Attentional demand estimation with attentive driving models

Petar Palasek[1]
petar.palasek@mindvisionlabs.com
Nilli Lavie[1,2]
n.lavie@ucl.ac.uk
Luke Palmer[1]
luke.palmer@mindvisionlabs.com

[1] MindVisionLabs
London, UK
[2] Institute of Cognitive Neuroscience
University College London, London, UK

Abstract

The task of driving can sometimes require the processing of large amounts of visual information; such situations can overload the perceptual systems of human drivers leading to 'inattentional blindness', where potentially critical visual information is overlooked. This phenomenon of 'looking but failing to see' is the third largest contributor to traffic accidents in the UK. In this work we develop a method to identify these particularly demanding driving scenes using an end-to-end driving architecture, imbued with a spatial attention mechanism and trained to mimic ground-truth driving controls from video input. At test time, the network's attention distribution is segmented to identify relevant items in the driving scene which are used to estimate the attentional demand on the driver according to an established model in cognitive neuroscience. Without collecting any ground-truth attentional demand data - instead using readily available odometry data in a novel way - our approach is shown to outperform several baselines on a new dataset of 1200 driving scenes labelled for attentional demand in driving.

1 Introduction

Perception in humans has a limited capacity [19], and when a certain task requires high levels of perceptual processing, perceptual systems can become overloaded, resulting in salient objects being completely undetected. Clearly this is problematic when it is safety-critical to notice and perceive certain salient events. For example, when driving, failing to notice a pedestrian crossing the road or an important road-sign could have potentially serious consequences. Therefore, in this work we aim to identify situations in which attentional demand during driving is high, and the chance of missing critical visual information is increased.

In driving research there is much work on the related and multi-faceted attribute of 'mental workload' [6, 34]. This measure combines contributions of the visual demands of the primary task (i.e. driving), other secondary tasks, and the capacity of the driver into a single value of workload that a driver experiences. Some studies have looked specifically at the components of workload which are determined by perception of the driving scene, for example [13] found that traffic density was associated with higher reports of workload by drivers, and similarly [37] found that higher reports of mental workload were associated with urban environments compared to rural and highway driving. However, [9] note that the relationship between driving environment and workload has produced inconsistent results (e.g. [7]); they posit that this is due to the large variation in complexity of driving situation encompassed by such large classifications as ‘urban’ and ‘rural’. In this work we instead analyse the specific content of the scene rather than coarse scene-level categories to estimate attentional demand.

Recent computer vision work [25] approached the topic of estimating the workload involved in perception of the driving scene by labelling a collection of urban driving clips with a ‘demand’ value and treating the problem as a regression from video input to this attentional demand value using C3D features [30]. While [25] found a good level of prediction using a support vector regression model, the method required a large scale labelling operation and the model did not implicitly produce explanations for its predictions. The method developed by [33] again treats driver workload estimation as a supervised learning task, where instead of using the scene features they classify driver gaze patterns using HyperLSTM (an application tailored LSTM variant). [31] approached a related question of estimating the difficulty of visual search for a given image (i.e. how difficult it is to find a specific object within an image). They also treated it as an image regression problem where an SVR model was trained to predict human search times (collected in a large-scale labelling experiment) with VGG19 features as well as human interpretable ones (e.g. ‘objectness’).

We instead develop a novel approach using principles from cognitive science to estimate the attentional demand of a driving situation on a driver. The attentional demand of a visual perception task can be seen as analogous to the concept of perceptual load [17, 18, 19], which is operationally defined as the number of items in a visual task necessary to attend to in order to complete the task, such that increasing the number or relevancy of items or units in the task increases the perceptual load. An open question in the field is how does one define a relevant item in the task? Laboratory tasks investigating the effects of perceptual load on perception have typically manipulated load in simple displays using fairly austere stimuli where the units of the task are easily defined [20]. In real-world tasks however, it is more difficult to define task-relevant items. For example when driving, a couple holding hands and walking together may be attended to as a single item even though we may think of them as two ‘objects’; while a part of the scene (e.g. the space between two parked cars) may be attended to but would not correspond to an ‘object’ as we know it.

In this paper we therefore allow the concept of the task-relevant ‘object’ to be learned, unrestricted by standard object definitions (as used in semantic/instance segmentation, for example) and develop a novel method to estimate the attentional demand of a driving scene based on these cognitive principles. First we design a novel end-to-end driving architecture that includes a spatial attention mechanism allowing the prioritisation of visual information. After training this network to predict driving commands, we are able to then identify task-relevant items (TRIs) using the network’s internal attentional state and calculate the attentional demand of the driving scene according to an operational cognitive model [17]; this algorithm allows the real-time estimation of scene difficulty during driving using only front-facing video input, and trained only with readily-available vehicle odometry data. Finally we collect a new dataset to validate our method and show that it far better aligns with human judgments of driving demand than other baseline methods.

2 Related work

The first attempt at training a model to drive a car dates back 30 years ago when a simple 3-layer neural network [26] was trained to directly map from a camera to the steering angle. More recently, end-to-end driving models have been developed using powerful CNNs to predict driving controls (e.g. [4, 5, 14]), trained using large amounts of ground-truth human driving data. These types of driving models allow us to introduce novel processes into the driving pipeline; in our case we adopt and develop this approach by introducing an interpretable attention mechanism.

Attention mechanisms in neural networks imbue the models with the ability to amplify or attenuate certain features or regions of the input, allowing the network to better focus its resources while also implicitly providing explanations for the decision or behaviour of the network [12]. Attention mechanisms have been employed successfully in natural language processing tasks (e.g. [2, 3, 32]), and more recently have gained traction in the visual domain. For example, squeeze-excitation-networks [15] employ a channel-based attention mechanism allowing the network to amplify and attenuate feature channels conditional on the input content, resulting in state-of-the art result on the ImageNet dataset. Spatial attention mechanisms differ in that they allow the network to prioritise certain locations or regions of the input, and have found success in visual question answering [1], image captioning [22, 36], and action recognition [11]. The attentional pooling network [11] parameterises a weight for each location in the feature map as a 1x1 convolution of the previous feature map - this simple addition leads to state-of-the art RGB-only action recognition, while also producing sensible and interpretable explanations for network decisions. We therefore implement and develop attentional pooling as the attention mechanism for our end-to-end driving model.

Attention has recently been investigated in the context of driving. For example, [24] produced a predictive model of driver gaze by collecting a large dataset of ground-truth driver eye-fixations and training a pixel-to-pixel model to estimate gaze distributions from input video. Although the model does indeed predict eye position, it is currently unclear what relationship this has to attention itself which is a related but distinct process (e.g. [27]). [16] so designed and trained an end-to-end driving network with an attention mechanism, however they used the output of their attention mechanism to inform an occlusion-based visualisation method (following e.g. [38]) to determine explanatory scene regions for model behaviour, rather than to estimate cognitive states of the driver. The work presented here is therefore the first to estimate human attentional state by simulating an attention mechanism in a neural network trained only to complete the task at hand, without any supervision regarding the attentional state itself.

3 Attentive driving model

3.1 Temporal attentional pooling (TAP)

Given a sequence of images $\mathcal{I} = \{I_1, ..., I_T\}$ and a ground truth label y assigned to the whole sequence, we would like to develop a model which predicts the label $\hat{y}$ for the given sequence, while also providing a way of interpreting which parts of the input were important in making that prediction. To this end we propose extending the attentional pooling method of [11] to the temporal domain using a convolutional LSTM, and so define an architecture we call temporal attentional pooling (TAP). The TAP architecture consists of three components: a perceptual module for extracting frame-based visual representations (VGG19 [28]), followed by temporal integration - we use a convolutional LSTM to maintain the spatial structure of the data (ConvLSTM [35]), and finally an attention mechanism to prioritise certain regions of the input (attentional pooling [11]).

The ConvLSTM [35] is an extension of the LSTM model which can be used to extract a representation of a sequence including a spatial structure. The main parts of the ConvLSTM are its two internal states: the hidden state H and the cell state C; and the gates which control how the internal states are modified: the input gate i, the forget gate f, and the output gate o. We denote the value of a gate or a state variable at a certain time step by including a subscript t, i.e. the cell state C at time t is denoted as C_t and the hidden state at the previous time step is H_{t-1}. Given a 3D tensor X_t as the input at time step t, all of the internal gates and states are updated according to the following equations:

$$i_t = \sigma\left(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i\right) \tag{1}$$

$$f_t = \sigma\left(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f\right) \tag{2}$$

$$o_t = \sigma\left(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o\right) \tag{3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh\left(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c\right) \tag{4}$$

$$H_t = o_t \odot \tanh\left(C_t\right) \tag{5}$$

where σ denotes the sigmoid function, $*$ is the convolution operator, $\odot$ denotes the Hadamard product, W_* are the filter weights used in the convolutions, and b_* are the biases. Note that the equations presented above are slightly different than the ones described in [35]; we do not include the cell state C when calculating the gate activations, i.e. we do not use peephole connections [10] as we did not notice an effect on the performance when they were used.

The hidden state H and the cell state C, as well as the input X are 3D tensors with equal trailing two dimensions. The number of channels in H and C is equal and is set arbitrarily as a hyperparameter of the model, while the number of channels of X depends on the choice of the feature extraction model used for processing the input images, i.e. $X_t = FE(I_t)$, where we used FE to denote a convolutional network, such as VGG19 [28] with its fully connected layers removed. The output of the ConvLSTM model is the value of its hidden state H_T, i.e. the hidden state at the last time step, after the whole sequence $\mathcal{I}$ has been processed. We can interpret H_T as a representation of the whole sequence, and we feed it into the final part of the proposed TAP, which is the classification module with an attention mechanism.

The output of the ConvLSTM is a spatial feature map to which we apply an attentional pooling [11] decision layer to predict the target label. Scores for each class are calculated as inner products of two attentional heatmaps; 1) a class-dependent heatmap representing which parts of the input are indicative of a particular class, and 2) the class-independent heatmap representing which parts are important for classifying the given sample in general. More formally, the score of a sample M belonging to class k can be written as:

$$score\left(M,k\right) = \left(M\mathbf{a}_k\right)^T \left(ReLu(M\mathbf{b})\right), \tag{6}$$

where we use M to denote a 3D tensor (in our case the tensor H_T) viewed as a matrix of shape $(n \times ch)$, with n being the number of spatial locations in the tensor and ch the number of its channels. The vectors $\mathbf{a}_k$ and $\mathbf{b}$ denote the class-specific and class-agnostic weights, respectively. Reshaping the product $M\mathbf{b}$ into a matrix of the same size as the spatial size of the tensor H_T results in a heatmap which can be interpreted as a distribution of importance map across space. Note that we pass the class-agnostic heatmap through a ReLu activation

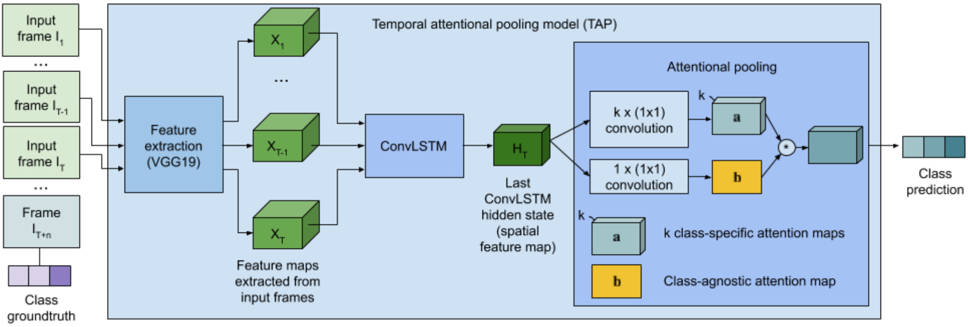

Input frame I_1
...
Input frame I_{T-1}
Input frame I_T
...
Frame I_{T+n}
Class groundtruth
Temporal attentional pooling model (TAP)
Feature extraction (VGG19)
X_1
...
X_{T-1}
X_T
Feature maps extracted from input frames
ConvLSTM
H_T
Last ConvLSTM hidden state (spatial feature map)
Attentional pooling
k x (1x1) convolution
1 x (1x1) convolution
k
a
b
k
k class-specific attention maps
Class-agnostic attention map
Class prediction

Figure 1: Illustration of the proposed TAP model. The input sequence is first passed through a feature extraction network and the resulting features are then processed by a ConvLSTM which outputs a time-aggregated representation of the whole sequence. This representation then undergoes attentional pooling, which predicts the label for the shown sequence. The class-agnostic attention map is what we later use for estimating the attentional demand of the driving scene, as it represents which parts of the scene are important to perceive for making the overall driving decision (in contrast to the class-specific maps which encode local evidence for specific driving controls).

function otherwise a negative value in the class-agnostic heatmap would not necessarily mean that the feature at that location is unimportant (since the class-dependent attention values can take on negative values). An illustration of the final TAP model can be seen in Figure 1.

3.2 Estimating attentional demand

In the cognitive science literature (e.g. [19]), the perceptual load of a task is operationally defined in terms of the number and relevance of the visual 'items' it is necessary to perceive in order to complete the task. Here we name these task-relevant items (TRIs), and such items may or may not be 'objects' in the normal sense. To identify such TRIs at test time using a trained TAP driving model, we interpret the class-agnostic attention map as identifying regions important for completing the task (i.e. for producing the correct driving control). To identify TRIs from the produced pixel-level attention map, the map is threshold-segmented to isolate contiguous regions of highly relevant pixels. The threshold is set at 1, which is indicative of the attentive driving model amplifying the features describing this pixel region. Contiguous positive regions of this binary map are identified as TRIs where each TRI is represented as a set of pixel coordinates. The attentional demand contribution of each TRI is calculated as the maximum attention value within the TRI. The overall attentional demand of a scene is then the sum of the contributions of each TRI. The algorithm for calculating the attentional demand is shown explicitly in Algorithm 1.

3.3 Baseline estimators

We compare our estimation approach with several baseline methods. Many works in the human-machine interaction literature have reported correlations between driver behaviour

Algorithm 1 Algorithm for calculating demand from attention map.

```
function CALCULATE_DEMAND($att$)
    $bin\_att \leftarrow zeros\_like(att)$                          ▷ initialise a zero array of same shape as att
    $bin\_att[att \geq 1] \leftarrow 1$                             ▷ create binary attention mask
    $TRIs \leftarrow connected\_components(bin\_att)$               ▷ find the set of TRIs
    $total\_demand \leftarrow 0$                                    ▷ initialise output
    for each TRI ∈ TRIs do                                          ▷ each TRI is a set of pixels
        $att\_value \leftarrow 0$
        for each p ∈ TRI do
            $att\_value \leftarrow \max(att\_value, att[p])$        ▷ find max value in att map
        $total\_demand \leftarrow total\_demand + att\_value$        ▷ increment total demand value
    return $total\_demand$
```

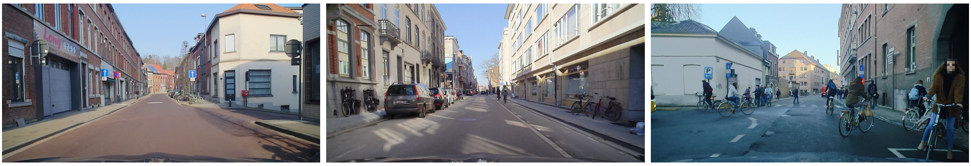
measures and mental workload measures (e.g [21, 23, 29]). Therefore we calculate a suite of sensor-based load estimators in the following ways: **Yaw rate variance** (YRV): yaw rates (serving as a proxy measure of steering commands) recorded during the preceding 12 frames (i.e 1.2 seconds) are aggregated and the variance computed across time; **Yaw rate mean** (YRM): same as previous but mean yaw rate computed across time; **Acceleration variance** (AccV): variance computed across forward acceleration in the preceding 1.2 seconds; **Acceleration mean** (AccM): mean of acceleration across the preceding 1.2 seconds; **Sensor regression**: train a kernel regression model on the 600 instances from the training set, features are constructed as a concatenation of the preceding 12 frames for the yaw rate and forward acceleration values (we implemented both linear (LIN) and RBF kernel variants (RBF)); **File size** (FS): a common and simple baseline for image complexity (e.g. [31]), we use the file size of the query frame compressed as a JPEG.

4 Datasets

4.1 End-to-end driving dataset

We trained our end-to-end attentive driving model using a subset of approximately 100 hours of driving video footage collected from a car driven in the city of Leuven, Belgium. The car was equipped with a global positioning system (GPS) and an inertial measurement unit (IMU) collecting the following sensor data: latitude, longitude, altitude, speed, roll, pitch, yaw, roll rate, pitch rate, yaw rate, forward acceleration, lateral acceleration, and vertical acceleration. Each sensor data stream was linearly interpolated to provide ground-truth values associated with each video frame. In this work we use the video data recorded from a central front-facing RGB camera as input to our model, collected at 10 frames per second at a resolution of 640 x 1280 pixels with a field of view spanning 60° of horizontal visual angle.

Within the dataset *safety-critical traffic events* were identified by the driver in real-time during data collection and noted by the passenger using a button-box which marked the time of each event; therefore the dataset provides excellent examples of difficult driving scenes to feed our model, which are generally hard to find in the open source dataset alternatives. To construct our training dataset from the full 100 hours 3300 13-second (i.e. 130 frame at 10 FPS) sequences of driving were randomly selected, such half contained a safety-critical



(a) Low (b) Median (c) High

Figure 2: Examples of scenes in the validation dataset ordered according to labeled demand.

event (defined to always be at frame 80 in the 130 frame sequence), and the other half did not, to provide variation in driving scene demands. We then randomly selected 2500 of these sequences as the training set, and kept the remaining 800 as a held-out test-set.

4.2 Attentional demand validation set

We collected attentional demand labels for a subset of our dataset. Three-hundred sequences were randomly chosen from the training and test sets, and two frames were extracted from each sequence, the 20th frame and the 80th frame (i.e. the critical event frame). This resulted in a total of 1200 driving scenes, 600 from the train and 600 from the test set.

In the labelling phase, the 1200 frames were randomly ordered and presented to four labellers sequentially, such that each labeller labelled every frame. The labellers were experienced drivers and were instructed to estimate the attentional demand they would feel if they were driving in the scene. The concept of attentional demand was also explained in intuitive terms such that high demand would correspond to wanting passengers in the car to stop talking to allow you to concentrate on the road. Demand was indicated on an ordinal scale of 1-8 inclusive, one being the least demanding driving situation, and 8 being the highest possible. To account for labeller preference, each labeller's ratings were normalised independently, and then averaged across raters; example driving scenes with their corresponding demand labels can be seen in Figure 2. Inter-labeller agreement was highly significant, with the average correlation between pairs of labellers being 0.61, with a standard deviation of 0.08.

5 Experiments and results

5.1 Training details

In this subsection we describe how we trained the proposed TAP model. We begin by defining a simple car control task; given a sequence of T frames $\mathcal{I} = \{I_1, ..., I_T\}$ from a driving video (i.e. video taken from a front-facing camera mounted on a car), predict whether the driver should accelerate, decelerate or continue driving at constant speed at time step $T+n$. As we know the acceleration values a_t for each video frame, we can generate the ground truth label y for a selected sequence by simply binning the acceleration value a at time step $T+n$ into one of three predefined value ranges. In our experiments we label the driving decision for a sequence as 'decelerate' if $a_{T+n} < -0.3m/s^2$, 'accelerate' if $a_{T+n} > 0.3m/s^2$, and 'constant' for other values. We set $T = 12$ and $n = 24$, meaning that the input sequence consists of 12 frames and its corresponding label comes from 24 frames in the future. Taking the video sampling rate into account (10 FPS), the length of the input sequence is 1.2 seconds long, and the driving control we are trying to predict is from 2.4 seconds in the future.

We process each of the frames I_t (resized to 224×448 pixels) in the input sequence by passing them through a feature extraction network; in our experiments we used the convolutional layers of VGG19 [28] with the last pooling layer removed as the feature extraction network. The network was pretrained on the ImageNet dataset [8]. The extracted feature maps X_t (having 512 channels and a spatial size of 14×28 pixels) were then fed into the convLSTM sequence feature extraction module described in Section 3.1, and the resulting representation was used as the input to the attentional pooling module which gave the label prediction at its output. The number of hidden channels in the convLSTM used was set to 128. Each of the convolutional filters W_* in the convLSTM were of size 1x1 px. The entire TAP network was trained end-to-end for a total of 200 epochs by minimising cross-entropy loss with stochastic gradient descent with momentum ($m = 0.9$) with a learning rate of 0.01 and training batch size 24 on 8 GeForce GTX 1080 Ti GPUs. The learning rate was divided by 2 halfway throughout the training. The total number of parameters in the network is 20.3M. One epoch consisted of showing 2496 uniformly random samples from the training set. The training took around 145 seconds per epoch, meaning the 200 epochs took around 8 hours. The classification accuracy on the testing set after the 200 epochs was 53.1%.

6 Results

After training the TAP model for 200 epochs, we then extracted attention maps for each of the 1200 frames in the validation demand dataset. The map was calculated by feeding the preceding 12 frames sequentially into the TAP model and taking the attention of the current frame. Then overall attentional demand values for each frame/attention map were calculated using Algorithm 1. Our dependent variable for benchmarking our method was Pearson's r correlation between the estimator's prediction and the ground truth demand values. Results of our method and those baselines described in Section 3.3 are presented in Table 1.

	Att (TRIs)	AttV	YRM	YRV	AccM	AccV	Lin	RBF	FS
Full	**0.38***	0.28*	0.04	-0.09	-0.21*	-0.02	N/A	N/A	0.00
Test	**0.38***	0.27*	0.04	-0.08	-0.18*	-0.01	0.16*	0.23*	0.00

Table 1: Results. Pearson's r between predicted demand values estimated by several methods (columns) and ground-truth demand, on the full 1200 example validation set (row 1) and the 600 examples sampled from the test set (row 2). (*) represents a significant correlation. Att (TRIs) is our proposed method based on task items, AttV is a simple variance across the attention map, YRM, YRV, AccM, AccV correspond to yaw rate and acceleration mean and variance respectively, Lin is a linear regression model, RBF is a kernel regression, and FS is file size.

As can be seen in Table 1, our method of aggregating relevance scores across task-relevant items far outperforms the other measures of demand, with its correlation being significant at the $p < e^{-17}$ level. Indeed even the more approximate attentional method, whereby the variance of the attention across space is calculated, results in a closer match to the ground truth. Somewhat surprisingly, both yaw rate measures conveyed essentially no information about the attentional demand of the situation, even though steering is the variable most associated and studied in the context of driver workload. The significant negative correlation between acceleration and demand indicates that people tend to be slowing

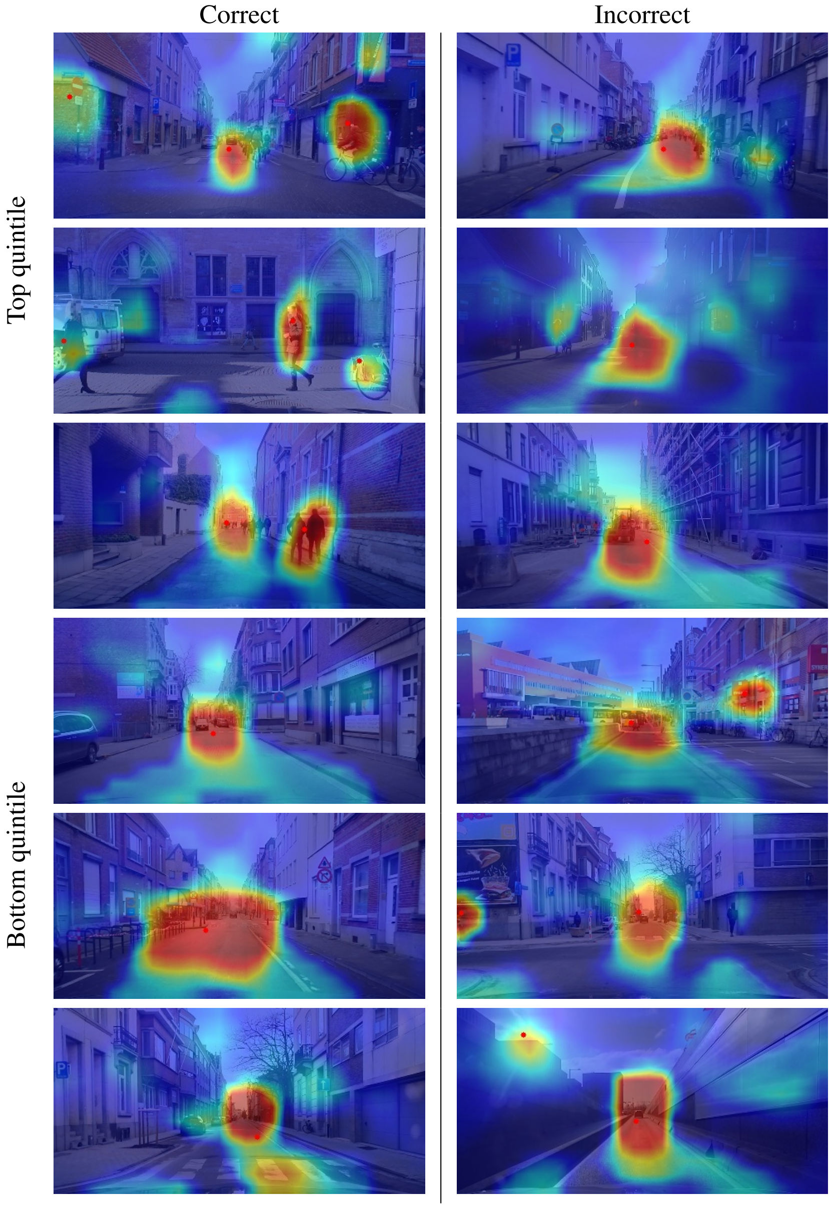


Figure 3: Examples of correctly and incorrectly estimated attentional demands. The first three rows show example scenes from the top quintile (the top 20% of scenes after sorting by ground-truth attentional demand), while the bottom three rows show example scenes from the bottom quintile (the bottom 20% least demanding scenes). Faces and registration plates have been blurred to conceal personally identifiable information. The heat map represents the class-agnostic attention field produced by the TAP model, and overlaid red dots represent identified TRIs by segmentation.

down when a difficult driving situation is ahead, a very reasonable result, while combining accelerometer and yaw rate sensor readings and learning an RBF kernel regression model only marginally outperformed the acceleration mean on the held out test set.

Some examples of correctly and incorrectly estimated attentional demands are presented in Figure 3. For instance, the example in the third row of the first column represents one of our motivating thoughts for developing this method - two people walking in unison holding hands are identified as a single task-relevant item and therefore only contribute to the demand of the situation as a single object.

7 Conclusion

In this paper we proposed an end-to-end driving architecture with an attention mechanism trained to predict the future driving command given a sequence of frames taken from a front-facing camera mounted on a car. We showed how the resulting attention maps learned on such a simple task can be interpreted and used for estimating the driver's attentional demand. We collected a validation dataset and showed how our proposed approach outperforms several baselines at the problem of attentional demand estimation.

8 Acknowledgements

This work was supported by Toyota Motor Europe. We thank Jonas Ambeck-Madsen, Gabriel Othmezourri and Deniel Olmeida Reino of Toyota Motor Europe for collecting and supplying the driving dataset, and for many interesting discussions.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[3] Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. Table-to-text: Describing table region with natural language. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[4] Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller. Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv preprint arXiv:1704.07911*, 2017.

[5] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[6] Dick De Waard. *The measurement of drivers' mental workload.* Groningen University, Traffic Research Center Netherlands, 1996.

[7] Dick De Waard, Maaike Jessurun, Frank JJM Steyvers, Peter TF Reggatt, and Karel A Brookhuis. Effect of road layout and road environment on driving performance, drivers' physiology and road appreciation. *Ergonomics*, 38(7):1395–1407, 1995.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Vérane Faure, Régis Lobjois, and Nicolas Benguigui. The effects of driving environment complexity and dual tasking on drivers' mental workload and eye blink behavior. *Transportation research part F: traffic psychology and behaviour*, 40:78–90, 2016.

[10] Felix A Gers and Jürgen Schmidhuber. Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 3, pages 189–194. IEEE, 2000.

[11] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *Advances in Neural Information Processing Systems*, pages 34–45, 2017.

[12] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):93, 2018.

[13] Xueqin Hao, Zhiguo Wang, Fan Yang, Ying Wang, Yanru Guo, and Kan Zhang. The effect of traffic on situation awareness and mental workload: simulator-based study. In *International Conference on Engineering Psychology and Cognitive Ergonomics*, pages 288–296. Springer, 2007.

[14] Simon Hecker, Dengxin Dai, and Luc Van Gool. End-to-end learning of driving models with surround-view cameras and route planners. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–453, 2018.

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[16] Jinkyu Kim and John F Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *ICCV*, pages 2961–2969, 2017.

[17] Nilli Lavie. Perceptual load as a necessary condition for selective attention. *Journal of Experimental Psychology: Human perception and performance*, 21(3):451, 1995.

[18] Nilli Lavie. Distracted and confused?: Selective attention under load. *Trends in cognitive sciences*, 9(2):75–82, 2005.

[19] Nilli Lavie. Attention, distraction, and cognitive control under load. *Current directions in psychological science*, 19(3):143–148, 2010.

[20] Nilli Lavie and Sally Cox. On the efficiency of visual selective attention: Efficient visual search leads to inefficient distractor rejection. *Psychological Science*, 8(5):395–396, 1997.

[21] Charles C Liu, Simon G Hosking, and Michael G Lenné. Predicting driver drowsiness using vehicle measures: Recent insights and future challenges. *Journal of safety research*, 40(4):239–245, 2009.

[22] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383, 2017.

[23] Gustav Markkula and Johan Engström. A steering wheel reversal rate metric for assessing effects of visual and cognitive secondary task load. In *Proceedings of the 13th ITS World Congress*. Leeds, 2006.

[24] Andrea Palazzi, Davide Abati, Simone Calderara, Francesco Solera, and Rita Cucchiara. Predicting the driver's focus of attention: the dr (eye) ve project. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[25] Luke Palmer, Alina Bialkowski, Gabriel J Brostow, Jonas Ambeck-Madsen, and Nilli Lavie. Predicting the perceptual demands of urban driving with video regression. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 409–417. IEEE, 2017.

[26] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

[27] Michael I Posner. Orienting of attention. *Quarterly journal of experimental psychology*, 32(1):3–25, 1980.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Pierre Thiffault and Jacques Bergeron. Monotony of road environment and driver fatigue: a simulator study. *Accident Analysis & Prevention*, 35(3):381–391, 2003.

[30] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[31] Radu Tudor Ionescu, Bogdan Alexe, Marius Leordeanu, Marius Popescu, Dim P Papadopoulos, and Vittorio Ferrari. How hard can it be? estimating the difficulty of visual search in an image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2157–2166, 2016.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[33] Ruohan Wang, Pierluigi V Amadori, and Yiannis Demiris. Real-time workload classification during driving using hypernetworks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3060–3065. IEEE, 2018.

[34] Christopher D Wickens. Multiple resources and mental workload. *Human factors*, 50 (3):449–455, 2008.

[35] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[36] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[37] Kristie L Young, Michael A Regan, and John D Lee. *Measuring the effects of driver distraction: Direct driving performance methods and measures*. CRC Press, Boca Raton, FL, 2009.

[38] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833. Springer, 2014.