Driver Activity Recognition by fusing Multi-object and Key Points Detection

Pablo Pardo-Decimavilla, Luis M. Bergasa, Elena López-Guillén, Ángel Llamazares, Navil Abdeselam and Manuel Ocaña

RobeSafe research group, Electronics Department, University of Alcalá, Spain
pablo.pardod@edu.uah.es, {luism.bergasa, elena.lopezg, angel.llamazares, navil.abdeselam, manuel.ocanna}@uah.es

Abstract. Driver distraction recognition plays a fundamental role in road safety. In this paper, we present a modular architecture based on the fusion of key points and object detection for predicting driver's actions. From multi-camera infrared recordings, we will temporarily detect among a variety of actions that lead to distractions. Our system detects objects of interest and extracts key points from the driver. They are merged by generating features that relate them and processed with a ML-based classification algorithm. Finally, filters are applied to reduce bounces and add temporal context to the detections. Our proposal has been validated on two state-of-the-art datasets for driving distractions. Through several experiments we show that fusion substantially improves related action inference and improves domain adaptation. In addition, our framework is lightweight, explainable and has a low latency as it performs frame-by-frame inference. The modularity of the network allows us to upgrade parts independently or eliminate a camera without having to modify the entire network.

Keywords: advanced driver distraction detection, object detection, pose estimation

1 Introduction

In 2021 there were 921 fatal accidents on Spanish roads in which 1,004 people died and 3,728 were seriously injured [1]. During this year, 19,800 people were killed in road crashes in Europe [2]. One in three road accidents is due to inattentive driving, such as using a mobile phone, manipulating the navigator, eating, smoking, or due to fatigue or stress. Many lives could be saved if distractions were reduced.

This project will focus on monitoring and inferring different actions that directly produce distractions while driving. The driver will be alerted to stop the action and even in autonomous vehicles it will be possible to take control of the car to avoid an accident.

Based on the KNDAR project [19], we have built an architecture that fuses multi-class object detection and driver key points to infer actions. It is a modular

architecture that processes in real time three images corresponding to three different points of view of the driver. As shown in Figure 1, the first step is to identify the driver and key objects that are important for certain actions. Then the key points of the previously detected driver are inferred. With all this information we build features that relate both the key points and the important objects in the scene. Then, through a ML algorithm each frame is classified with the most likely action to finally filter it and get the start and end time of each detected action. As we will see below, the algorithm will be trained and evaluated on two state-of-the-art datasets for driving distractions.

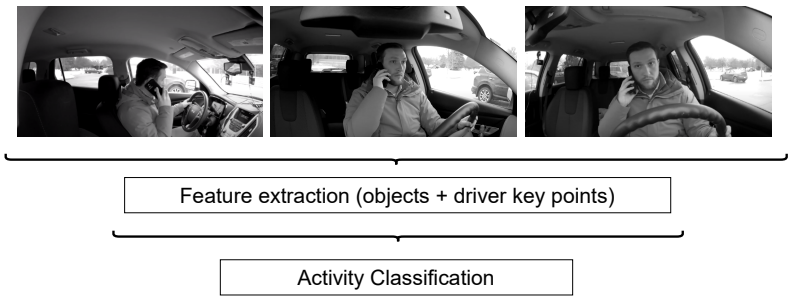

Fig. 1: Graphical abstract.

Our major contributions are, on the one hand, to improve notoriously the actions recognition related to some object (e.g., Phone Call, Text) regarding our baseline (KNDAR). By merging the object detection with the driver key points we improve the performance of these actions and therefore the overall performance, hardly increasing the computational cost. Additionally, through different experiments we found that our network performs better on other datasets on which it has not been trained, reaching a higher generalization capability. On the other hand, we improve the final filtering stage. By doing the inference frame by frame, as in KNDAR, temporal context is lost, being the system more susceptible to bounces. We treat the time probability of each of the actions as temporal series. We apply different filters that reduce bounces and eliminate false negatives, adapting this algorithm to the nature of each action.

To properly explain the project, we first analyze the state of the art of some of the network components, then, we explain how we have integrated them into a single network, and finally we perform experiments and empirically test our proposal regarding other state-of-the-art approaches.

2 Related Works

This topic is relatively new and there are not many specific systems that solve this problem. A widely used solution is training or finetuning video understanding networks that are focused on characterizing the different actions within the

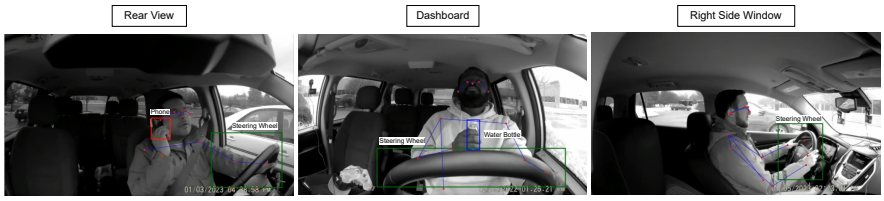

Fig. 2: Object detection and driver key points for the three viewpoints.

image [18]. They are very competitive but they are slow and need a significant length of video to achieve competitive results. Other works combine Convolutional Neural Networks with complex feature extractors and Autoenconders [12]. Approaches like KNDAR [19] use less compute-intensive features and use ML techniques to infer the action frame by frame giving superior performance with less information. On the contrary, they sacrifice temporal context but exhibit superior adaptation for real-time processing. We will focus on the state of the art of the two main stages of our network.

2.1 Persons and object detection

Multiclass detectors are a fundamental component of the state-of-the-art deep learning models for object detection tasks. These detectors are designed to recognize and localize multiple object categories simultaneously in an input image. They achieve this by leveraging advanced convolutional neural network architectures and sophisticated training strategies that enable them to learn complex patterns and relationships between objects and their surroundings. EfficientDet [17] is a family of object detection models that achieve state-of-the-art performance on the COCO dataset [14]. DETR [6] is a transformer-based object detection model that uses a set-based approach to predict the locations and classes of objects in an image. Cascade R-CNN [4] is an extension of the popular R-CNN object detection framework that achieves state-of-the-art performance too. YOLOv8 [10] is a lightweight and efficient object detection model that achieves state-of-the-art performance on several benchmark datasets. It is particularly effective at detecting small objects and can handle a large number of object classes. Today YOLO it is one of the most widely used and has an outstanding performance with a low number of parameters.

2.2 Pose Estimation

Pose estimation refers to the task of determining the spatial orientation and position of an object or a human body in an image or a video. For this task there are two types of approaches. On the one hand, bottom-up networks estimate the human body parts in the image followed by calculating the pose. On the other hand, a top-down approach localize the humans in the image and then estimate the parts followed by calculating the pose. In the previous stage we used a

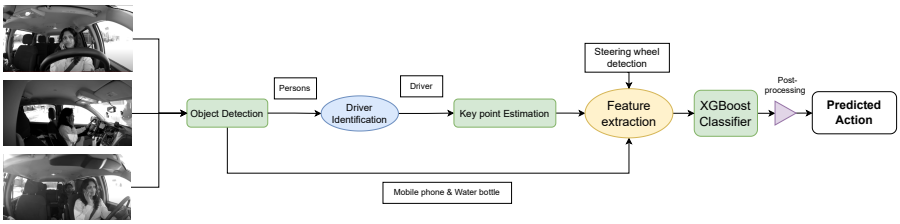

Fig. 3: Architecture integration.

detector to make the double detection of the driver and objects of interest, that is why we are interested in a pose estimation based on the second approach. AlphaPose [8] is a popular deep learning-based framework for multi-person pose estimation and, as we can see in Table 1, it offers superior performance compared to other networks in the state of the art.

Table 1: State-of-the-art pose estimators results on COCO dataset

Method	**AP @0.5:0.95**
OpenPose (CMU-Pose) [5]	61.8
Detectron (Mask R-CNN) [9]	67.0
KAPAO [15]	68.5
AlphaPose [8]	**73.3**

3 Architecture integration

So far we have focused on the two main components of our system. Next, we detail how they are integrated and how they feed each other, as shown in the Figure 3. This modular architecture brings explanability and the possibility of reusing detections and features by other networks.

3.1 Feature extraction and merging

The network starts with multi-object detection through YOLOv8-m. A batch of images equal in size to the number of cameras we have is the input to our system, from which the bounding boxes of the detected classes are obtained. We use the YOLO weights trained in COCO, which provide a total of 80 different detected objects. Of all of them we are interested in person (0), bottle (39) and mobile phone (67) classes. We filter the detection and keep the objects of interest applying non-maximum suppression to the detections. From all detected persons we have to choose the one that corresponds to the driver. We assume the car is driven on the left-hand side. From all the people detected, we will choose the one who is located in that area and whose bounding box is the largest to make sure that he is the one sitting in the front row of the vehicle, as his perspective will be greater.

Next, we generate the features that will feed the classification algorithm. We start from the features proposed in Table 2 of the KNDAR project [19]. Distances and angles between key points that interact or are related are calculated. In addition, the position shift with the previous frame is also added. On top of that, a binary category has been added for mobile phone detection to indicate its presence in the scene, in addition to the distance between the left wrist, right wrist, left ear and right ear to the center of the mobile phone bounding box. Note that if the detection module detects more than one cell phone it keeps the one closest to the driver that is the most likely to interact with it. The same applies to the water bottle that also has a feature that reports its existence and its distance from the driver's wrists and nose to the center of the object.

Additionally, as the camera does not move over time, we have manually integrated the position of the steering wheel that provides interesting information. For this case we introduce two categorical variables that indicate whether each of the driver's hands are inside the steering wheel's bounding box and the distance from it to each wrist. This potentially help reduce false positive detections and improve the global performance. We can see in Figure 4 the distances drawn on the image for both the mobile phone and the steering wheel. In total we have 207 features per frame, which corresponds to 69 per viewpoint.

Fig. 4: Distances from the center of the steering wheel and mobile phone to wrists and ears (only for mobile phone).

3.2 Activity Classifier

Once the features have been extracted from the frame we obtain the 16 probabilities for each of the classes, described in Table 2, which will later be filtered to obtain the start and end of the action by using the XGBoost Classifier. It is a popular machine learning algorithm that uses an ensemble of decision trees to make predictions by combining the outputs of individual trees. The algorithm works by iteratively building new decision trees that attempt to correct the errors made by the previous trees, with each new tree focusing on the samples that were misclassified by the previous trees. XGBoost Classifier is known for its ability to handle structured data and perform well on small-to-medium sized datasets. It can typically be trained faster and with fewer computational resources than deep learning models, making it a suitable choice for this task.

3.3 Postprocessing

Output Filtering Inferring an action frame by frame loses temporal context and produces a noisy output. To establish in which second the action begins and ends we use a simple filtering process. Our baseline [19] takes the most probable action in each frame. We have noticed that considering the confidence level of each action, rather than solely focusing on the most probable one within the frame, leads to significantly improved results. This is because it incorporates temporal information into the decision-making process. For each of the 16 distractions and one corresponding to the "no distraction", we have a stream of confidence probabilities. A low pass filter is applied to each one independently. This eliminates fast changes in probability. Unlike filters based on averages, this one totally eliminates actions considered as noise without modifying the probability of those that are not. In this way, the curve will be significantly smoothed and peaks corresponding to false positives in isolated frames are eliminated. Then, among the 17 probabilities in a frame, the highest of the 16 distractions are taken, as long as they exceed the minimum threshold. If none of them exceeds the threshold, the "no distraction" is established. In summary, this filter provides context by temporally relating the predictions of an action. Moreover, all this happens frame by frame without needing a video fragment.

Filter by length In addition to the filter an extra post-processing is applied to discard actions with very short duration. For this purpose, a study was made of the average and minimum duration of each studied action. Actions such as talking on the phone or chatting tend to last much longer than others such as picking something up off the floor. Therefore, applying the same minimum duration to each class will be misleading. This filter is applied to extract the duration of each detected action and if the duration is less than the one established for that class, it is discarded.

4 Experimental Evaluation

4.1 Dataset

To measure the performance of our network we take two datasets. The first one is used by our baseline, so we will be able to quantify the added improvements. The second one is a dataset more adapted to our network that let us to compare our project within the state of the art.

Dataset I Track 3 dataset of the AI City Challenge 2023 [16] provides videos of drivers performing the actions shown in Table 2 while simulating driving. The action is simultaneously recorded from three angles denoted as "dashboard", "rear view" and "right window view". The cameras are infrared to be able to cover the problem during the day and with total absence of light. It provides a total of 25 drivers of different ages, sexes and appearance for training and another

5 for validation (without labels). Six videos corresponding to two sequences and three point of views each are provided for each driver. One without objects in the face and head and another one with them, in order to difficult the inference. The ground truth provided contains the action performed in each video and the time in seconds of the beginning and end of each action.

Table 2: Driving Distractions

ID	Distraction	ID	Distraction
0	Normal Driving	8	Adjust control panel
1	Drinking	9	Pick up from floor (Driver)
2	Phone Call (Right)	10	Pick up from floor (Passenger)
3	Phone Call (Left)	11	Talk to passenger at the right
4	Eating	12	Talk to passenger at backseat
5	Text (Right)	13	Yawning
6	Text (Left)	14	Hand on head
7	Reaching behind	15	Singing or dancing with music

Dataset II Cairo Distracted Driver Dataset (AUC-DDD) [3] is one of the most widely-used driver distraction datasets. The dataset contains videos of 44 different drivers in five different cars performing distractions. It includes the same as in Dataset I with the removal of action 4, 9, 10, 13, 14, 15 and the addition of "Hair or makeup". The images are taken from the passenger door in RGB during the day. It also includes two versions, the D2.1, in which participants are in a real car, and the D2.2, in which they are in a simulator.

4.2 YOLO - Object Detection

One of the most used version of YOLO is v5, but with the release of v8 we made a comparison of its improvement for driver detection. We validated v3-spp, v5 and v8 in SViRO [7] and TiCAM [11], two datasets focused on detecting vehicle occupants. All the models of each version trained with COCO were tested on both datasets to find the one that suits our needs, by finding a compromise between performance and inference time. The best performing model is YOLOv8-m, which gets a 0.85 performance reducing the inference time to 5.7ms.

In addition to detecting people, we will detect objects of interest for the actions. For this challenge the two objects that are related to an action are the cell phone, that is presented in actions 2, 3, 4 and 5, and the bottle of water in action 1. For the eating action there is no visible food of interest. When using COCO-trained YOLO weights both people and objects are detectable classes.

In order to obtain better results, a finetuning of the detector was performed. The detection of people was improved with the above mentioned datasets. In this way person inference will be adapted to the environment of interest.

4.3 Metrics

In the 2023 edition of the challenge the average activity overlap score has been proposed as the main metric. Given a ground-truth with start time (gs) and end time (ge), the prediction that best fits within a temporal window of ±10s around is chosen. Being g the ground-truth and p the prediction, the following formula is defined:

$$os(p, g) = \frac{max(min(ge, pe) - max(gs, ps), 0)}{max(ge, pe) - min(gs, ps)} \tag{1}$$

The final score is the average overlap score among all matched and unmatched activities. In addition we have also considered using the F1-score, as it was the one used in the 2022 edition. It is the harmonic mean of precision and recall, and it ranges from 0 to 1, with higher values indicating superior performance. The formula is:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2}$$

where precision is the number of true positive predictions divided by the number of all positive predictions, and recall is the number of true positive predictions divided by the number of all actual positive instances. In summary, the F1 score provides a balanced measure of precision and recall, making it a useful metric for evaluating the performance of classifiers.

4.4 Ablation study

Our architecture supports a variable number of cameras. Each one provides valuable information for each action. This is why we have performed an ablation study by eliminating each camera to quantify the amount of information it provides. Table 3 shows the OS score with each camera pair evaluated in Dataset I. As we can see, the one that gives the most information is the dashboard camera, followed by the rear camera and finally the right camera.

Table 3: OS score for testing Dataset I. **C1**: Dashboard; **C2**: Rear; **C3**: Right;

C2 - C3	**C1 - C3**	**C1 - C2**	**All**
0.293	0.305	0.317	0.3605

4.5 Global performance

We show how the network improves class by class regarding KNDAR proposal (our baseline). We can see in Figure 5 a clear improvement in all actions of Dataset I. Above all, the most benefited classes are the static ones, that is,

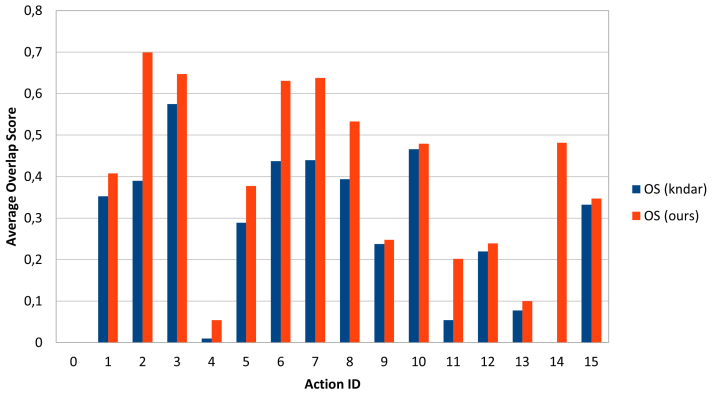

Fig. 5: Comparative histogram of the OS score in the different actions (shown in Table 2) between the baseline and our method.

those that do not need previous frames to determine the action. These achieve a performance in the state of the art exceeding 60% in many of them. We can see that this improvement is much higher for the actions where the object detection is more accurate. For Phone Call, as the phone is at the height of the driver's head, so there is no difficulty in detecting it as it is not hidden. For Text, as the phone is lower, it is hidden for some cameras and can be obscured by another passenger or an object in the vehicle. Therefore, the inference of these actions is intimately linked to the object detection. On the other hand, we can see a poorer performance in the classes that do need context. This is the case of singing or talking to passengers. These are actions that even for a human would be practically impossible to infer by looking at a single frame. They are also difficult to generalise as they are specific to each individual. All classes have benefited directly or indirectly from the hands-to-wheel distance feature, improving overall performance. This feature helps to locates the driver in space. If only the driver's key points are used we cannot know where exactly it is located inside the vehicle. On the one hand, many classes use these features to infer the action, notably those where hand position on the steering wheel is intimately related to it. On the other hand, it helps to reduce false positives or discard some actions.

We have run our experiments on a computer running Ubuntu Linux version 20.04 and equipped with 8-core Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz CPU and NVIDIA RTX 2080Ti GPU. The system with the three cameras is running at 15-17Hz, including object and key point detection and subsequent action inference and post-processing.

The overall performance of the network in Dataset I, following the official metric, for the public leaderboard was **0.3605**, achieving the 18th position of 27 participants (March 25th, 2023). As we mentioned before, our work was not focused on improving all classes, but only some of them. If we take those that are directly benefited by the merging of objects, we achieved a **0.5515**, placing it

among the top 10 in the public leaderboard. Also our baseline last year achieved a **0.2558** F1 score while ours achieves a **0.3766** in F1 score.

Table 4: OS score for relevant actions between the baseline and our proposal.

ID	Action	KNDAR	Ours	Delta (%)
1	Drinking	0.353	0.408	+15.60
2	Phone Call (Right)	0.390	0.699	+79.23
3	Phone Call (Left)	0.575	0.647	+12.52
5	Text (Right)	0.289	0.377	+30.42
6	Text (Left)	0.437	0.631	+44.43
	Mean	**0.267**	**0.380**	**+42.34**

In table 5, we can see a comparison of the performance and inference time of the state-of-the-art methods evaluated in Dataset II and extracted from [13]. Our method ranks among best state-of-the-art proposals by achieving low inference time and high performance. Furthermore, compared to other methods, our network is much more adaptable to other types of data, such as depth or infrared images. Furthermore, cameras can be added or removed in a simple way and new objects can be added for the detection of new classes.

Table 5: Different methods of classification of distractions validated in Dataset II. AA: Average Accuracy; AF: Average F1 Score; IT: Inference Time;

Model	AA	AF	IT(ms)
AlexNet	0.738	0.741	2.61
GWE-Resnet50	0.8169	NA	NA
VGG-19	0.833	0.835	20.46
ResNet50	0.877	0.882	14.26
InceptionV3-RNN	0.884	0.899	23.42
ResNet152	0.8852	NA	62
Densenet-201	0.890	0.895	46.05
GWE-InceptionV3	0.9006	NA	NA
InceptionV3-BiLSTM	0.917	0.931	23.30
InceptionV3-BiGRU	0.917	0.922	23.24
ResNet+HRNN+Inception	0.9236	NA	114
ResNet101 + ResNet50	0.9428	0.9427	668.20
Ours	**0.9075**	**0.9025**	**29.60**

As we can see in Table 6, we have evaluated the method with better results and similar inference time to ours in other public datasets. We trained both models in D2.1 and were evaluated in D2.2 and D1. We observe that our architecture generalises more effectively as it is independent of the type of person and car where it is located, outperforming in both datasets. We are less influenced by the perspective and position where the driver appears. Moreover, we are able to isolate and exclude passengers from detection. The domain adaptation of our

network is higher as the pose estimator and object detector is trained on a much wider variety of people than the above mentioned dataset. This serves as strong evidence of the effectiveness of our approach.

Table 6: Model generalisation comparison. AA: Average Accuracy.

Dataset	**InceptionV3**	**Ours**	**Delta (%)**
D2.2 (AA)	0.2198	0.3701	**+68.38**
D1 (AA)	0.2835	0.5138	**+81.23**

5 Conclusion

A modular and lightweight architecture for driver distraction recognition through the fusion of objects detection and driver key points is presented. Our system consists of three stages; firstly, we infer the people inside the vehicle (including the driver) and other objects of interest. Secondly, we infer the driver's key points. Thirdly, we extract features from all the data and make a prediction of the action through a ML method. Finally, we apply filters to determine the current action and its duration over time. We trained and validated the project in the Track 3 dataset of the AI City Challenge 2023 and Cairo Distracted Driver Dataset.

Fusion hardly increases the computational cost by reusing the detector for key point generation. A few features are added to the existing ones, but they have a great impact on performance, improving it significantly. The proposed filter is applied in real time and provides temporal context to the prediction, as it takes into account the confidence of previous frames. Our approach is markedly more appropriate for diverse environments and data types, outperforming in domain adaptation. Additionally, the architecture exhibits real-time performance and low prediction latency.

As a future improvement we plan adding tracking to the key points and give context to each inference. Enhancing object detection could also lead to improvements in actions associated with the detected objects.

Acknowledgements. This work has been supported from the Spanish PID2021-126623OB-I00 project, funded by MICIN/AEI and FEDER, the TED2021-130131A-I00, PDC2022-133470-I00 projects, funded by MICIN/AEI and the European Union NextGenerationEU/PRTR, and the collaboration scholarship for the 2022-2023 academic year (22C01/007899), financed by the Ministry of Education.

References

1. La moncloa. 07/01/2022. los accidentes de tráfico se cobraron la vida de 1.004 personas el pasado año [prensa/actualidad/interior]. (Accessed on 04/01/2023)
2. Preliminary 2021 eu road safety statistics. (Accessed on 04/04/2023)
3. Abouelnaga, Y., Eraqi, H.M., Moustafa, M.N.: Real-time distracted driver posture classification (2018)

4. Cai, Z., Vasconcelos, N.: Cascade r-cnn: high quality object detection and instance segmentation. IEEE transactions on pattern analysis and machine intelligence pp. 1483–1498 (2019)
5. Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
6. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, pp. 213–229. Springer (2020)
7. Cruz, S.D.D., Wasenmuller, O., Beise, H.P., Stifter, T., Stricker, D.: Sviro: Synthetic vehicle interior rear seat occupancy dataset and benchmark. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 973–982 (2020)
8. Fang, H.S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., Li, Y.L., Lu, C.: Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
9. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron (2018)
10. Jocher, G., Chaurasia, A., Qiu, J.: YOLO by Ultralytics (2023). URL https://github.com/ultralytics/ultralytics
11. Katrolia, J.S., Mirbach, B., El-Sherif, A., Feld, H., Rambach, J., Stricker, D.: Ticam: A time-of-flight in-car cabin monitoring dataset. arXiv preprint arXiv:2103.11719 (2021)
12. Kiran, B.R., Thomas, D.M., Parakkal, R.: An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. Journal of Imaging **4**(2), 36 (2018)
13. Koay, H.V., Chuah, J.H., Chow, C.O., Chang, Y.L., Rudrusamy, B.: Optimally-weighted image-pose approach (owipa) for distracted driver detection and classification. Sensors **21**(14) (2021). DOI 10.3390/s21144837
14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740–755. Springer (2014)
15. McNally, W., Vats, K., Wong, A., McPhee, J.: Rethinking keypoint representations: Modeling keypoints and poses as objects for multi-person human pose estimation. arXiv preprint arXiv:2111.08557 (2021)
16. Naphade, M., Wang, S., Anastasiu, D.C., Tang, Z., Chang, M.C., Yao, Y., Zheng, L., Rahman, M.S., Arya, M.S., Sharma, A., Feng, Q., Ablavsky, V., Sclaroff, S., Chakraborty, P., Prajapati, S., Li, A., Li, S., Kunadharaju, K., Jiang, S., Chellappa, R.: The 7th ai city challenge (2023)
17. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10,781–10,790 (2020)
18. Tran, M.T., Vu, M.Q., Hoang, N.D., Bui, K.H.N.: An effective temporal localization method with multi-view 3d action recognition for untrimmed naturalistic driving videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3168–3173 (2022)
19. Vats, A., Anastasiu, D.C.: Key point-based driver activity recognition. In: 2022 IEEE Conference on Computer Vision and Pattern Recognition Workshops, *CVPRW'22*, vol. 1 (2022)