baseline model, we wonder if the combination of them could achieve even better results.

REFERENCES

[1] J. R. Pinto, P. Carvalho, C. Pinto, A. Sousa, L. Capozzi, and J. S. Cardoso, "Streamlining action recognition in autonomous shared vehicles with an audiovisual cascade strategy," in *Proc. 17th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl. (VISAPP)*, 2022, pp. 467–474.

[2] P. M. Ferreira, D. Pernes, A. Rebelo, and J. S. Cardoso, "Learning signer-invariant representations with adversarial training," in *Proc. 12th Int. Conf. Mach. Vis. (ICMV)*, vol. 11433, W. Osten and D. P. Nikolaev, Eds. SPIE, 2020, p. 114333D, doi: 10.1117/12.2559534.

[3] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[4] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. CVPR*, Jun. 2011, pp. 3169–3176.

[5] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3551–3558.

[6] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[8] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal residual networks for video action recognition," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2016, pp. 3476–3484.

[9] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7445–7454.

[10] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4694–4702.

[11] G. Cheron, I. Laptev, and C. Schmid, "P-CNN: Pose-based CNN features for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3218–3226.

[12] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[13] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[14] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6546–6555.

[15] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4724–4733.

[16] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Computer Vision—ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 318–335.

[17] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5534–5542.

[18] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.

[19] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6201–6210.

[20] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 579–583.

[21] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4570–4579.

[22] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1963–1978, Aug. 2019.

[23] H. Duan, Y. Zhao, K. Chen, D. Shao, D. Lin, and B. Dai, "Revisiting skeleton-based action recognition," 2021, *arXiv:2104.13586*.

[24] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[25] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *CoRR*, vol. abs/1801.07455, 2018. [Online]. Available: http://arxiv.org/abs/1801.07455

[26] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 140–149.

[27] Y. Li, Y. Li, and N. Vasconcelos, "Resound: Towards action recognition without representation bias," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 513–528.

[28] Y. Li and N. Vasconcelos, "REPAIR: Removing representation bias by dataset resampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9572–9581.

[29] J. Choi, C. Gao, J. C. E. Messou, and J.-B. Huang, "Why can't I dance in the mall? Learning to mitigate scene bias in action recognition," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019.

[30] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. C. Niebles, "What makes a video a video: Analyzing temporal information in video understanding models and datasets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7366–7375.

[31] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *CoRR*, vol. abs/1705.06950, pp. 1–22, May 2017.

[32] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick, and A. Oliva, "Moments in time dataset: One million videos for event understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 502–508, Feb. 2020.

[33] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 961–970.

[34] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.

[35] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2556–2563.

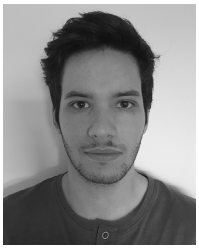
[36] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic, "The 'something something' video database for learning and evaluating visual common sense," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5843–5851.

[37] J. Materzynska, G. Berger, I. Bax, and R. Memisevic, "The jester dataset: A large-scale video dataset of human gestures," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2874–2882.

[38] P. Augusto, J. S. Cardoso, and J. Fonseca, "Automotive interior sensing—Towards a synergetic approach between anomaly detection and action recognition strategies," in *Proc. IEEE 4th Int. Conf. Image Process., Appl. Syst. (IPAS)*, Genova, Italy, Dec. 2020, pp. 162–167.

[39] M. Pinto, "Automotive interior sensing: Human interaction recognition," M.S. thesis, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal, 2021.

[40] J. R. Pinto, T. Gonçalves, C. Pinto, L. Sanhudo, J. Fonseca, F. Gonçalves, P. Carvalho, and J. S. Cardoso, "Audiovisual classification of group emotion valence using activity recognition networks," in *Proc. 4th IEEE Int. Conf. Image Process., Appl. Syst. (IPAS)*, Genova, Italy, Dec. 2020, pp. 114–119.
[41] S. Jenni and P. Favaro, "Deep bilevel learning," *CoRR*, vol. abs/1809.01465, pp. 1–16, Sep. 2018.
[42] V. Choutas, P. Weinzaepfel, J. Revaud, and C. Schmid, "PoTion: Pose motion representation for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7024–7033.
[43] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," *CoRR*, vol. abs/1705.07750, pp. 1–10, May 2017.
[44] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: Generic features for video analysis," *CoRR*, vol. abs/1412.0767, pp. 1–8, Dec. 2014.
[45] M. E. Kalfaoglu, S. Kalkan, and A. A. Alatan, "Late temporal modeling in 3D CNN architectures with BERT for action recognition," 2020, *arXiv:2008.01232*.
[46] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "Multi-fiber networks for video recognition," *CoRR*, vol. abs/1807.11195, pp. 1–16, Jul. 2018.
[47] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," *CoRR*, vol. abs/1812.03982, pp. 1–10, Dec. 2018.
[48] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" *CoRR*, vol. abs/1711.09577, pp. 1–10, Nov. 2017.
[49] A. J. Piergiovanni, A. Angelova, A. Toshev, and M. S. Ryoo, "Evolving space-time neural architectures for videos," *CoRR*, vol. abs/1811.10636, pp. 1–18, Nov. 2018.
[50] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," *CoRR*, vol. abs/1904.02811, pp. 1–11, Apr. 2019.
[51] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," *CoRR*, vol. abs/1711.11248, pp. 1–10, Nov. 2017.
[52] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning for video understanding," *CoRR*, vol. abs/1712.04851, pp. 1–10, Dec. 2017.
[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Las Vegas, NV, USA: IEEE Computer Society, Jun. 2016, pp. 770–778.
[54] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, pp. 1–12, Mar. 2017.
[55] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.



LEONARDO CAPOZZI received the M.Sc. degree in informatics and computing engineering from the Faculdade de Engenharia da Universidade do Porto (FEUP), in 2020, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. He is also a Research Assistant with the Visual Computing and Machine Intelligence Research Group, INESC TEC, Porto, Portugal. His M.Sc. thesis focusing on identifying suspects using drawn sketches. His Ph.D. studies are focused on the detection of actions in video, and the use of biometrics and soft biometrics to identify individuals. His research interests include computer vision, pattern recognition, biometrics, and machine learning in general.

VÍTOR BARBOSA is currently pursuing the master's degree in informatics engineering and computing with the Faculdade de Engenharia da Universidade do Porto (FEUP). He is also working on his M.Sc. dissertation focusing on activity recognition in scenarios with multiple individuals. His research interests include computer vision, action recognition, and machine learning in general.

CAROLINA PINTO has been a Deep Learning Researcher at Bosch, since 2020, for interior vehicle sensing and autonomous driving. Her M.Sc. dissertation was on human interaction recognition using visual information captured from sensors inside the vehicle. Her main work conducted at Bosch was focused on using audiovisual information for occupant emotional monitoring, violence detection, and activity recognition. Her research interests include pattern recognition, biometrics, deep learning, and computer vision.

JOÃO RIBEIRO PINTO (Student Member, IEEE) received the M.Sc. degree in bioengineering (field of biomedical engineering) from the Faculdade de Engenharia da Universidade do Porto (FEUP), in 2017, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. He is also a Research Assistant with the Visual Computing and Machine Intelligence Research Group, INESC TEC, Porto, Portugal. His research has focused on contributing to make the ECG a viable and stronger biometric characteristic in realistic conditions. His M.Sc. thesis on the use of the electrocardiogram (ECG) for biometric recognition of vehicle drivers. His Ph.D. studies are focused on using ECG and face, both acquired almost unnoticeably from vehicle drivers, to recognize them, and continuously monitor their wellbeing. His research interests include biometrics, biosignals, pattern recognition, computer vision, and machine learning in general.

AMÉRICO PEREIRA received the B.Sc. and M.Sc. degrees in computer science from the Faculty of Science, University of Porto, Portugal, in 2011 and 2013, respectively, where he is currently pursuing the Ph.D. degree in electrical and computer engineering with the Faculty of Engineering. He joined the INESC TEC—Institute for Systems and Computer Engineering, Technology and Science, in 2014, where he is currently a Researcher with the Center of Telecommunications and Multimedia. His research interests include computer vision and image/video processing, with an emphasis on machine learning.

PEDRO M. CARVALHO (Senior Member, IEEE) is currently a Senior Researcher at INESC TEC and an Invited Professor at the School of Engineering, Polytechnic of Porto. As part of its activities at INESC TEC, he participated or was a principal investigator in more than 20 research and development projects, including national, European, and with companies, and has more than 40 papers published in international journals and conferences. His research interests include computer vision, multimedia systems, and decision support systems.

JAIME S. CARDOSO (Senior Member, IEEE) is currently a Full Professor at the Faculty of Engineering, University of Porto (FEUP). From 2012 to 2015, he served as the President of the Portuguese Association for Pattern Recognition (APRP), affiliated to the IAPR. His research interests include computer vision, machine learning, and decision support systems, image and video processing focuses on medicine and biometrics, the work on machine learning cares mostly with the adaptation of learning to the challenging conditions presented by visual data, with a focus on deep learning and explainable machine learning. The particular emphasis of the work in decision support systems goes to medical applications, always anchored on the automatic analysis of visual data. He has coauthored more than 300 papers, more than 100 of which in international journals, which attracted more than 7000 citations, according to Google scholar.