Modeling Required Driver Attention Level Based On Environmental Risk Factors Using Deep Convolutional Neural Networks

Jayani Withanawasam1) Ehsan Javanmardi2) Yanlei Gu3) Shunsuke Kamiyo4)
1) Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1, Hongo, Bunkyo City, Tokyo, 113-8656, Japan (E-mail: withanawasamjayani@kmj.iis.u-tokyo.jp)
2) Institute of Industrial Science, The University of Tokyo, 4-6-1, Komaba, Meguro-ku, Tokyo, 153-8505, Japan.
3) College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga, Japan.

Received on August 19, 2021

ABSTRACT: Understanding the level of environmental risk using vehicle-mounted camera traffic scenes is useful in advanced driver assistance systems (ADAS) to improve vehicle safety. We propose a fast, memory-efficient computer vision based environmental risk perception method using a weakly supervised convolutional neural network-based classifier. We use traffic scenes from Berkley deep drive dataset to evaluate the proposed method. Experimental results demonstrate that the proposed method correctly classifies required driver attention levels by considering multiple environmental risk factors. Further, we use class activation mapping to demonstrate that the proposed network is capable of identifying the underlying environmental risk factors.

KEY WORDS: Safety, Road environment recognition, Image processing/ information processing, Smart vehicles/ Environmental risk, Computer vision (C1)

1. Introduction

Smart vehicles such as autonomous vehicles and vehicles equipped with advanced driver assistance systems (ADAS) aim to facilitate improved vehicular safety in next-generation intelligent transportation systems. Environment perception using vehicle-mounted cameras is a fundamental step in such systems. The early perception of potentially risky situations helps human drivers to stay on alert and prevent dangerous situations such as fatal traffic accidents. A large number of traffic accidents are caused because of the inattention of the drivers due to different reasons such as mobile phone usage, eating and drinking while driving, and changing channel or volume of radio(1,2). Good drivers consider numerous environmental risk factors such as pedestrians, gaps to the neighboring vehicles, and weather conditions such as rain to determine when to be more vigilant (Fig. 1). For example, drivers should pay more attention during a rainy night having multiple vehicles and pedestrians around than during a clear day having ample free road space. Recent computer vision based methods such as pedestrian detection and drivable area segmentation assist smart vehicles for safer vehicle operation. To complement the existing environment perception methods in smart vehicles, we introduce an improved road environment recognition method that collectively considers different environmental risk factors and determines the required human driver attention level. The acquired knowledge provides better situational awareness for smart vehicles in diverse applications. For example, it is useful for human driver distraction monitoring systems to issue more informative, context-based warnings.

We define three required driver attention levels (high, medium and low) based on the presence of different underlying environmental risk factors such as pedestrians on the crosswalk, moving close to neighboring vehicles. The environmental risk factors are determined by factors that contribute to traffic accidents based on statistical surveys(3).

A fully supervised learning based approach to solve this problem requires labels for both required driver attention level and underlying environmental risk factors (pixel-level, object level, and image level). This requires an enormous labor-intensive labeling effort. Therefore, we use a weakly supervised Convolutional Neural Network (CNN) to directly learn the high-level concept of required driver attention level. It is more efficient to learn a model that can learn the constituent, low-level environmental risk factors automatically, given only the high-level, single image-level label per traffic scene. We assess the accuracy of the model by using a subset of traffic scenes in Berkley deep drive dataset. Autonomous vehicles and ADAS require real-time operation in a mobile, resource-constrained setting. Therefore, in addition to model accuracy, we consider the memory consumption and processing latency in our proposed method as well.

Further, in safety-critical applications such as ADAS, we should guarantee that the model actually makes predictions based on the relevant underlying risk factors given only the high-level labels for required driver attention level. Therefore, we train the model end-to-end only using image-level labels (without object bounding boxes or pixel-wise labels for semantic segmentation) and localize the regions that contribute towards its prediction using class activation maps to verify if they include the right environmental risk factors.

Predicting a traffic scene as low attention in an actual high attention scenario is more dangerous than predicting as high attention when it is actually low attention. Accordingly, in addition to standard accuracy, we evaluate our model using a different evaluation measure as well. Further, we clearly demonstrate an
ADAS application of the proposed method using a human driver distraction monitoring systems for automotive engineering.

2. Related work

2.1. Vision based traffic scene understanding

Vision-based traffic scene understanding using vehicle-mounted cameras has become increasingly popular due to recent advancements in deep learning. Widely used vision tasks such as object detection and recognition (e.g., pedestrian detection) and semantic segmentation (e.g., segment the image to regions such as road, sky, building) methods mainly focus on “what is where?” in a traffic situation. While such methods are beneficial in smart vehicles to facilitate advanced collision avoidance features (e.g., emergency brake), a better situation awareness requires further understanding of potentially dangerous traffic situations where drivers should pay more attention. Vision-based methods are proposed for tasks such as localizing road scene regions where pedestrians likely to appear and potentially unsafe lane change behaviors. Such methods cover a few specific traffic scenarios. The level of attention drivers should pay depends on multiple environmental factors such as pedestrians, nearby vehicles, weather conditions (snow, rain), and road layout. In a real-world situation such factors do not appear in isolation, so such global, contextual factors should be collectively considered to determine a risky situation. In another work, a more global risk rating for the traffic scene is given by using Youtube videos on road accidents. However, only a subset of real-world traffic situations that drivers should remain more vigilant is followed by an actual road accident, and rarely the respective video footage is shared online, so the coverage of potential environmental risk factors is still relatively low. Further, subjective human opinion is used to classify each traffic scene into three levels namely normal, caution, and warning. In contrast, environmental factors that contribute to road accidents (e.g., statistical survey based on experienced police officers) are considered in another work to more objectively determine a global risk level for the traffic scene using standard CNNs. However, even though the smart vehicle systems such as autonomous vehicles and ADAS operate in a mobile, resource-constrained, real-time setting, lightweight neural architectures that have a better focus on balancing the tradeoff between both accuracy and operational cost (e.g., processing latency, memory) are not yet considered in this work when determining the neural network architecture.

2.2. Lightweight convolutional neural networks

Progressively larger deep networks lead to increased latency and high memory consumption. Therefore, it is challenging to deploy such models in mobile, embedded systems with real-time operational requirements. For example, smart vehicles are usually equipped with Field Programmable Gate Array (FPGA) based or Application Specific Integrated Circuit (ASIC) based embedded vision applications for ADAS. Such components have limited on-chip memory. Therefore, having smaller models is a necessity for practical usage of vision-based environment recognition applications. Moreover, smaller models require lesser communication bandwidth during Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications as well. This is important for over-the-air model updates between server and vehicle.

Accordingly, relatively lightweight CNNs that use fewer parameters and reduce frequently used base operations such as multiply and add during matrix multiplication are important for reducing the memory consumption and processing latency during real-time operation. Such networks are designed in both hand-crafted manner or automatically by using Neural Architecture Search. In addition to standard CNNs, we consider such lightweight CNN architectures as well.

Computationally modeling the driver gaze as an attention mechanism is a popular research problem. Driver gaze can be used to model the attended location by moving eyes pointing that direction (i.e., overt attention). Such methods are useful to find the attended region (spatially) in a given traffic scene. However, the current understanding of human selective visual attention also involves the act of mentally shifting one's focus without moving one's eyes (covert attention). Accordingly, driver gaze is capable of modeling only a subset of human driver attention. Further, such methods are less suitable for objectively determining the level of required driver attention in a given traffic scene temporally (in comparison to other traffic scenes in another time) due to subjective human risk perception.

3. Proposed method

Our goal is to equip smart vehicles with an awareness of potential danger when perceiving a traffic scene so that the acquired knowledge is useful to know when drivers should pay more attention.

3.1. Required driver attention level classifier

We represent this awareness of potential danger quantitatively by specifying three required driver attention levels for traffic scenes. Risky situation is a potentially dangerous situation that might lead to hazardous event in future. One possible way to determine these driver attention levels is to ask humans to label data based on their sense of danger. However, human risk perception varies based on different factors such as gender, age, driving experience and cultural aspects. Since human risk perception is subjective, we need a systematic way to more
objectively define environmental risk in our method. Extensive statistical surveys are conducted based on more reliable sources such as opinions of experienced police officers\(^{33}\) and driving instructors\(^{22}\) (about different causes for traffic accidents and when to be more vigilant). Consequently, we use environmental factors that contribute to traffic accidents (e.g., poor weather conditions such as rain, snow, driving close to parked vehicles)\(^{33}\) to define a potentially dangerous situation in our method. We have given the criteria for three required driver attention levels using the identified environmental risk factors in Table 1\(^{(10)}\).

Table 1 Required driver attention level and the underlying environmental risk factors.

| Required driver attention level | Underlying environmental risk factors                                      |
|---------------------------------|------------------------------------------------------------------------------|
| High attention                  | • Pedestrians and cyclists on the road                                       |
|                                 | • Vehicles immediately ahead or too close                                   |
|                                 | • Complex road layout such as bend, inter-sections, merging lanes           |
|                                 | • Road-side parked cars                                                    |
|                                 | • Crosswalk                                                                |
|                                 | • Poor weather conditions such as rain, snow                               |
|                                 | • High traffic complexity (more than three vehicles around in the visible range) |
| Medium attention                | • Less than or equal to three vehicles around (medium traffic complexity)    |
|                                 | • No vehicles around (night)                                                |
| Low attention                   | • No vehicles around in daytime (Low traffic complexity)                    |

We process input traffic scenes from a vehicle-mounted monocular camera. Example traffic scenes for each class category are given in Fig. 2.

![Fig. 2 Example traffic scenes for each driver attention level](image)

We use Convolutional Neural Networks (CNN) for required driver attention level classification inspired by its recent advances in various applications. CNN is a specific type of artificial neural network (for images) that is loosely inspired by the human visual cortex.

Forward propagation in artificial neural networks calculates an intermediate value \(Z\) by multiplying input feature map \(A^{[l-1]}\) with weights \(W\) and adding a bias \(b\) in each layer \(l\). Then, non-linear activation \(g\) is applied to generate the output activation \(A^{[l]}\) as given in Equation 1.

\[
Z^{[l]} = W^{[l]} \cdot A^{[l-1]} + b^{[l]}, \quad A^{[l]} = g^{[l]}(Z^{[l]}) \tag{1}
\]

The equivalent operation to Equation 1 in convolutional layer is given in Equation 2.

\[
G[m, n] = (A \ast W)[m, n] = \sum_{j} \sum_{k} W[j, k]A[m - j, n - k] \tag{2}
\]

Convolution operation is applied in CNNs for each input image or feature map \(A\). \(m, n\) refers to input pixels coordinates and \(j, k\) refers to convolutional kernel coordinates. Convolution operation is followed by a pooling operation to reduce the noisy activations and provide invariance to small translations in the input. This is achieved by summarizing the feature maps (e.g., max pooling, average pooling). We recommend\(^{(23)}\) for detailed information about CNNs. Various intermediate layers of CNN are proven useful to capture object-like features with weak supervision both in object detection\(^{(24)}\) and scene classification\(^{(25)}\). This property is desirable in our proposed weakly-supervised method as each required driver attention level could contain different objects such as pedestrians and vehicles.

Convolutional kernel is used as a filter (using weights) to extract only salient features from the input either image or a feature map from the previous layer. Multiple convolutional kernels are using in CNNs to automatically capture different types of features such as edges, color, and texture. Each kernel is convolved across the input in a sliding window manner to compute the output feature map. Element wise multiplication and addition is performed in each sliding window step. All the output feature maps are stacked together as multiple channels and sent to the next layer. Standard CNNs involved large number of convolutional filters of multiple sizes such as \((7 \times 7, 5 \times 5, 3 \times 3)\)\(^{(26, 27)}\) along convolutional layers to enable a greater representational power by capturing diverse image features at a higher scale.

This advantage comes at the cost of dramatically increased computational resources. For example, in a two-convolution layer setting, a uniform increase of convolutional kernels will result in about quadratic increase in computation\(^{(28)}\). The increased number of weights in the network requires more memory (for both storage and processing). Further, the number of feature maps also increases along the depth of the CNN hence requiring more processing memory. Such resource requirements hinder efficient and cost-effective mobile, real-time operations in smart vehicles.

Therefore, accommodating lightweight deep neural networks with fewer parameters and reduced operations (e.g., multiply-adds during inference time) is beneficial to decrease memory consumption and processing latency.

3.1.1 Pointwise convolution

One such simple and yet powerful method is to use pointwise convolution\(^{(29)}\) (also known as \(1 \times 1\) convolution) that is widely adopted in CNN architectures\(^{(14,15,28)}\). Pointwise convolution is used for dimension reduction in lightweight CNNs\(^{(14,15)}\) to obtain a channel-wise low dimensional embedding. In pointwise convolution, \(1 \times 1 \times D\) convolutional kernel is applied to an input of...
dimension H*W*D to obtain a lower-dimensional output of H*W*1 (depthwise). Pointwise convolution can be considered as a parametric cross channel pooling on a normal convolutional layer having each pooling layer performing a linear combination on the input feature maps\(^{(20)}\).

In SqueezeNet\(^{(11)}\), a novel CNN architecture with a fire module is introduced to reduce the number of parameters in the network (model compression) while preserving a good accuracy. Fire module consists of two components namely squeeze layer and expand layer. In squeeze layer, pointwise convolution is used to replace 3*3 filters that were prominent in popular standard CNN architectures\(^{(20)}\). Squeeze layer only comprises 1*1 convolutional filters to attain fewer parameters. Expand layer consists of both 1*1 and 3*3 convolutional filters. Pointwise convolution is used to limit the number of input channels to the expand layer as well.

3.1.2 Depthwise separable convolution

The standard convolution performs filtering the input features and combining the outputs in a single step as given in Equation 3.

\[ G_{k,l,m} = \sum_{i,j,n} K_{i,j,m} A_{k-1,i,j-1,n} \tag{3} \]

Depthwise separable convolution\(^{(13)}\) factorizes standard convolution operation into two separate operations namely depthwise convolution and pointwise convolution to enable more lightweight filtering. Depthwise convolution applies only a single convolutional filter per channel as given in Equation 4.

\[ \hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} A_{k+1,i,j+1,m} \tag{4} \]

Pointwise convolution is used to combine the output channels of the depthwise convolution. The goal is to reduce the model size and the computation time. This helps to model spatial co-relation and cross-channel co-relations of the features separately while reducing the number of parameters required. Given H*W*D input, K*K*D filter, and N output channels (N filters), the standard convolution operation has a computational cost of K*K*D*N*H*W (assuming stride of one). The computational cost increases multiplicatively on the number of input channels D. Depthwise separable convolution has a reduced computational cost of 1/N + 1/K\(^2\).

3.1.3 Group convolution and channel shuffle

The computational complexity of pointwise convolution can be further reduced by having channel sparse connections instead of dense 1*1 computations across the entire input. In group convolution\(^{(14)}\), different filters are dedicated for each grouped input feature map to enable model-parallelization. Since each convolutional filter operates only on a subset of the input channels, the operational cost can be drastically reduced. Further, the reduced number of dense channel connections allows room for more convolutional filters thus achieving this increased computational cost while still having a better representational power. Even though group convolution enables more efficient computation, it prevents information from flowing between such groups. Information flow is allowed only within groups thus weakening its representational power. To mitigate this issue in group convolution, channel shuffle\(^{(14)}\) is introduced to blend the information flow between different groups. This facilitates each group to derive information from other groups as well. Having a convolutional layer with G groups and output of G*N channels, output channel dimension is reshaped, transposed and flattened before sending it as input to the next layer to achieve the channel shuffling effect.

Handcrafting CNN architectures to balance the tradeoff between accuracy and operational cost is challenging as there can be many possible architectures in a given search space. Therefore, in addition to the handcrafted CNN architectures, Neural Architecture Search (NAS) that is based on reinforcement learning is also used to come up with more lightweight architectures\(^{(15)}\). Such methods approach the network design as a multi-objective optimization problem that considers both the model inference latency and accuracy and reward models with the best tradeoff.

3.2. Verifying the underlying environmental risk factors using implicit attention in convolutional neural networks

We have used a weakly supervised training approach to classify the required driver attention levels instead of using a labor-intensive, fully supervised approach where labels are given to both required attention levels and their corresponding underlying environmental risk factors. Accordingly, we give labels only for the required driver attention levels. However, when applying such method in a safety critical application such as ADAS, it is important to verify that the classification predictions for the required driver attention levels are made of accurate underlying environmental factors (i.e., provide a rationale for model’s prediction). For example, we verify that the model is able to justify correctly that the driver should pay high attention because there is a pedestrian on the road.

During the training process, CNNs inherently learn an implicit attention mechanism that focuses strongly on some regions of input space than other regions. This can be explicitly analyzed and quantified using the Jacobian sensitivity of the output vector w.r.t. input vector (Equation 5). Recently, gradient-based methods\(^{(30-31)}\) have been effective in analyzing the aforementioned implicit attention mechanism in CNNs.

\[ J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{5} \]

Gradient weighted class activation mapping\(^{(32)}\) can be used to obtain a rough region localization map in the image given a target class category. We re-purpose this characteristic to visually verify that the model prediction on predicted required driver attention level is made upon correct underlying environmental risk factors. First, \(f_k\), the importance of each activation map \(A^k\) to class \(c\) is derived using Equation 6. \(y^c\) is the predicted class score \(c\) and \(Z\) represents the spatial locations in each activation map.
\[ \beta_k = \frac{1}{Z} \sum \sum \frac{\partial y^c}{\partial A^c_{ij}} \]  

(6)

Then, ReLU operation is applied for the weighted forward activation maps to preserve only the activations that have a positive contribution on the predicted required attention level using Equation 7.

\[ L_{Grad-CAM} = ReLU \left( \sum_k \beta_k A^k \right) \]  

(7)

Improvement for the Gradient weighted class activation mapping is proposed\(^{(33)}\) by modifying \( \beta_k \) as given in equation 8.

\[ \beta_k = \sum_i \sum_j \alpha_{ij} \cdot relu \left( \frac{\partial y^c}{\partial A^k_{ij}} \right) \]  

(8)

\( \alpha \) is a weighting co-efficient for the pixel-wise gradient for class \( c \) convolution map \( A^k \) (Equation 9). Relu is used to only consider the gradients that contribute positively towards the class prediction.

\[ \alpha_{ij} = \frac{1}{\sum_{l,m} \frac{\partial y}{\partial A_{lm}}} \quad \text{if} \quad \frac{\partial y}{\partial A^k_{ij}} = 1 \]

\[ = 0 \quad \text{otherwise} \]  

(9)

4. Evaluation

Experiments conducted for the two main modules namely required driver attention level classifier and class activation mapping for underlying environmental risk factor verification are explained in this section.

We perform experiments on a subset of the Berkeley Deep Drive dataset (BDD100k)\(^{(34)}\). BDD100k dataset has a variety of traffic scenes that belong to different weather conditions and different times of the day under diverse lighting conditions. We randomly select 1300 images and remove the bad quality traffic scenes. We label the remaining traffic scenes as per the criteria mentioned in Table 1. Training dataset contains 1139 traffic scenes in total having 684, 322 and 133 images for each class category high, medium, and low respectively. Test dataset has 150 images (50 images for each class category). In total the dataset has 1289 traffic scenes. The environmental risk factor distribution that corresponds to Table 2 is given in Fig. 3.

4.1. Lightweight convolutional neural networks

Smart vehicles such as autonomous vehicles and vehicles equipped with ADAS require real-time operation in a mobile, resource-constrained setting. It is important to reduce the operational cost without compromising the model accuracy. Therefore, in addition to standard CNN (ResNet-18)\(^{(27)}\) used in previous work\(^{(10)}\), we conduct further experiments with more lightweight neural networks to facilitate better operational cost in terms of processing latency (process higher number of frames per second) and reduced memory consumption with a competitive accuracy. We consider ResNet-18 as the baseline architecture in this experiment.

Fig. 3 Environmental risk factor distribution

When it comes to lightweight CNN architectures, we conduct experiments with both hand-crafted CNN architectures (e.g., SqueezeNet\(^{(11)}\), MobileNet\(^{(13)}\), ShuffleNet\(^{(14)}\)) as well as automatically generated CNNs using neural architecture search (e.g., MNASNet\(^{(15)}\)). SqueezeNet employs pointwise convolution (section 3.1.1) to improve the operational cost. Depthwise separable convolution (section 3.1.2) is introduced in MobileNet and further improvements are proposed in MobileNet v2\(^{(17)}\). In MobileNet v2, a novel layer module called inverted residual with linear bottleneck is introduced to facilitate shortcut connections (similar to residual connections in ResNet architectures\(^{(27)}\)) between thin bottleneck layers. This module allows better information flow between layers with more memory efficient architecture. Group convolution along with channel shuffle (section 3.1.3) is used in ShuffleNet to improve the operational cost while still having a reasonable representational power with lesser parameters. In ShuffleNet v2\(^{(18)}\), it is further improved using channel split operation.

We use above CNN models initialized with ImageNet\(^{(15)}\) weights for required driver attention level classification. We use Adam\(^{(30)}\) as the optimizer for training. We set hyper-parameters, learning rate to 0.0003, number of epochs to 40, and batch size to 4. We use balanced data sampling for each class category in the dataset during each epoch to avoid bias due to data imbalance in the training dataset. Experiments are conducted on a computer with an Nvidia GeForce GTX 1080 GPU\(^{(39)}\) for both training and inference.

Accuracy of the above classification models is given in Table 2 along with classwise recall. Confusion matrices are given in Fig. 4 and operational cost in terms of memory and processing latency is given in Table 3.
Another interesting observation is that handcrafted neural networks architectures such as ResNet-18, MobileNet v2 and SqueezeNet have better performance over more recent automated neural network architectures using neural architecture search such as MNASNet.

Nvidia GeForce GTX 1080 GPU\(^{(39)}\) is a general purpose hardware used in various domains. On the other hand, more power-efficient, GPU based, embedded AI devices such as Nvidia Drive\(^{(40)}\) are recently introduced for on-board equipment in intelligent vehicles. They contain multiple optimizations to enable more efficient model inference. For example, ARM based processors such as Nvidia Jetson AGX Xavier\(^{(41)}\) are equipped with more advanced GPU features namely vision accelerators and deep learning inference accelerators. Accordingly, Nvidia Jetson AGX Xavier has a competitive processing latency with Nvidia GeForce GTX 1080 when compared them for convolutional neural network based methods\(^{(42)}\). Therefore, Nvidia AGX Xavier is successfully used for real-time processing in traffic engineering tasks as well\(^{(43)}\).

Deep convolutional neural network models trained on general purpose hardware are interoperable with such special purpose embedded AI devices. Therefore, the proposed deep learning based model can be trained on general purpose hardware and the trained model can be deployed on on-board equipment and used during inference time to get model predictions. Accordingly, the proposed deep learning based method can achieve a competitive performance with less power consumption using such on-board equipment.

### 4.2. Evaluating the safety of the model

The risk of misprediction of the proposed classifier is not symmetric. For example, if the model predicts as ‘low attention’ for an...
actual ‘high attention’ traffic scene, the result can be more dangerous than if the model predicts ‘low attention’ for an actual ‘high attention’ scene. Therefore, we evaluate our method using a different evaluation strategy\(^{(10)}\), in addition to the standard accuracy incorporate the aforementioned aspect as well.

Accordingly, we calculate the safety-precision as given in the Equation 7.

\[
\text{Safety\_Precision} = \frac{TP_i}{TP_i + \sum_{j \in \{0,1\}} FP_j}
\]

where \(i = 2, (0:\text{ high}, 1:\text{ medium}, 2:\text{ low})\) \(\text{ Equation 7}\)

Further, we calculate the safety-recall as given in the Equation 8.

\[
\text{Safety\_Recall} = \frac{TP_i}{TP_i + \sum_{j \in \{1,2\}} FN_{ij}}
\]

where \(i = 0, (0:\text{ high}, 1:\text{ medium}, 2:\text{ low})\) \(\text{ Equation 8}\)

We calculate the safety score of the model as given in Equation 9.

\[
\text{Safety\_Score} = 2 \times \frac{\text{Safety\_Precision} \times \text{Safety\_Recall}}{\text{Safety\_Precision} + \text{Safety\_Recall}}
\]

\(\text{ Equation 9}\)

The experimental results for safety is given in Table 4.

**Table 4 Quantitative results for safety score**

| Trained model  | Safety precision | Safety recall | Safety score |
|----------------|------------------|---------------|--------------|
| ResNet-18      | 0.82             | 0.86          | 0.84         |
| MNASNet        | 0.84             | 0.92          | 0.87         |
| SqueezeNet     | 0.78             | 0.80          | 0.79         |
| MobileNet v2   | 0.80             | 0.86          | 0.90         |
| ShuffleNet v2  | 0.71             | 0.82          | 0.76         |

**4.2. Verifying the underlying environmental risk factors using class activation mapping**

The class activation maps for traffic scenes from each required driver attention level are given in Fig. 5. The model is able to accurately visually justify the corresponding underlying environmental risk factor that is the reason for the traffic scene to classify as a given required driver attention level. The model is capable of correctly localizing the underlying environmental risk factors for class ‘high attention’ such as ‘pedestrian on the road’, ‘vehicle immediately ahead’, and ‘road-side parked vehicles’. However, the model does not perform well with traffic scenes that contain snow and also traffic scenes with bends due to the fewer training examples available. Further, in some cases, we experience visually noisy localization regions.

**4.2.2. Improvements in class activation mapping**

In our case, there are many traffic scenes that contain multiple objects with the same class such as multiple pedestrians and multiple vehicles (different traffic complexity levels, parked vehicles). It is discussed in recent work\(^{(33)}\) that in some scenarios, Grad-CAM\(^{(32)}\) fails to properly localize objects in an image if the image contains multiple occurrences of the same class and class activation maps can be visually noisy. One possible reason for this limitation could be due to the problem of the unweighted average of partial derivatives considered in equation 7. Feature maps of lesser spatial footprint could fade away because of this issue. In Grad-CAM++,\(^{(33)}\) an improvement to Grad-CAM is proposed by taking a weighted combination of gradients by reformulating equation 7 using a new weighting coefficient \(\alpha\). The presence of objects in all feature maps is taken into consideration with equal importance using this method. We apply this change to improve the class activation mapping to verify the underlying environmental risk factors in our method.

In Smooth Grad-CAM\(^{(44)}\), an improvement for rapidly fluctuating partial derivatives is proposed by taking random samples in a neighborhood of an input image and averaging the resulting class activation maps as a smoothing technique using Gaussian noise. The proposed technique is capable of deriving visually sharper class activation maps.

\[
\text{class}(x) = \arg\max_{c \in C} S_c(x)
\]

\(\text{ Equation 10}\)

\[
M_c(x) = \frac{\partial S_c(x)}{\partial x}
\]

\(\text{ Equation 11}\)

\[
M'_c(x) = \frac{1}{n} \sum_{i=1}^{n} M_c(x + \mathcal{N}(0,\sigma^2))
\]

\(\text{ Equation 12}\)

where \(n\) is the number of samples, and \(\mathcal{N}(0,\sigma^2)\) represents Gaussian noise with standard deviation \(\sigma\). Improvements using the aforementioned methods to mitigate the issue of visually noisy class activation maps are given in fig. 6.

**Fig. 6 Improvements in class activation maps**

**5. Application in Advanced Driver Assistance Systems**

The majority of the traffic accidents are caused because of driver distraction due to various reasons such as mobile phone usage and consuming food and drinks while driving\(^{(12)}\). Automated human driver distraction monitoring systems is a novel ADAS system in smart vehicles to facilitate improve vehicle safety by issuing a warning for distracted drivers\(^{(43)}\). The damage that can occur due to driver distraction is higher if the driver is distracted in a traffic situation to which he/she should pay more attention to (e.g. if there are pedestrians crossing the road during a rainy night). Current driver distraction monitoring systems can be further enriched to issue different levels of warnings based on such situational context using our proposed method. An example diagram for the proposed ADAS application is given in Fig. 7.
5. Conclusion

In this paper, we proposed a fast, memory-efficient CNN based classifier to predict the driver attention level required for a given traffic scene. We experimentally demonstrated that such networks can achieve a better operational cost without compromising the accuracy. Further, we have verified the reliability of the proposed method using recent class activation mapping techniques. Additionally, we demonstrated an integrated application of the proposed method in real-world smart vehicles with driver distraction monitoring systems.

References

(1) Sundfør, H.B., Sagberg, F., and Høye, A., Inattention and distraction in fatal road crashes – Results from in-depth crash investigations in Norway. Accident Analysis & Prevention 125, p.152-157, 2019.

(2) Richard, T.A.Y., and Knowles, D., Driver inattention: drivers’ perception of risks and compensating behaviours, IATSS research 28(1), p.89-94, 2004.

(3) Rolison, J.J., Regev, S., Moutari, S., and Feeney, A., What are the factors that contribute to road accidents? an assessment of law enforcement views, ordinary drivers opinions, and road accident records, Accident Analysis & Prevention 115, p. 11–24, 2018.

(4) Brunetti, A., Buongiorno, D., Trotta, G.F., and Bevilacqua, V., Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. Neurocomputing 300, p.17-33, 2018.

(5) Wang, S., Fidler,S., Urtasun, R., Holistic 3d scene understanding from a single geo-tagged image, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 3964–3972. https://doi.org/10.1109/CVPR.2015.7299022

(6) Kozuka, K., and Carlos Niebles, J., Risky Region Localization with Point Supervision. Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 246-253), 2017.

(7) Yurtseven, E., Liu, Y., Lambert, J., Miyajima, C., Takeuchi, E., Takeda, K., and Hansen, J.H., Risky action recognition in lane change video clips using deep spatiotemporal networks with segmentation mask transfer, 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 3100-3107), 2019. October.

(8) Yu, S.Y., Malawade, A.V., Muthirayan, D., Khargonekar, P.P., and Faruque, M.A.A., Scene-graph augmented data-driven risk assessment of autonomous vehicle decisions. arXiv preprint arXiv:2009.06435, 2020.

(9) Wang, Y. and Kato, J., Collision risk rating of traffic scene from dashboard cameras, 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), p. 1-6, IEEE.

(10) Withanawasam, J., Javanmardi, E., Wong, K., Javanmardi, M., and Kamijo, S., Road scene risk perception for intelligent vehicles using end-to-end affordance learning and visual reasoning, Pattern Recognition. Springer International Publishing, pp. 849–862, 2020.

(11) Jandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., and Keutzer, K., SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360, 2016.

(12) Consumer Reports. Teslas new autopilot: Better but still needs improvement. https://www.consumerreports.org/tesla/tesla-new-autopilot-better-but-needs-improvement/, 2016, accessed April 16, 2021.

(13) Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

(14) Zhang, X., Zhou, X., Lin, M., and Sun, J., Shufflenet: An extremely efficient convolutional neural network for mobile devices, Proceedings of the IEEE conference on computer vision and pattern recognition (p. 6848-6856), 2018.

(15) Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q.V., Mnasnet: Platform-aware neural architecture search for mobile, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2820-2828), 2019.

(16) A. Palazzi, D. Abati, F. Solera, R. Cucchiara, et al., Predicting the driver’s focus of attention: the dr (eye) ve project, IEEE transactions on pattern analysis and machine intelligence 41 (7), p. 1720–1733, 2018.

(17) Tawari A., and Kang, B., A computational framework for driver’s visual attention using a fully convolutional architecture, 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2017, p. 887–894.

(18) Sjøberg, L., Moen, B.E., and Runaldo, T., Explaining risk perception: an evaluation of the psychometric paradigm in risk perception research, Rotundes publikasjoner Rotunde 84, p. 55–76, 2004.

(19) Ping, P., Sheng, Y., Qin, W., Miyajima, C., and Takeda, K., Modeling driver risk perception on city roads using deep learning, IEEE Access 6, p. 68 850–68 866, 2018.

(20) Machado-León, J. L., de Oña, de Oña, J. R., Eboli, L., and Mazzulla, G., Socio-economic and driving experience factors affecting drivers’ perceptions of traffic crash risk, Transportation research part F: traffic psychology and behaviour 37, p. 41–51, 2016.

(21) Lund, I. O., and Rundmo, T., Cross-cultural comparisons of traffic safety, risk perception, attitudes and behaviour, Safety Science 47 (4), p. 547–553, 2009.

(22) Kokubun, M., Konishi, H., Higuchi, K., Kurahashi, T., Umemura, Y., and Nishi, H., Assessment of drivers’ risk perception using a simulator. R&D Review of Toyota CRDL 39(2), p. 9–15, 2004.

(23) Goodfellow, I., Bengio, Y., and Courville, A., Deep learning. MIT press, 2016.

(24) Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A., Object detectors emerge in deep scene cnns, arXiv preprint arXiv:1412.6856, 2014.

(25) Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A., Learning deep features for scene recognition using places database, in Advances in neural information processing systems, p. 487–495, 2014.
(26) Krizhevsky, A., Sutskever, I. and Hinton, G.E., Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25, pp.1097-1105, 2012.

(27) He, K., Zhang, X., Ren, S., and Sun, J., Deep residual learning for image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, p. 770–778, 2016.

(28) Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition (p. 1-9), 2015.

(29) Lin, M., Chen, Q., and Yan, S., Network in network. arXiv preprint arXiv:1312.4400, 2013.

(30) Simonyan, K., Vedaldi, A., and Zisserman A., Deep inside convolutional networks: Visualising image classification models and saliency maps, arXiv preprint arXiv:1312.6034, 2013.

(31) Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A., Learning deep features for discriminative localization, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, p. 2921–2929.

(32) Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., Grad-CAM: Visual explanations from deep networks via gradient-based localization http://arxiv.org/abs/1610.02391.

(33) Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks, 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), p. 839-847.

(34) Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T., BDD100K: A diverse driving dataset for heterogeneous multitask learning, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, p. 2636-2645, 2020.

(35) Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L., Imagenet: A large-scale hierarchical image database, 2009 IEEE conference on computer vision and pattern recognition, p. 248–255, 2009.

(36) Kingma, D. P., and Ba, J., Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

(37) Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, p. 4510-4520, 2018.

(38) Ma, N., Zhang, X., Zheng, H.T. and Sun, J., Shufflenet v2: Practical guidelines for efficient cnn architecture design, Proceedings of the European conference on computer vision (ECCV), p. 116-131, 2018.

(39) Nvidia GeForce GTX 1080. https://www.nvidia.com/en-in/geforce/products/10series/geforce-gtx-1080/, 2021, accessed August 19, 2021.

(40) Nvidia Drive - Autonomous Vehicle Development Platforms. https://developer.nvidia.com/drive/, 2021, accessed August 16, 2021.

(41) Nvidia Jetson AGX Xavier. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/, 2021, accessed August 19, 2021.

(42) Hossain, S. and Lee, D.J., Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices. Sensors, 19 (15), p. 3371, 2019.

(43) Panero Martinez, R., Schiopu, I., Cornelis, B. and Munteanu, A., Real-time instance segmentation of traffic videos for embedded devices. Sensors, 21(1), p. 275, 2021.