Automatic Detection of Certain Unwanted Driver Behavior

Boris Faizov[1], Vlad Shakhuro[1] and Anton Konushin[1,2]

[1]*Lomonosov Moscow State University, Leninskiye Gory, 1, Moscow, 119991, Russia*
[2]*NRU Higher School of Economics, Pokrovsky Bulvar, 11, Moscow, 109028, Russia*

Abstract

This work is devoted to the automatic detection of unwanted driver behavior such as smoking, using a mobile phone, and eating. The various existing datasets are practically unsuitable for this task. We did not find suitable training data with RGB video sequences shot from the position of the inner mirror. So we investigated the possibility of training the algorithms for this task on an out-of-domain set of people faces images. We also filmed our own test video sequence in a car to test the algorithms. We investigated different existing algorithms working both with one frame and with video sequences and conducted an experimental comparison of them. The availability of temporal information improved quality. Another important aspect is metrics for assessing the quality of the resulting system. We showed that experimental evaluation in this task should be performed on the entire video sequences. We proposed an algorithm for detecting undesirable driver actions and showed its effectiveness.

Keywords

Computer Vision, Action detection, Action classification, Driver distraction, Domain adaptation

1. Introduction

A large number of road accidents occur because the driver is distracted while driving. Different countries have fines for such violations, but they are very difficult to track. Modern computer vision methods can solve this problem. Nowadays state cameras automatically fine only for easy-to-track violations. Improving drivers' actions controlling is the next step in building such systems. Also, it can be used in monitoring car-sharing and taxi drivers as they often employ a large number of unqualified persons who often break the rules. Examples of distraction actions which we focused on in our work are: talking on the phone, using a smartphone, eating, smoking.

For such a system to work, a video camera must be installed in the cab of the car, which will record the actions of the driver. The most convenient location seems to us near the inner mirror, which will shoot the driver from the front-top. However, we did not find any publically available datasets with RGB images from the specified camera position. Moreover, we did not find datasets in which all the actions that we focused on in this work would be presented. In our

GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27–30, 2021, Nizhny Novgorod, Russia

boris.faizov@graphics.cs.msu.ru (B. Faizov); vlad.shakhuro@graphics.cs.msu.ru (V. Shakhuro); anton.konushin@graphics.cs.msu.ru (A. Konushin)
0000-0003-4961-3820 (B. Faizov); 0000-0002-1586-9257 (V. Shakhuro); 0000-0002-6152-0021 (A. Konushin)

CEUR Workshop Proceedings (CEUR-WS.org)

experiments we tried these sets: State-Farm's competition [1], Drive&Act [2], out-of-domain set of people faces images provided by "Tevian" [3] company. To test the algorithm, we filmed in a car our own video sequence. In it, we alternated performing distractions with normal driving.

Also, such a system must be effective to work in real-time. We tried different methods. To classify by one frame, we used ResNet-50 [4] and MobileNetV2 [5]. After frame-by-frame classification to improve accuracy we proposed an algorithm for the aggregation of consecutive frames. To process video fragments, we tried: a complex Inflated 3D Model [6] architecture and a simple model in which the frame-by-frame outputs of the convolutional backbone are aggregated by LSTM or a fully connected layer. In our experiments, we showed that 5 frames per second is enough speed for such system.

Another important aspect that we investigated is metrics for assessing the quality of the system. In our task we need to determine what action is currently taking place, if necessary, taking into account past frames. So we decided that our task is similar to online action detection. In practice, quality is usually assessed either on separate frames from the test sample or on already cut video fragments with known boundaries of actions. We concluded that experimental evaluation of such methods should be performed on the entire video sequence. We don't have to detect distraction on each separate frame and it is enough to find distraction in at least one frame. It is also important to monitor the false positive rate and per-class recall.

To conclude our paper we proposed a method for car driver's action classification. This method evaluates separate frames and then aggregates temporal information over the last frames. We showed that data from other domains could be successfully used for training such an algorithm.

2. Related work

The approaches for solving this problem could be divided into frame-by-frame classification and video action detection.

Most of the papers and techniques for this task work with a single frame. Conventional convolution neural network architectures such as VGG[7], ResNet[4], MobileNet [5] etc. can be used for the classification of driver's images. Simple specialized CNN architecture was proposed in [8]. They also collected two sets of data to account for the poor illuminations and different road conditions. The evaluation was performed both on their new set and on (SEU) dataset [9]. Another fast CNN model used Principal Component Analysis (PCA) technology to whiten the driver's image in [10]. To reduce computational complexity and memory requirements for VGG [7] network researchers introduce modifications in its architecture in [11] specifically for the task of driver monitoring. Several papers [12, 13] used a genetically weighted ensemble of convolutional neural networks with different CNN's classifying different cropped body parts

Another approach is to use video sequence classification methods such as [14, 15, 6]. However, these approaches contain 3D-convolutions and have too high computational complexity for practical use in real-time systems. But it is possible to aggregate information extracted by conventional CNNs from the last few frames. For example, a model can submit the extracted features to the LSTM [16, 17]. Also, SoTA-models [6] often analyze the optical flow in parallel with RGB-frames. Optical flow represents the motion direction of each pixel between two image frames. It is a powerful idea and it has been used to improve accuracy when classifying videos. It helps algorithms to prioritize motion as a key characteristic of the scene. But in a real embedded system calculating the optical flow can be too slow. In [2] the authors applied the video models to their new dataset and got the best quality with the Inflated 3D Model (I3D) [6]. Authors of I3D architecture suggested converting successful 2D image classification models into 3D ConvNets. They used the Inception-V1 [18] model and inflated all the filters and pooling kernels – endowing them with an additional temporal dimension. Parameters also were bootstrapped from the pre-trained ImageNet model.

3. Datasets

Collecting video sequences for this task is challenging. Firstly, it is unsafe to collect completely real data in which a person is really driving a car and is asked to perform distractions. This approach can result in an emergency situation. Secondly, the collected data should be representative and contain a large number of people. The models shouldn't learn to work only on a few specific persons.

There are several publically available datasets for driver action recognition:

1. State-Farm Kaggle competition dataset [1]: 26 participants, 10 classes, 22424 training images. For safety reasons the truck was dragging the car around on the streets — so these drivers weren't really driving. Unfortunately, the usage of StateFarm's dataset is limited to the purposes of the competition. Frame examples are shown in figure 4(a).
2. Distracted Driver Dataset [12]: 44 participants, 10 classes, 14478 frames. It was inspired by State-Farm's dataset and has similar classes.
3. Drive&Act [2] dataset: 15 participants, hierarchical annotation scheme with 83 fine-grained categories. Unlike previous datasets, it consists of video sequences which is much useful in a real scenario. Every person was asked to perform twice a pre-defined sequence of actions. Set has RGB and depth images from a side view, 3D body poses information from front-top view, and infrared images from six different views. Frame examples are shown in figure 1.

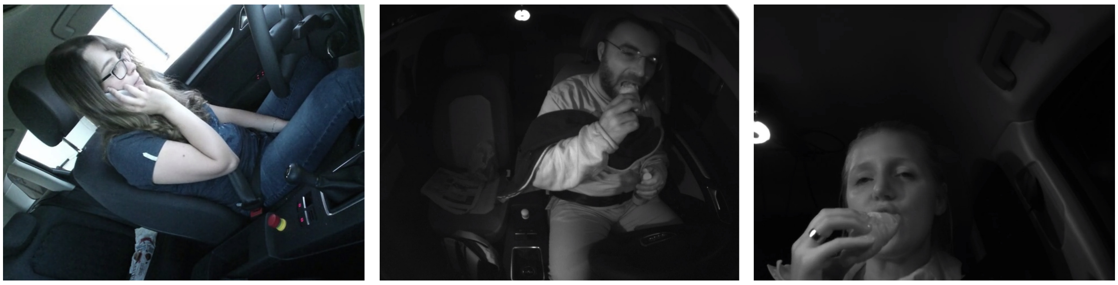

Figure 1: Examples of images from Drive&Act [2] dataset. Left to right: RGB side view; Infrared front-top view; Infrared steering wheel view.

There are many different kinds of distractions. Some actions of interest may be missing. By mixing different datasets, the model can learn to identify different sets of classes by environment and driver, rather than by their actions. Therefore, with a new class, it might be necessary to collect the entire dataset again.

In our experiments, we used proprietary data provided by "Tevian" [3] company. It has RGB images of neutral, smoking, and talking on the phone people. In the experiments, we divided the sample into training (75%) and validation (25%) so that the proportions of the classes were preserved. We needed these data to train our model find people who smoke while driving because there is no such class in other datasets. Examples of frames are in figure 2. The big disadvantage of this set in this task is that we have to cut out the faces of people in the car. This could lead to the loss of contextual information in the frame and deterioration of the possible quality.

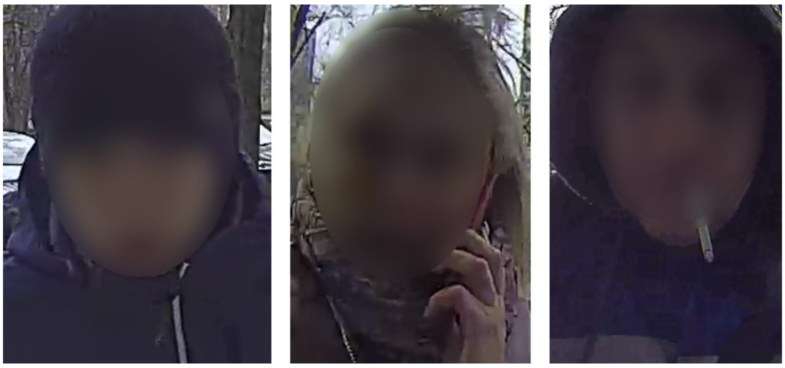

Figure 2: Examples of people images. For privacy reasons, people's faces were cut out from examples and blurred. Left to right: Doing nothing; Phone talking; Smoking.

We also filmed our own test video sequence. For safety, the car wasn't moving during the shooting. It presents classes for safe driving, eating (food, drinks), telephone (texting, calls), smoking. The length of the sequence is 14 minutes. Examples of frames are shown in figure 3.

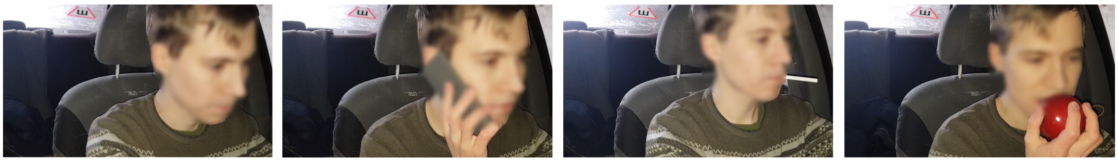

Figure 3: Examples of images from our own test sequence. Left to right: Doing nothing; Phone talking; Pretend smoking, Eating.

Drivers in [1, 12, 9] and RGB-images from Drive&Act [2] are photographed from the side view. It seems to us that this choice is not very successful, since in a real system (especially in a taxi car) the camera with such an arrangement can be obstructed by a neighboring passenger or luggage.

4. Proposed method

Low-performance models such as Inflated 3D Model can be used if training takes place on labeled video sequences. We decided to try a more efficient approach. We propose to first train a frame-by-frame classifier of driver actions based on MobileNetV2. We took the network pre-trained on the ImageNet and fine-tuned it on the training set images. Further, sequential frames may be independently classified and then aggregated with an additional neural network. We considered such frame aggregational neural networks:

1. A single-layer perceptron that accepts concatenated frame attributes as input and outputs an action class.
2. Recurrent LSTM network with 128 features.
3. Transformer [19] network with 4 layers, 2 heads and 128 features.

When training on a frame-by-frame dataset, there is no way to train a video model. But to improve the quality of the method in the real-time system, we still need to aggregate the results of the last few frames. For this, we have proposed our method:

1. Consider that the system has a speed equal to 5 frames per second.
2. The system classifies each frame independently of the others.
3. The last 10 frames are considered. If at least 8 of them were classified as some class, which is not safe driving, then we consider that we have detected a distraction at the moment. Otherwise, we consider the current moment of time to be safe driving.

5. Metrics

We used several metrics to measure the performance of different approaches. For per-frame classification we used:

- Accuracy — the ratio of correctly classified images to the total number of images.
- False positives — the ratio of safe driving images, classified as driver distractions to the total number of safe driving images.
- Per-class recall.

For actions in video sequences, we used another set of metrics. By action, we mean a set of frames with one separate video event as a whole. The system does not need to be triggered at every point in time during a distraction event. It is enough to find at least one moment with each action so that the driver could be fined. We need to take into consideration that each frame has previous frames and we can use this data. So metrics should be calculated after aggregation of the last few frames. Also, we need to have metrics that combine information from an entire video.

- Accuracy on frames after aggregation with previous frames.
- False positive on frames after aggregation with previous frames.
- Per-class recall on frames after aggregation with previous frames.

- False positive on actions. This means that for each ground truth action we collect a set of all triggered classes during this action. Then if this action is safe driving, but we have distracted class in a set of triggered classes, then this action is a false positive.
- Per-class recall on actions. This is similar to the previous metric, but we check if our model triggered a real class of each action.
- Per-class mAp. First, the frames are ranked according to their confidence from high to low. $Precision@k = \frac{TP(k)}{TP(k)+FP(k)}$, where $TP(k)$ and $FP(k)$ are the number of true positives and false positives accordingly at the first k frames. The average precision of a class is then defined as $AP = \sum_k \frac{Precision@k*\mathbb{1}[frame\ k\ is\ TP]}{total\ positive\ frames}$. The mean of the AP over all classes is final mAP.

When testing, we took into account that it is possible to trigger an action in the range of $+-40$ frames from it. This type of misclassification was still considered correct.

6. Results

6.1. Experiments with State-Farm dataset

We tried neural network models on State-Farm competition [1] dataset. Among the conventional models we have chosen ResNet-50 [4] and MobileNetV2 [5]. Having studied the existing solutions, we decided to try not only RGB pictures, but also the segmentation of people in the frames. The segmentation model was similar to the RGB model, segmentation masks were passed to the input instead of color images. We also tried to concatenate outputs of RGB and segmentation models and add one fully connected layer on top of these features to obtain classifications. The segmentation was obtained using the human parsing model [20]. Examples of frames and segmentation are shown in figure 4(a,b). In addition to the usual augmentations (turns, crops, noise) we used augmentation where two different pictures of the same class are mixed, an example in figure 4(c). This additional augmentation improved the quality. It seems to us that this happened because such transformation prevented the model from remembering specific people and there were not enough different persons in the frames of the original training set. However, we noticed that marked and test sets have subsets of the same people in images. Therefore, we divided the labeled sample into training and validation sets according to people identifiers: 19 persons in the training set, 26 in the validation set. The results are shown in table 1. The operating time here and further was measured on the Intel Core I7-3770K CPU processor.

6.2. Basic experiments on Drive&Act

6.2.1. Inflated 3D Model.

To begin with, we tried to reproduce the results of paper [2] for the classification of fine-grained activities of 34 classes by Inflated 3D Model with frame resolution 224×224. The model was trained separately on RGB images from side view and optical flow. When training an RGB stream, we took 36 frames with a stride of 2 frames, and when training an optical flow stream, we take 18 frames with a stride of 2 frames. During testing, all frames of a video fragment are

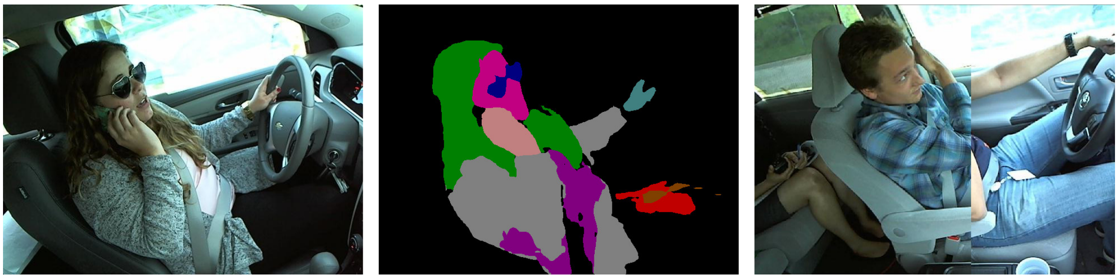

Figure 4: Examples of images from State-Farm [1] dataset. Left to right: (a) Original RGB; (b) Image segmentation; (c) Mix of pictures from the same class.

Table 1
Results on State-Farm [1]

Method	Private score	Accuracy on validation	Frames/sec on CPU
ResNet-50 RGB	0.4180	81.14	7
ResNet-50 Segmentations	0.4231	94.34	2
ResNet-50 RGB+Segmentations	0.3425	95.83	1.5
MobleNetV2 RGB	0.4342	76.95	14

Table 2
Reproduced results of paper [2]

Method	Average per-class accuracy	Frames/sec on GPU	Frames/sec on CPU
Our on RGB images	53.15	760	12.5
Our on optical flow	68.21	870	15.5
Our on RGB+optical flow	68.97	-	-
Paper Drive&Act	69.57	-	-

taken. The results are shown in table 2. Our average per-class accuracy on validation 68.97 has almost reached the value from the original paper 69.57. But in a low-performance real-time system, it is unlikely that it will be possible to calculate the optical flow.

6.2.2. Three selected classes.

In our experiments, we decided to focus on three important activities: using a smartphone (talking on the phone, interacting with phone), food (eating, drinking), safe driving (all other classes). In addition to the Inflated 3D Model model, we decided to train the frame-by-frame model with MobileNetV2 architecture. We also tried to train MobileNetV2 only on cropped heads. Heads were cropped from segmentations obtained with human parsing model [20]. The results are shown in the table 3.

Table 3
Results on Drive&Act with only three selected classes.

Method	Accuracy	False positives	Per-class recall
Inflated 3D Model	90.63	2.92%	-
MobileNetV2	88.75	0.53%	safe: 99.47 phone: 28.02 eating: 25.98
MobileNetV2 only on heads	86.16	1.18%	safe: 98.82 phone: 15.09 eating: 11.56

6.3. Experiments on Drive&Act

In this section, we focused on results on three selected classes of the Drive&Act dataset (safe driving, eating, phone). The best results when training and testing on the Drive&Act set were obtained with the following training setup:

1. Convert original markup that was discontinuous, merging consecutive clips of similar classes.
2. Training a frame-by-frame classifier for all 34 classes on full (not cropped)[522] frames.
3. Extraction of frame features from the outputs of the penultimate layer of the classifier.
4. Training LSTM network / One-layer linear perceptron / small transformer on 3 selected classes of interest with zero class weight equal to 4.0. The network will classify by the last 10 frames with FPS=5.

All consecutive frames with the preceding ones in the video are viewed independently, therefore, there are as many triggers in the confusion matrix as there are frames in the video.

Results are shown in table 4. Best of all, in our opinion, is the LSTM network, because it has a low false-positive rate and good recall/accuracy. The problem with the transformer was that it was heavily overfitted due to a lack of data. We tried to reduce the number of transformer layers or the dimension of the features, but we were unable to improve the quality. Maybe transformers don't fit well for this particular dataset. We also investigated frames, where the LSTM model had false positives and concluded that they were normal in most cases. They occurred mostly in boundaries between actions in the dataset due to the not accurate markup. Aggregating frames without neural networks reduce the false-positive rate, but it worsens recall on distraction classes.

Learning only on people's heads worked much worse. It seems to us that this is because the context is not visible.

6.4. Experiments on faces images and our test images

Our test video consists of RGB images from the front-top view. But in Drive&Act videos from the front-top view are infrared. Therefore, this data is not suitable for testing our method. Data with face images provided by "Tevian"[3] company is more acceptable — these are RGB pictures with people's faces and three classes: nothing, phone, smoking. Therefore, we trained

Table 4
Experiments on Drive&Act videos with different heads on top of MobileNetV2 backbone.

Method	Accuracy on aggregated frames	False positive on aggregated frames	Per-class recall on aggregated frames
LSTM	90.15	0.08%	safe: 99.92 phone: 43.02 eating: 26.85
Linear perceptron	90.45	0.13%	safe: 99.87 phone: 39.07 eating: 33.85
Transformer	89.94	0.51%	safe: 99.49 phone: 36.53 eating: 33.46
Aggregating frames by last 10 frames without NN	87.33	0.32%	safe: 99.68 phone: 7.14 eating: 9.59

Table 5
Final results on frames with faces and on frames from our test video if MobileNetV2 model trained on frames with people faces.

Experiment	Accuracy	False positive	Per-class recall	mAp
On faces data	98.18	0.81%	safe: 99.19 smoking: 50.47 phone: 78.59	91.68
On our test frames from video	85.52	0.79%	safe: 99.21 smoking: 14.14 phone: 35.46	90.50

the MobileNetV2 classifier on people faces. Then we applied it to our test video. In this case, we need to cut out the head of a person. We again did this using the human parsing model [20]. The results obtained during frame-by-frame testing are in table 5. And the results on our test video are in table 6. Our result on the test video still contained 10 false-positive frames after aggregation of the last 10 frames, but they all were just before the person in the video started simulating smoking. Per-frame recall values may seem low, but metrics on action level are better. Even if we didn't trigger on a class at every time moment of action, we still found 2 of 5 smoking acts and 4 of 11 phone using acts.

7. Conclusion

In this paper, we considered the problem of recognizing distracted drivers. We proposed several methods for driver's action classification using frames and video sequences. In the absence of video sequences suitable for the training set, we proposed a method for aggregating the last frames of a video sequence and experimentally demonstrated its effectiveness. We have

Table 6
Final results on our test video with aggregation of frames with MobileNetV2 model trained on frames with people faces.

Exp.	Accuracy on aggregated	False positive on aggregated	Per-class recall on aggregated frames	False positive on actions	Per-class recall on actions
Test video	87.75	0.04%	safe: 99.96 smoking: 8.27 phone: 23.88	2.56	safe: 97.44 smoking: 40.00 phone: 36.36

shown that the driver distraction classification problem can be solved using out-of-domain training data. The proposed model was trained on photographs of people's faces and showed high quality on our test video sequence.

References

[1] State Farm Distracted Driver Detection, State farm distracted driver detection, 2016. URL: https://www.kaggle.com/c/state-farm-distracted-driver-detection/overview.

[2] M. Martin, A. Roitberg, M. Haurilet, M. Horne, S. Reiß, M. Voit, R. Stiefelhagen, Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2801–2810.

[3] Tevian, Tevian, 2021. URL: https://tevian.ai/.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.

[6] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.

[7] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[8] C. Yan, F. Coenen, B. Zhang, Driving posture recognition by convolutional neural networks, IET Computer Vision 10 (2016) 103–114.

[9] C. Zhao, B. Zhang, J. He, J. Lian, Recognition of driving postures by contourlet transform and random forests, IET Intelligent Transport Systems 6 (2012) 161–168.

[10] X. Rao, F. Lin, Z. Chen, J. Zhao, Distracted driving recognition method based on deep convolutional neural network, Journal of Ambient Intelligence and Humanized Computing (2019) 1–8.

[11] B. Baheti, S. Gajre, S. Talbar, Detection of distracted driver using convolutional neural net-

work, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2018, pp. 1032–1038.

[12] Y. Abouelnaga, H. M. Eraqi, M. N. Moustafa, Real-time distracted driver posture classification, arXiv preprint arXiv:1706.09498 (2017).

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

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 4489–4497.

[15] Z. Qiu, T. Yao, T. Mei, Learning spatio-temporal representation with pseudo-3d residual networks, in: proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5533–5541.

[16] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, L. Fei-Fei, Every moment counts: Dense detailed labeling of actions in complex videos, International Journal of Computer Vision 126 (2018) 375–389.

[17] N. Srivastava, E. Mansimov, R. Salakhudinov, Unsupervised learning of video representations using lstms, in: International conference on machine learning, PMLR, 2015, pp. 843–852.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, arXiv preprint arXiv:1706.03762 (2017).

[20] P. Li, Y. Xu, Y. Wei, Y. Yang, Self-correction for human parsing, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).