Impact of visual noise in activity recognition using deep neural networks - an experimental approach

Leonardo Capozzi
INESC TEC and
Faculty of Engineering
University of Porto
Porto, Portugal
leonardo.g.capozzi@inesctec.pt

Pedro Carvalho
INESC TEC and
School of Engineering
Polythecnic of Porto
Porto, Portugal
pedro.carvalho@inesctec.pt

Afonso Sousa
INESC TEC and
Faculty of Engineering
University of Porto
Porto, Portugal
afonso.s.sousa@inesctec.pt

Carolina Pinto
Bosch Car Multimedia
Braga, Portugal
carolina.pinto@pt.bosch.com

João Ribeiro Pinto
INESC TEC and
Faculty of Engineering
University of Porto
Porto, Portugal
joao.t.pinto@inesctec.pt

Jaime S. Cardoso
INESC TEC and
Faculty of Engineering
University of Porto
Porto, Portugal
jaime.cardoso@inesctec.pt

***Abstract*—The popularity of deep learning methods has increased significantly, in no small part due to their impressive performance in several application scenarios. This paper focuses on recognising activities in an in-vehicle environment and measuring the impact that factors such as resolution, aspect ratio, field of view and framerate have on the performance of the model. The use of deep learning methodologies in recent years has increased the amount of data required to train and test the models. However, such data is often insufficient, unavailable, or lacks suitable properties. Publicly available action recognition datasets have been analysed, collected, and prepared to assess the classification results in such scenarios, which provides important guidance for use in a real-world setting.**

***Index Terms*—activity recognition, deep learning, robustness**

I. Introduction

With the increased popularity of deep machine learning, data has gained augmented importance with many initiatives targeting the collection and preparation of datasets. However, in areas such as autonomous (shared) vehicles, these datasets are scarce, may not have the necessary characteristics, and are essentially private. This poses a problem for the research and development of models in topics for these scenarios since large amounts of data are needed to train robust models.

The recognition of emotions, activities, and unwanted behaviours plays a key role in autonomous shared vehicles [1]. Unwanted behaviour is an action or activity that can be described as undesirable in a given context due to its potentially harmful consequences. To this end, the set of human activities that can be categorised as unwanted differs according to the considered scenario. For example, eating or drinking in a shared car may be regarded as unwanted since it may cause discomfort to other passengers.

This paper analyses the impact of visual noise on deep neural network-based models for the recognition of a set of human activities, with three main contributions. First, given the general unavailability of datasets targeting the proposed specific scenario, several public datasets were analysed, collected, and filtered to identify segments encompassing a subset that more closely resembles the target scenario. We studied the impact of different variations of input information (frame rate, resolutions, aspect ratio), both from a computational performance and model accuracy perspective. Also, the performance of the model under different types of visual noise, such as cropping and occlusions, was analysed since these are quite common in footage obtained from the inside of the vehicle where parts of the body might not be completely visible.

The remainder of this paper is organized as follows: the proposed methodology is presented in section II; the data and training process are detailed in section III; the conducted experiments are explained in section IV; section V presents and discusses the results; and conclusions are drawn in section VI.

II. Methodology

This paper presents the results of a study on the robustness of models for classifying human activity in the presence of different types of noise. In the last years, advances in deep learning methodologies have surpassed previous approaches to the problem, with 3D Convolution Neural Networks (CNN) showing enormous potential.

3D CNNs are formed of 3D convolutions, which allow the network to process the temporal information of the input

This work is supported by: European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project no. 039334; Funding Reference: POCI-01-0247-FEDER-039334] and by the Portuguese funding agency, FCT – Fundação para a Ciência e a Tecnologia within the PhD grant "SFRH/BD/137720/2018".

throughout the network [2]–[11]. Before 3D networks, models consisted of multiple streams that processed the temporal component; however, these methods used 2D convolutions, which meant that the temporal information was added to the channels, sometimes limiting their performance. The output of a 2D convolution is always an image, which means that if the temporal component of a video is added to the channels, then the information is more easily lost. The output of a 3D convolution, on the other hand, is a video volume that preserves the temporal information of the input, making it better for video classification.

The model used in this study is based on the model proposed in [12], which follows the architecture presented in Figure 1. It is composed of the 3D ResNet50's convolutional encoder blocks, followed by an average pooling layer, dropout layer, and fully-connected layer with an input size of 2048 and an output size equal to the number of classes. The convolutional encoder is composed of one 3D convolutional layer containing 64 filters with a size of $7 \times 7 \times 7$, followed by a batch normalisation layer and ReLU activation function. The subsequent layers are three residual blocks of type A, four blocks of type B, six blocks of type C, and four blocks of type D. Each block contains 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 first two convolutional layers of each block have 64 (type A), 128 (type B), 256 (type C), or 512 filters (type D), and are followed by a batch normalisation layer. The last convolutional layer of each block has four times the number of filters as the first two layers and is followed by a batch normalisation layer and ReLU activation function. More details on the network architecture are available on the original ResNet paper [13].

III. Experimental Setup

A. Data

Given the absence of public datasets specific to the target scenario, several datasets were analysed and filtered to identify an adequate subset.

The following were selected for the subsequent tests: Moments in Time [14]; HMDB51 [15]; Hollywood [16], [17]; MSR DailyActivity3D [18]; UWA3D Multiview [19]; SBU [20].

For training the model, 21 classes from the Moments in time were selected, considering those that are more closely related to the scenario of autonomous shared vehicles. For each dataset, we selected the classes that were in the set of 21 classes used to train the model, which allowed us to compute cross-dataset accuracies. Table I provides more detailed information about the datasets and the corresponding number of classes.

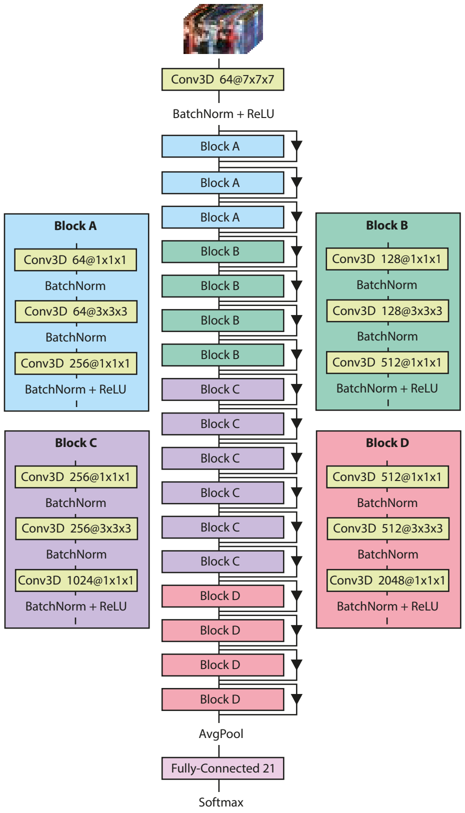

Fig. 1. Network architecture (adapted from [12]).

TABLE I
SUMMARY OF THE DATASETS AND CORRESPONDING CLASSES USED IN THE EXPERIMENTS.

Dataset	Number of videos used	Number of classes used	Resolution
Moments in Time [14]	10771	21	varied
HMDB51 [15]	814	7	varied
Hollywood [16], [17]	243	3	varied
MSR DailyActivity3D [18]	60	3	640x480
UWA3D Multiview [19]	107	2	320x240
SBU [20]	100	3	640x480

B. Training

The network used in this work (described in Section II) was pre-trained on the Moments in Time dataset, and later fine-tuned with a subset of 21 classes from the Moments in Time dataset. These classes were chosen because they were the most relevant for the in-vehicle scenario (see Table XI for the list of classes). All videos were resized to 224×224 resolution and then normalised using the mean and standard deviation from ImageNet, as these were the settings used on the pre-trained network. This pre-processing step was applied to all videos when training and testing the model. Since the base network was trained with 21 classes, we chose datasets that contained at least 2 of these 21 classes. This allowed us to test whether there was a drop in performance when videos from datasets other than the one it was trained on were presented to the model.

During training, the parameters of the 3D ResNet-50 layers were frozen, except for the final fully-connected layer. This was done to take advantage of the pre-training on the Moments in Time dataset [14].

The loss function used to train the model was cross-entropy, with a batch size of 32 videos, and the Adam optimiser with a learning rate of 1×10^{-4}.

IV. Experiments

The following sub-sections describe a set of experiments intended to assess the impact of a given type of visual noise or input characteristics, namely: input resolution; frame rate; obstruction of the field-of-view (FOV). The first two types are important from two different axes: noise and efficiency. Reducing the resolution or the frame rate means less information to be processed which can translate to less memory required and fewer operations (less computational complexity), but in turn, it may imply that relevant information is absent. This is an important trade-off, especially in autonomous (electric) vehicles, as computational resources are limited, and power consumption is a major concern. FOV obstructions, or occlusions, are important factors, favoured by the camera positioning in the vehicles and reduced space.

A. Input resolution

The base model was trained on videos downsized to 224 × 224 resolution; therefore, videos with a different aspect ratio would be stretched. It is relevant to assess if the videos' original aspect ratio would affect the performance of the model. Note that the network accepts different resolutions as it contains an adaptive average pooling layer. For the first experiment, the original aspect ratio was preserved by resizing the smaller dimension of the image to 224px and adjusting the other dimension accordingly. In a second experiment, the videos were inputted to the model without any resizing, i.e., keeping the original resolution.

The next experiment consisted of slightly adjusting the weights of the model trained on the MMIT dataset (finetuning). To accomplish this, we performed a small training session using the HMDB51 dataset and the Hollywood dataset. The idea was to allow the already trained network to adapt to the new datasets in order to get a performance gain.

B. FOV obstruction

Several tests were performed to verify the performance of the model in scenarios where some visual information is missing, since there are situations inside the vehicle where the field of view is obstructed, and the sensor can only see part of the person. To simulate these scenarios, parts of the video that may contain relevant information were cropped. More specifically, the following cases were analysed: cropping at the waist; cropping the right side of the body; cropping the left side of the body.

In order to automatically calculate where to crop the frames, the pose of each person in the frame was first calculated, extracting the corresponding bounding box and the coordinates of the main body parts. The pose extraction model [21] has the advantage of calculating 3D coordinates (it also estimates the distance from the camera). In situations where the individuals were very close to the camera and parts of the body were not visible, e.g. the legs, the method randomly placed the key points corresponding to these parts on the screen. This did not have a negative effect as the key points used were visible on the screen in most of the videos. Figure 2 shows the key points predicted by this model.

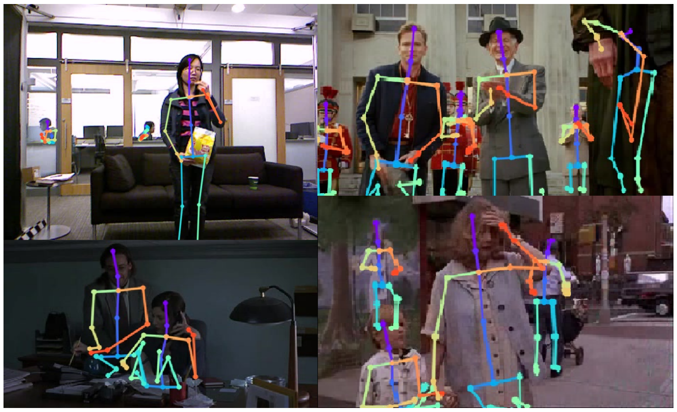

Fig. 2. Key points calculated by the pose estimation model.

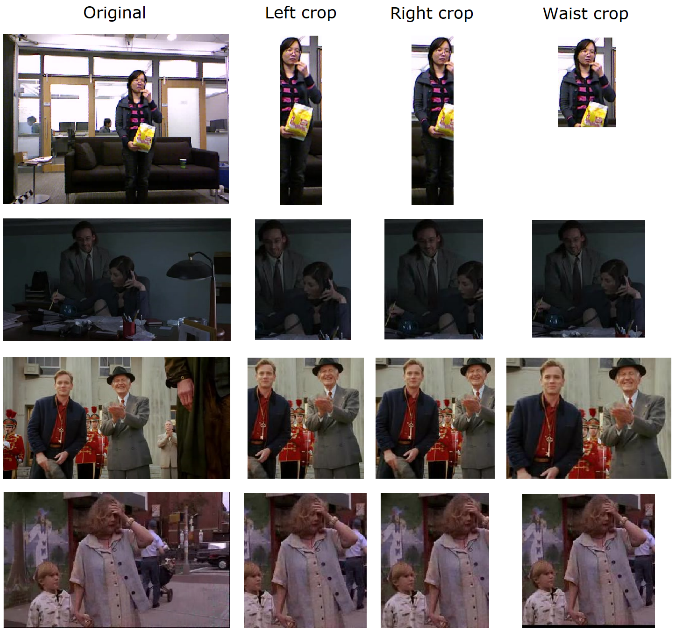

Fig. 3. Frames before and after cropping.

Once the pose was determined, the crop of the image was calculated. In situations where there was only one person in the frame, calculating the crop was fairly straightforward as only that person needed to be considered. In situations where multiple people were visible, it was necessary to determine which of them were in the foreground (performing the actions) and which were in the background (irrelevant to the action recognition task). This process took into account the size of the largest person in the frame and assumed that all people above a certain threshold were in the foreground. When performing a right/left side crop, we cropped 30% of the rightmost/leftmost person in the frame. When performing a waist crop, we cropped at the waist of the person nearest to the bottom of the frame. Figure 3 shows examples of different crops.

In a subsequent test, the model was fine-tuned with data augmentation (randomly selecting right crop, left crop, waist crop, or keeping the original for each video) to check if there would be any performance gain. The model was fine-tuned using the MMIT, HMDB51, and Hollywood datasets. Data augmentation was used to avoid overfitting and improve the performance of models by increasing the amount of data, through the application of transformations to the original data. In this case, the transformation applied could be too drastic to improve accuracy, possibly removing information that is critical to make a prediction. This could have the opposite effect and hinder the performance of the model. To test whether this was the case, we checked the accuracy of the model that was fine-tuned on the Moments in Time dataset with data augmentation, on the other datasets.

C. *Input frame rate*

We performed a series of tests to verify the impact that the input frame rate had on the model's accuracy. It is important to know if different frame rates have or not a large impact on the accuracy of the model, since using higher frame rates comes with associated computational costs, which is not ideal in situations where the computational power available is limited. In these experiments, we trained the model on the MMIT, HMDB51, and Hollywood datasets, with a training frame rate of 1, 2, 4, and 8 frames per second. This resulted in 12 different models, each trained with a different combination of dataset and frame rate. We tested the accuracy of each of the trained models using frame rates of 1, 2, 4, and 8 frames per second.

V. Results

A. *Input resolution*

Table II shows the performance of the model under changes in resolution and aspect ratio. As expected, the performance of the model decreased slightly as it was presented with data that did not match the resolution of the data it had been trained with. Preserving the aspect ratio of the original input caused a slight decrease in performance and using the original higher resolution also did not add any benefit. The model also showed good accuracy on the datasets it was not trained on, demonstrating that it was able to apply what it learned from the Moments in Time dataset to other datasets.

Table III shows the performance of the model after finetuning the parameters. Performance increased considerably, since it was better adapted to the new data, rather than relying exceptionally on what it had previously learned from the Moments in Time dataset. In terms of resolution and aspect ratio changes, the results were consistent with those in Table II and showed a slight decrease in accuracy when using the original aspect ratio.

The use of lower resolutions can be an important advantage in autonomous vehicle scenarios as they reduce the computational complexity or, in other words, increase the amount of information that the model can process in the same period of time. Table IV shows the number of frames processed by the model per second, using an NVIDIA GeForce GTX 1080 GPU, depicting a significant gain. The advantage is clear, especially considering that, as previously shown, these resolution differences have a very small impact on model accuracy.

TABLE II
Model accuracy with and without rescaling after training on MMIT

Dataset	Rescaling to 224x224px (%)	Rescaling to 224px preserving aspect ratio (%)	Original resolution (%)
MMIT	**51.49**	51.23	44.62
HMDB51	**69.53**	67.44	66.46
Hollywood	**52.67**	49.79	48.56
MSRDailyAct3D	**45.00**	**45.00**	41.67
UWA3D Mult	**70.09**	71.03	**70.09**
SBU	**50.00**	48.00	35.00

TABLE III
Model accuracy with and without rescaling after fine-tuning on each dataset.

Dataset	Rescaling to 224x224px (%)	Rescaling to 224px preserving aspect ratio (%)	Original resolution (%)
HMDB51	**78.53**	77.91	77.91
Hollywood	**67.35**	65.31	65.31

B. *FOV obstruction*

Table V summarises the accuracy of the model after performing the different crops to simulate obstructions in the field of view. For the MMIT, HMDB51, Hollywood, and SBU datasets, a slight drop in performance is observed due to cropping. However, the MSR DailyActivity3D and the UWA3D Multiview datasets did not suffer from such a drop. This is likely because they contain simpler studio-like scenarios; when we apply a crop, the action information is not completely lost, but in many cases, the background noise is removed, and therefore the accuracy does not decrease.

Table VI shows the performance of the model after finetuning. The performance increased considerably for the HMDB51 and Hollywood datasets, but there was not much gain for the MMIT dataset, as this was the dataset that the model was originally trained on.

TABLE IV
Processing capabilities of the model using different resolutions.

Resolution	Frames Per Second
224x224px	1043
300x224px	757
640x480px	134

TABLE V
MODEL ACCURACY AFTER TRAINING ON MMIT.

Dataset	No crop (%)	Waist crop (%)	Right crop (%)	Left crop (%)
MMIT	**51.49**	44.88	45.40	45.65
HMDB51	**69.53**	54.05	57.25	55.04
Hollywood	**52.67**	51.44	46.50	50.21
MSRDailyAct3D	45.00	**71.67**	56.67	63.33
UWA3D Mult	70.09	**84.11**	72.90	78.50
SBU	50.00	**51.00**	43.00	38.00

TABLE VI
MODEL ACCURACY AFTER FINE-TUNING ON EACH DATASET WITH DATA AUGMENTATION.

Dataset	No crop (%)	Waist crop (%)	Right crop (%)	Left crop (%)
MMIT	**49.94**	44.10	45.78	45.40
HMDB51	**79.75**	65.03	71.78	62.58
Hollywood	**69.39**	61.22	55.10	55.10

Comparing Tables V and VII, we notice that the fine-tuned model performs worse on most of the datasets, which could mean that training with the full frames yields better performance than training with frames missing crucial information.

C. *Input frame rate*

Tables VIII, IX and X show the accuracy of the model using the MMIT, HMDB51 and Hollywood dataset, respectively in the presence of different input frame rates. We can observe that the model appears to have the best performance when it is trained and tested with the same frame rate. When the training frame rate is different from the testing frame rate, there is a slight performance decrease, which means that the model should be trained with a frame rate similar to the frame rate provided by the final system. We can also note that higher frame rates generally lead to higher accuracies, as the model is able to extract more precise information about the movement, particularly for faster actions. The performance increase from using a higher frame rate is not very large, which means that, in a system with limited computational resources, it might be advantageous to have a 2% to 5% decrease in accuracy for a boost in computational performance.

TABLE VII
MODEL ACCURACY AFTER FINE-TUNING ON MMIT WITH DATA AUGMENTATION.

Dataset	No crop (%)	Waist crop (%)	Right crop (%)	Left crop (%)
MMIT	**49.94**	44.10	45.78	45.40
HMDB51	**69.16**	57.25	59.46	59.58
Hollywood	46.91	**51.03**	42.39	44.86
MSRDailyAct3D	51.67	**71.67**	53.33	60.00
UWA3D Mult	65.42	**81.31**	71.96	74.77
SBU	**49.00**	41.00	43.00	42.00

TABLE VIII
MODEL ACCURACY AFTER TRAINING WITH THE MMIT DATASET USING VARIOUS FRAME RATES.

Testing Training	1 fps	2 fps	4 fps	8 fps
1 fps	**48.64**	41.76	41.89	36.84
2 fps	50.19	51.49	**52.14**	50.97
4 fps	51.23	49.94	**53.44**	53.18
8 fps	50.84	49.94	53.31	**54.22**

TABLE IX
MODEL ACCURACY AFTER TRAINING WITH THE HMDB51 DATASET USING VARIOUS FRAME RATES.

Testing Training	1 fps	2 fps	4 fps	8 fps
1 fps	**79.51**	75.77	78.22	73.93
2 fps	76.87	80.80	**84.60**	**84.60**
4 fps	77.06	78.83	**86.81**	86.50
8 fps	73.93	78.83	84.72	**85.21**

Tables XI, XII and XIII show the best accuracies achieved in every class with different frame rates when testing the model with the MMIT, HMDB51 and Hollywood dataset, respectively. The results also suggest that in most cases, higher frame rates lead to higher accuracies. However, there are a few classes that are outliers, such as singing, kissing, celebrating, laughing, and eating; these classes generally contain slow movements, which means that using a high frame rate does not give the model any additional information, making them more invariant to the used frame rate than classes with fast movements.

VI. CONCLUSION

This paper focused on the recognition of activities in an in-vehicle setting, measuring the performance of the model under different scenarios. Given the unavailability of data for the specific in-vehicle environment, we collected a set of publicly available action recognition datasets, and chose 21 classes that we found relevant for our scenario.

Several experiments were conducted, considering different input frame rates, resolutions, aspect ratios, as well as obstructions of the field of view.

Results showed that the accuracy of the model decreased slightly when training and testing on different resolutions. Moreover, it depicted good accuracy on the datasets it was not

TABLE X
MODEL ACCURACY AFTER TRAINING WITH THE HOLLYWOOD DATASET USING VARIOUS FRAME RATES.

Testing Training	1 fps	2 fps	4 fps	8 fps
1 fps	**67.35**	59.18	61.22	51.02
2 fps	53.06	63.27	63.27	**67.35**
4 fps	63.27	**65.31**	**65.31**	**65.31**
8 fps	61.22	51.02	51.02	**75.51**

TABLE XI
THE BEST ACCURACIES ACHIEVED IN EVERY CLASS OF THE MMIT DATASET USING DIFFERENT FRAME RATES.

Class	1 fps	2 fps	4 fps	8 fps
fighting	45.00	**75.00**	60.00	55.00
punching	75.68	75.67	**83.78**	**83.78**
pushing	46.15	46.15	38.46	**53.85**
sitting	30.00	26.67	30.00	**36.67**
sleeping	56.84	49.47	**60.00**	53.68
coughing	21.43	28.57	21.43	**35.71**
singing	**76.17**	73.36	71.50	73.36
speaking	57.14	53.57	57.14	**64.29**
discussing	29.16	26.38	**33.33**	**33.33**
pulling	45.45	59.09	54.55	**63.64**
slapping	25.80	32.26	**35.48**	25.81
hugging	82.35	82.35	**88.23**	82.35
kissing	9.09	**27.27**	**27.27**	18.18
reading	14.29	14.28	**28.57**	**28.57**
telephoning	45.45	45.45	**54.55**	45.45
studying	64.71	58.82	**82.35**	58.82
socializing	15.00	20.00	20.00	**25.00**
resting	40.00	53.33	**66.67**	53.33
celebrating	**70.83**	**70.83**	**70.83**	66.66
laughing	**20.00**	**20.00**	**20.00**	**20.00**
eating	57.14	**71.43**	57.14	57.14

TABLE XII
THE BEST ACCURACIES ACHIEVED IN EVERY CLASS OF THE HMDB51 DATASET USING DIFFERENT FRAME RATES.

Class	1 fps	2 fps	4 fps	8 fps
punching	73.07	**84.61**	79.23	81.54
pushing	**92.59**	91.85	**92.59**	**92.59**
speaking	67.19	67.19	77.19	**83.75**
hugging	84.29	95.24	96.67	**100.00**
kissing	96.66	**100.00**	98.67	92.67
laughing	90.83	90.00	**94.58**	92.08
eating	71.66	78.33	87.77	**87.78**

trained on. After fine-tuning the model with other datasets, a considerable increase in accuracy was observed, since the model adapted to the new data. Measuring the computation time it was possible to conclude that using lower resolutions enables a significant increase in the number of frames that the model can process each second with only a minor impact on accuracy.

Removing part of the visual information caused a slight decrease in performance for most datasets. Fine tuning the model using data augmented with different types of cropping overall resulted in worse performance, suggesting that training with full frames is better than training with frames missing crucial information.

TABLE XIII
THE BEST ACCURACIES ACHIEVED IN EVERY CLASS OF THE HOLLYWOOD DATASET USING DIFFERENT FRAME RATES.

Class	1 fps	2 fps	4 fps	8 fps
hugging	**45.45**	9.09	36.36	27.27
kissing	87.50	95.83	95.83	**100.00**
telephoning	71.43	57.14	71.43	**85.71**

Using different frame rates in training and testing could be useful. However, the results indicate that the model had the highest accuracy when training and testing on the same frame rates. This suggests that the model should be trained with a frame rate similar to the frame rate provided by the final system. We can see that higher frame rates generally lead to higher accuracy, since the model is presented with more precise information about the movement. The increase in accuracy from using higher frame rates is not very high, therefore using lower frame rates might be advantageous in systems with limited computational resources. Additionally, we can see that classes that contain slow movements (e.g. singing, kissing) do not see performance gains when using higher frame rates, as the model is not being presented with any additional information.

References

[1] P. Augusto, J. S. Cardoso, and J. Fonseca, "Automotive interior sensing - towards a synergetic approach between anomaly detection and action recognition strategies," in *Fourth IEEE International Conference on Image Processing, Applications and Systems (IPAS 2020)*, Genova, Italy, Dec. 2020, pp. 162–167.

[2] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," *CoRR*, vol. abs/1705.07750, 2017. [Online]. Available: http://arxiv.org/abs/1705.07750

[3] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: generic features for video analysis," *CoRR*, vol. abs/1412.0767, 2014. [Online]. Available: http://arxiv.org/abs/1412.0767

[4] M. E. Kalfaoglu, S. Kalkan, and A. A. Alatan, "Late temporal modeling in 3d cnn architectures with bert for action recognition," 2020.

[5] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "Multi-fiber networks for video recognition," *CoRR*, vol. abs/1807.11195, 2018. [Online]. Available: http://arxiv.org/abs/1807.11195

[6] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," *CoRR*, vol. abs/1812.03982, 2018. [Online]. Available: http://arxiv.org/abs/1812.03982

[7] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" *CoRR*, vol. abs/1711.09577, 2017. [Online]. Available: http://arxiv.org/abs/1711.09577

[8] A. J. Piergiovanni, A. Angelova, A. Toshev, and M. S. Ryoo, "Evolving space-time neural architectures for videos," *CoRR*, vol. abs/1811.10636, 2018. [Online]. Available: http://arxiv.org/abs/1811.10636

[9] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," *CoRR*, vol. abs/1904.02811, 2019. [Online]. Available: http://arxiv.org/abs/1904.02811

[10] 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, 2017. [Online]. Available: http://arxiv.org/abs/1711.11248

[11] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning for video understanding," *CoRR*, vol. abs/1712.04851, 2017. [Online]. Available: http://arxiv.org/abs/1712.04851

[12] 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 *Fourth IEEE International Conference on Image Processing, Applications and Systems (IPAS 2020)*, Genova, Italy, Dec. 2020, pp. 114–119.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778.

[14] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfruend, C. Vondrick *et al.*, "Moments in time dataset: one million videos for event understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–8, 2019.

[15] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB: a large video database for human motion recognition,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[16] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[17] M. Marszałek, I. Laptev, and C. Schmid, “Actions in context,” in *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[18] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Mining actionlet ensemble for action recognition with depth cameras,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1290–1297.

[19] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. S. Mian, “Histogram of oriented principal components for cross-view action recognition,” *CoRR*, vol. abs/1409.6813, 2014. [Online]. Available: http://arxiv.org/abs/1409.6813

[20] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, “Two-person interaction detection using body-pose features and multiple instance learning,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012.

[21] G. Moon, J. Y. Chang, and K. M. Lee, “Camera distance-aware top-down approach for 3d multi-person pose estimation from a single RGB image,” *CoRR*, vol. abs/1907.11346, 2019. [Online]. Available: http://arxiv.org/abs/1907.11346