Predicting the Perceptual Demands of Urban Driving with Video Regression

Luke Palmer*[1] Alina Bialkowski*[1] Gabriel J. Brostow[2] Jonas Ambeck-Madsen[3] Nilli Lavie[1]
[1]Institute of Cognitive Neuroscience & [2]Computer Science Department
University College London, UK
[3]Toyota Motor Europe R&D, Brussels, Belgium
{luke.palmer.12, a.bialkowski, n.lavie}@ucl.ac.uk, g.brostow@cs.ucl.ac.uk
jonas.ambeck@toyota-europe.com

Abstract

To drive safely requires perceiving vast amounts of rapidly changing visual information. This can exhaust our limited perceptual capacity and lead to cases of 'looking but failing to see'; reportedly the third largest contributing factor to road traffic accidents. In the present work we use a 3D convolutional neural network to model the perceptual demand of varied driving situations. To validate the method we introduce a new labelled dataset of approximately 2300 videos of driving in Brussels and California.

1. Introduction

Human perception is known to have a limited capacity [5], and when a task demands high levels of perceptual (or, equivalently, attentional) processing, seemingly obvious and salient objects can go completely unnoticed (a phenomenon termed inattentional blindness [24]). There are obvious implications for safety in such cases of visual failure. For example, when driving a car or piloting a plane: failing to notice a crossing pedestrian, another road user, or important sign or signal could have potentially serious consequences. Indeed, a Department for Transport report [4] found 'looking but failing to see' to be the third most commonly reported contributory factor to road accidents in Britain. Therefore, in this work we aim to identify situations in which perceptual load during the task of driving is high, and the likelihood of a costly episode of inattentional blindness is elevated.

There is currently uncertainty regarding the exact antecedents of perceptual load; that is, what elements or features of a task dictate the amount of perceptual processing required to complete it. It is similarly unknown whether it is possible to predict the load from only the visual information present in a task. In the cognitive science literature, perceptual load has up to now only been defined operationally (*e.g*. [21, 20]): for example the task of finding an object based on a conjunction of visual features (*e.g*. find the round and green object) being more demanding than single feature search (*e.g*. find the red object), or searching for a target object amongst distractor objects being more demanding when the number of distractors is increased.

Some recent work (*e.g*. Roper *et al*. [26]) has expanded upon this definition, showing that visual features of a task can be predictive of perceptual load, although this work was constrained to austere psychophysical laboratory stimuli, where visual characteristics of the task were hand-labelled by the experimenter, (*e.g*. the letters C and T were defined as not similar, whereas the letters L and T were given a 'medium' similarity relationship). It is clear that such an approach breaks down when faced with real-world tasks and perception, which operate in visually complex and dynamic conditions, where classifications of stimuli into simple object categories or similarity groups are not readily available. In order to estimate load in the complex, safety-critical task of urban driving, we therefore apply approaches rooted in computer vision and machine learning to map from raw camera output to estimates of perceptual load induced by a driving situation.

Our first conceptual step is to frame perceptual load as a subjective visual attribute of the scene, and subsequently to regress from video input to the value of this attribute; implicitly capturing the features of the driving situation which induce demands on perception and attention. We therefore constructed a large corpus of driving scene video clips captured from a dashboard-mounted camera (Section 3.1) and crowd-sourced pairwise-comparison labels between clips to obtain ground-truth load values (Section 3.2). We then implemented a 3D CNN video description network in conjunction with support-vector regression to map from video pixel information to the estimates of demand, showing these deep features to vastly outperform an object-detection-based rep-

* Both authors contributed equally.

resentation more related to cognitive science work on the antecedents of perceptual demand (Section 3.3). We subsequently show that our model, trained on European driving footage, generalises to footage collected in the USA, making comparable judgements to annotators in choosing the most attentionally demanding driving situation (Section 3.4). In Section 4 we compute importance maps of highly demanding situations, showing that the model's predictions are dependent on intuitively reasonable scene information (*e.g*. pedestrians potentially stepping in front of the car).

2. Related Work

Estimating the human perception of subjective visual attributes has been receiving increasing research attention. For example, Gygli *et al*. [11] produced a computational model of image interestingness, finding that interestingness is correlated with basic image descriptors (*e.g*. GIST [25]) as well as the unusualness of an image within the corpus. Clothing style has also been investigated: Kiapour *et al*. [16] again regressed from image descriptors of clothed people (*e.g*. HOG [6]) to style attributes such as 'hipster', 'bohemian', and 'preppy', while Kovashka *et al*. [17] leveraged a similar approach to predict the 'shininess' of shoes, among other attributes. Dubey *et al*. [8] have recently produced research in a similar domain to ours, using outdoor street-level imagery (sourced from Google's Street View) to predict the perceived value of 6 attributes of the urban area, such as safety and liveliness, and introduced the use of deep convolutional architectures to map from images to the subjective attributes of interest.

A difficulty inherent in the study of such attributes in comparison to the more common pursuit of estimating classes or values where the property is definite in nature (*e.g*. is this an image of a dog or a cat?), is that the target value is subjective. For example, what constitutes 'gothic' clothing to one annotator may not to another, and this difficulty is exacerbated when annotators are asked to produce absolute values for attributes on a scale: what does '5 out of 10' mean in terms of gothicness? It has been shown that people produce more reliable attribute estimates in such domains when asked to compare examples and rank them with respect to the attribute of interest, rather than to produce an absolute cardinal value [10]. This data collection paradigm has been widely adopted in the attribute estimation literature, with the transformation from exemplar comparisons to attribute values commonly being accomplished with the Bradley-Terry-Luce model [3, 23] or the more recent Bayesian TrueSkill algorithm [12]; we therefore adopt this methodology to build our ground-truth perceptual load in driving dataset.

The attribute prediction work cited so far deals exclusively with estimating attributes of static images however, and there is a strong likelihood that the perceptual demand of a driving situation is related to the motion of objects and features through time (*e.g*. the motion of other cars on the road). As such, we instead aim to map from video clips to the attribute of interest, where there is currently only one attempt in the literature at a similar mapping. Jiang *et al*.'s work [15] aimed to predict the interestingness of videos, as given by Flickr's 'interesting' search filter, by using SIFT [22] in combination with other image features sampled at one frame per second. Their methods of ground-truth video annotation, being effectively a black-box proprietary function, and video description, which sacrifice much temporal resolution for the cause of compactness, are however questionable in how they relate to the understanding of video information. In this work we therefore build a perceptual load labelled ground-truth dataset from scratch using the pairwise comparison method previously applied successfully in the image domain, and furthermore use a state-of-the-art video description network (C3D [28]) to extract useful spatio-temporal representations. C3D is a 3-dimensional convolutional neural network, the architecture of which captures fine-scale variations in time as well as space, and has shown excellent performance in the domain of human action recognition, where representations of object motion patterns are known to be critical to the task. This representation is also compared against a model based on explicit car and pedestrian detections (Detection Bank; [1]), more in keeping with previous cognitive science literature investigating the role of task-related objects in dictating the perceptual demand of a task.

While there has of course been much research directly related to understanding the visual driving scene, such as lane detection [29, 13], street sign detection [14], and the estimation of ego-motion [30, 27], much of this is geared towards automated vehicle operation $per\ se$. The novelty of the work here is the estimation of a human-centric attribute with real-world applications for in-vehicle warning systems, which could signal reduced ability to detect critical stimuli for example; or for human-machine interaction in the driverless car era, where human control take-overs could be dependent on the driving situation with respect to perceptual demands.

3. Method

We characterise perceptual load as an attribute of the driving scene, and then associate perceptual load values to segments of driving footage. Once a dataset of video segments with associated load values is obtained, the modelling task becomes a regression problem between video segments and load values. In the following sections we describe our data collection procedure, followed by methods used to: 1) obtain consistent estimates of perceptual load from the combined judgements of many human annotators, 2) represent

video segments with compact semantically informative descriptors, and 3) map those descriptors to perceptual load values.

3.1. Data collection

We created two labelled datasets to develop and evaluate our models: one collected from scratch in Brussels (Section 3.1.1), and one using existing footage from California (Section 3.1.2) for a model generalisability study.

3.1.1 Brussels dataset

We captured a large corpus of driving scene video clips using a dashboard-mounted camera. The data collection vehicle was a Toyota Prius equipped with a high-quality dashboard mounted camera (Point Grey Flea3 model) and high-precision global positioning system (GPS). The camera was centrally placed on the dashboard facing forwards and captured 75° of visual angle at 30 frames per second. No zooming, focus, or gain adjustments were made during recording, focus was set at infinity, and the gain and shutter speed were locked. Camera aperture was opened at the beginning of each recording session as much as possible without allowing white objects in the scene to saturate. The recorded raw high-resolution images were later compressed using ffmpeg to a MPEG 4 video format at a resolution of 640×512 pixels. The GPS device recorded longitude, latitude, and altitude data at an average precision of 0.5 m at a rate of 180 Hz, synchronised with the camera shutter (6 GPS samples per video frame). Example video frames are presented in Figure 1.

Figure 1: Example frames from video captured in Brussels city centre

Two data collection routes were designed in and around central Brussels, Belgium. Routes were designed to capture variation in vehicle and pedestrian density throughout the day, and contained a variety of common urban road types: intersections, junctions, roundabouts, and straight roads. Each route was completed 5 times on separate, fine-weather, days. The 10 total runs resulted in the collection of over 12 hours of high-quality video and GPS data.

Each collected video was then viewed and manually partitioned into individual sequences according to several features of the driving situation. Any periods of very slow ego-motion were removed from the dataset (i.e. the data collection car travelling at a speed of less than approximately 5 miles per hour). There were 6 features used to describe the videos, which are detailed in Table 1.

Table 1: Features of the driving scene used to describe captured video and partition into individual sequences.

Feature	Possible values
Current road layout	Straight road; intersection (including junctions); roundabout
Carriageway type	Dual or single
Number of lanes	Integer value (from 1)
Current car manoeuvre	None; right turn; left turn
Pedestrian density	Integer value from 0 (no pedestrians in view) to 3 (large numbers of pedestrians)
Vehicle density	Integer value from 0 (no vehicles in view) to 3 (large numbers of vehicles)

A new sequence was declared and labelled when one or more of the features of the scene changed from the previous sequence. For example, if a group of pedestrians appeared on the pavement after exiting a building, where previously there had been no pedestrians in view, then, all else in the scene being equal, a new sequence was declared and the pedestrian density value increased from 0 to a higher value (depending on the number of pedestrians). Through this system a number of sequences were created, each labelled with the 6 features described above. The length of the partitioned sequences ranged from 2s to 18s.

Given the labelled sequences, a heuristic method was implemented to further partition the sequences into a selection of 2s video clips which would become the experimental dataset. Two-second clips of a sequence were more likely to be included in the dataset if they formed a grouping with clips from other sequences recorded at the same location; 2s clip groups were then more likely to be included if there was a high variance of pedestrian and/or vehicle density within that group of clips. Groupings of 2s clips at a single location were formed using GPS data: if the car position was within 10m for a duration of at least 1s across a pair (or more) of sequences then 2s clips were extracted from those sequences and formed a group at that location. Each group was then given a score dependent on the variance of pedestrian and vehicle densities of clips within that group,

$$score(G) = |G| \cdot (var_G[d_p] + var_G[d_v]), \qquad (1)$$

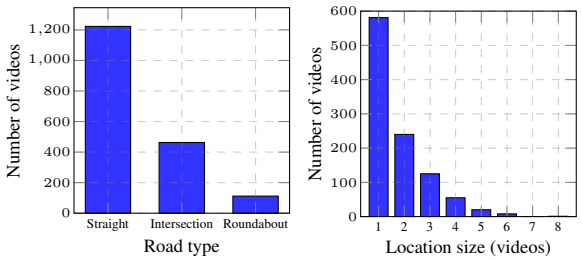

Figure 2: Statistics of the Brussels Dataset. On the left, the types of road situation in the dataset by frequency, and on the right the number of videos per location group size (*e.g.* there were 240 videos matched with one other video at the same location).

where G refers to the clip group, $|G|$ refers to the number of clips in the clip group G, and d_p and d_v refer to pedestrian and vehicle densities, respectively. The final dataset was then selected as the set of clips which maximised this score across possible groupings in a greedy fashion, resulting in a total of 1809 distinct 2s clips. Figure 2 displays descriptive statistics of the data set; the number of videos per location type and the number of location matched videos per size of location group.

3.1.2 Caltech validation dataset

To provide a dataset with different driving scenery for validating our model, we extracted a dataset of 2s clips from the Caltech Pedestrian Database [7]. The original database consists of 137 minute-long segments of driving footage captured around California and contains the bounding box annotations of pedestrians in every frame for the task of pedestrian detection. The scenery varies significantly from the Brussels dataset, including a busy airport terminal, highway driving and downtown and suburban Californian streets.

To provide an even distribution of video clip features, the footage was manually labelled with pedestrian and vehicle density, similarly to the Brussels dataset, as well as marking when the car was stationary (to exclude from the clip selection). Each pedestrian/vehicle density combination was then randomly and evenly sampled to provide a validation dataset of 511 videos.

3.2. Obtaining ground-truth perceptual load values

We employ a pairwise comparison methodology in order to assign ground-truth perceptual load values to each video in our datasets using the TrueSkill algorithm [12]. Using TrueSkill, the perceptual load of each video is represented as a Gaussian distribution, N(μ, σ), where μ represents the current estimate of the perceptual load, and σ represents the algorithm's current uncertainty regarding that estimate. After each comparison, the load distributions are adjusted. In our implementation, values for each video were initialised, before any comparisons were made, at $\mu = 25$ and $\sigma = 8.33$ (following Herbrich *et al*. [12]). After a sufficient number of comparisons, ratings became stable; this occurs at approximately 30 to 40 comparisons per stimulus in most applications. The μ of a video's load distribution was then taken as the ground-truth perceptual load value for that video, resulting in a dataset of video and load value pairs suitable for regression analysis.

A web-application was created to deliver the comparison task interface to participants through a web-browser as in Figure 3. Participants were sat at IBM PCs, with 24" monitors, equipped with Google Chrome software to view the pairwise comparison web-application. The data collection was split into two phases – one for Brussels and Caltech – using the same pairwise-comparison methodology, but providing independent ratings (i.e. not comparable across sets). In both data collection phases it was ensured that participants held a full driver's license. Subjects viewed pairs of video clips and were instructed to indicate which situation depicted by the video clips would require the greatest demand on attention if they were driving in that situation. This concept of attentional demand fits the operational definition of perceptual load put forth by Lavie [19] and is readily explained to laymen. In verbal instructions to the participants this was also explained by example, for instance: "in which driving situation would you be more likely to hush a talking passenger?".

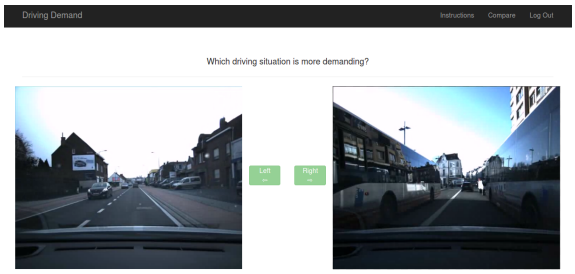

Figure 3: Pairwise-comparison labelling platform.

For the Brussels dataset, 83 participants were recruited via crowdsourcing company Pallas Ludens and paid 20EUR/hour for participation. Each participant performed pairwise comparisons for two 1-hour sessions on separate days and performed the comparison tasks under the supervision of Pallas Ludens at a facility in Germany. Each video was compared to another video 70 times, resulting in a total of 63,315 driving situation comparisons. The TrueSkill algorithm was applied to these comparisons to acquire an estimate of perceptual load level for each 2s video depiction of a driving situation. Figure 4 shows the histogram of perceptual load values after all comparisons were processed by the algorithm.

For the Caltech validation dataset, 82 participants were recruited in-house and paid £8-12/hour for participation. Each participant performed 30 minutes of annotation under supervision. A total of 7354 comparisons were collected on the Caltech dataset.

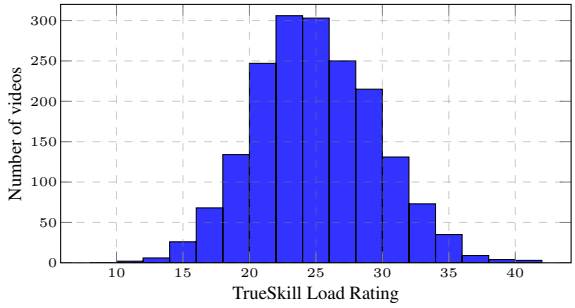

Figure 4: Histogram of perceptual load values in the Brussels video set, as estimated by the TrueSkill algorithm.

3.3. Extracting semantics from video

To represent task-relevant information in the driving videos, we implemented an object-centric baseline video representation: Detection Bank (DB; [1]). DB, in keeping with cognitive science work linking the number of relevant objects in a task to the perceptual demand of that task, encodes the location and counts of car and pedestrian detections throughout a video. Cars and pedestrians were detected using deformable parts model detectors (DPM; [9]) trained on the VOC2012 dataset. For each video frame, the DB representation computes the following detection statistics (per object category) for each grid cell in a spatial pyramid (entire frame, 2×2 grid, and 4×4 grid): the sum of scores of detections within that cell (above a detection threshold of -0.5), the number of detections, and a single bit that indicates whether or not there is a detection within that cell. By mean and max pooling these statistics temporally across frames we obtain a meaningful video-level representation capturing, *e.g*. the maximum number of detections, the average number of detections, and an empirical estimate of the detection probability for each grid cell and object category. This scheme results in a 252D feature vector for each video.

We also implemented a state-of-the-art CNN-based video representation scheme. Given the success of training CNNs on image based tasks such as object recognition [18] a natural extension is to the domain of video. A video is problematic for a naïve CNN learning approach due to the associated increase in number of learnable parameters. However, there also exists temporal redundancy in video (e.g. the appearance of an object will not change much frame-to-frame), and therefore the question of efficiently combining information across the temporal dimension has recently received much attention.

A successful approach to combining temporal information was introduced by Tran *et al*. [28]. Instead of combining information across multiple static representations, Tran and colleagues alter the convolutional filters themselves to incorporate temporal information. They parameterise 3-dimensional convolution filters at the earliest layers. On the Sports1M dataset, their 3-dimensional CNN network (or, C3D network), consisting of 8 convolutional layers (see Figure 5), achieved state-of-the-art performance of 46% classification accuracy. Furthermore, Tran *et al*. [28] found that video representations extracted from the first fully connected layer of the C3D network achieved state-of-the-art performance on the more general UCF-101 action recognition dataset. The convolutional filter weights learned using Sports1M videos therefore capture the essence of many motion based activities and concepts in unseen videos. As such we implement C3D with the aim of describing the spatio-temporal information present in driving scenarios to predict perceptual load.

Figure 5: C3D architecture. Each Conv layer implements $3\times3\times3$ 3D convolutional filters, and each pooling operation takes a maximum across $2\times2\times2$ cells. The number beneath the 'ConvXy' text refers to the number of feature maps in that layer (from [28]).

We extracted C3D features [28] for our video sets using the Sports1M pre-trained 8-layer 3-dimensional convolutional neural network. The descriptor we extracted for each given video is taken from the first fully-connected network layer, resulting in a 4096D vector. The network was realised using the $Lasagne$ and $Theano$ Python frameworks.

3.3.1 Video regression

For both the DB and C3D video representations we split the Brussels data into the same random 1/3 testing and 2/3 training sets (603 and 1206 videos respectively). For the DB representation we fitted a linear ridge regression model (with L_2 regularisation penalty of 0.1) to provide an easily interpretable baseline model of perceptual load. On the held out test set, the model predicted perceptual load value with a coefficient of determination, R^2, of 0.24, or equivalently, a correlation of 0.49. The load estimates from the model lead to an ordinal ranking of the test set exemplars which correlates with the ground-truth TrueSkill perceptual load estimates with a Kendall's τ of 0.321. The learned regression weights for object detections in the 4×4 grid of frame regions are shown in Figure 6.

For the dense C3D representation, after extracting the 4096D feature vectors for each video, we learned a map-

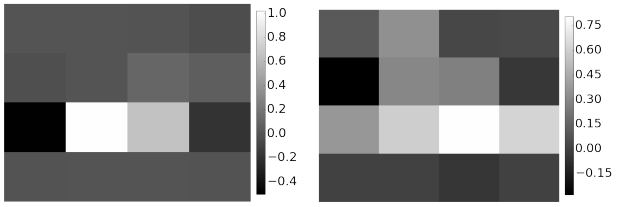

Figure 6: Regression weights for number of car (left) and pedestrian (right) detections for the Detection Bank representation (at the 4×4 grid resolution). Intuitively given the right-lane driving rules in Brussels, the model associates pedestrian detections close to the data collection car on the right-hand side with high perceptual load, and similarly associates oncoming cars driving in the left lane with high load.

ping from features to a perceptual load value estimate using RBF-kernel support vector regression (SVR). To tune the regression penalty, C, and kernel width, γ, parameters, a tree of Parzen estimators (TPE) sequential model-based optimisation routine was run for 500 iterations on the training set, maximising the 3-fold cross-validation R^2. Hyperparameter tuning was achieved using the $hyperopt$ Python package [2].

After training the model on the full training set using the best found configuration, an R^2 value of 0.56 was achieved on the unseen 603-exemplar test set (equivalent to a correlation of 0.75; see Figure 7), far surpassing the explicit object-based DB model. The results indicate a strong correlation between the model's estimates of perceptual demand in driving and those of human labellers. This is confirmed in terms of agreement of ratings relative to the ground truth load ratings as shown in Figure 8 – the model trained on the Brussels training set has an accuracy of 76.7%, which is similar to the average agreement of human raters (73.9%), as described in the following section.

3.4. Generalisation of the C3D model to Caltech dataset

To evaluate the generalisability of our model to other driving scenes, we used the same hyper-parameters and retrained the regression model on the whole Brussels dataset, and then evaluate its performance on the Caltech validation set. As the Caltech videos were not compared against the Brussels videos, the TrueSkill perceptual load ratings of the two video sets are incompatible, e.g. a middle-rating in Brussels may not translate to the same level of perceptual load of a middle-rated video from the Caltech set. We instead evaluate performance by computing the accuracy of the model's *relative* predictions versus the relative prediction obtained using the ground-truth TrueSkill ratings of that dataset.

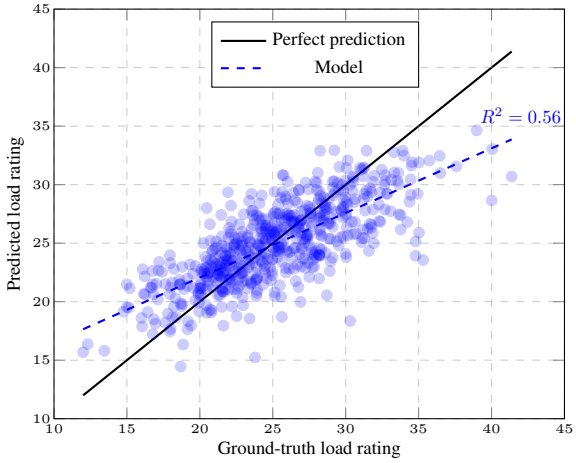

Figure 7: Regression performance using C3D features to predict load on the test set. Each blue marker represents a test set exemplar - its position on the x-axis is the ground-truth TrueSkill estimate of perceptual load, while the y-axis position is its predicted perceptual load according to an SVR model using raw C3D features, trained on the full training set (1206 examples). The $y = x$ black line represents a model with perfect predictive power (i.e. 100% of variance explained by the model); the dotted blue line represents the fit of the trained support vector regression model.

The accuracy of the model's predicted load ratings versus the ground-truth TrueSkill ratings was calculated by simulating the comparison of every possible combination of video pairs and summing the number of concordant pairs divided by the total number of pairs. This is equivalent to $\frac{(\tau+1)}{2}$, where τ is the Kendall Tau rank correlation coefficient. We compare this model performance metric to the same metric calculated for individual human labellers i.e. how well do their independent judgements align with judgements based on the overall TrueSkill load ratings. The results are presented in Figure 8.

The results in Figure 8 show that our model generalises quite well to a completely different driving environment (U.S. vs European streets) with only a slight degradation in performance, showing similar agreement with TrueSkill to the average human rater. Note that while the TrueSkill ratings were computed from all the rater's comparisons, labeller "accuracy" is below 100% due to non-transitivity and differences between rater's judgements. This indicates that there is some difference between what raters consider more demanding on their attention; which may also be exacerbated by the fact that a comparison decision was forced when many videos could be of very similar or equal perceptual load.

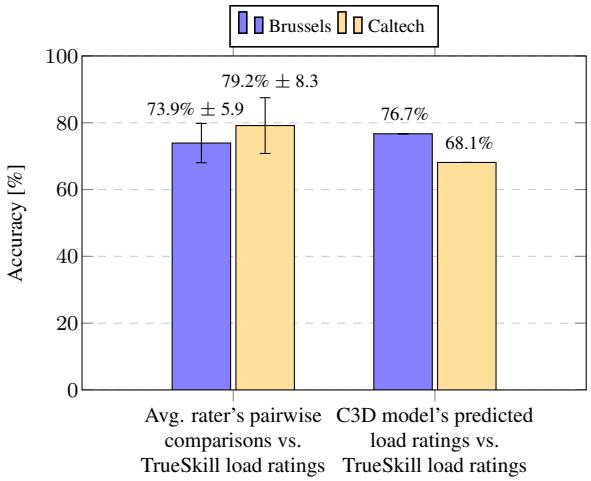

Figure 8: Model performance (right) vs. human labellers (left) on the Brussels and Caltech datasets.

4. C3D model visualisation

To understand what visual features the model finds indicative of perceptual load, we extend the approach of Zeiler and Fergus [31] for visualising instance-specific importance maps to the domain of video. We systematically remove information from the video by replacing a cubic spatio-temporal region of the video with a mid-grey box (see Figure 9), and observe how the estimated load varies. If a region is removed from the video and results in lower regressed load, it implies that the area contains information indicative of high perceptual load.

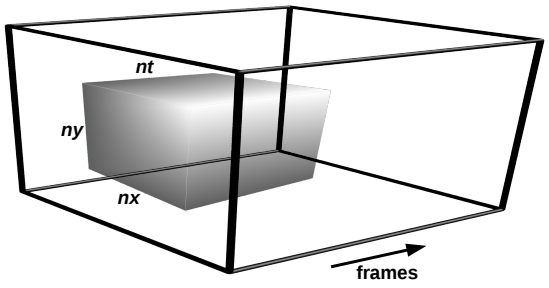

Figure 9: Occlusion-based visualisation method. The large cuboid represents the video in terms of a volume (or, equivalently image frames in time). A smaller mid-gray cuboid of dimension $nx \times ny \times nt$ is placed within the video, overwriting the original video information. The resulting occluded video is run through the trained perceptual load prediction model. A relevance map of the video can then be created by repeating this process with the occluding cuboid placed at different positions in the video cuboid. In the reported visualisations, we set $nx = ny = 150$ pixels and $nt = 15$ frames.

In our implementation of the visualisation method we generated 6000 occluded videos, with the occlusion being centred at equally spaced points in each dimension (points sampled in each $x \times y \times t$ dimension: $20 \times 20 \times 15$). A visualisation video volume was then generated by linearly interpolating the computed model estimates to the full resolution of the original video. Results of this procedure for some demanding driving situations can be seen in Figure 10.

From Figure 10 it is evident that the model's prediction in these high load settings is dependent on information which is intuitively demanding in a driving situation. For example, situations where a car crosses the path of the driver, or pedestrians approach a pedestrian crossing, critically require that the driver attend to, perceive, and potentially react to these events. An interesting point to note here is that the model has learned to identify these key driving concepts of car and pedestrian purely from regressing to ground-truth TrueSkill perceptual load estimates, in a weakly supervised fashion.

5. Conclusion

In this work we developed a method for predicting the level of perceptual demand in the complex task of urban driving. Through casting perceptual load as a subjective attribute of the visual scene, a model was learned to map from raw video input to aggregated judgements of a large batch of crowdsourced labels. The model shows near-human level performance in judging the most demanding driving situation between a pair of presented driving videos, a result which generalises to driving scenes collected in a different location to the training data. A model visualisation method was also developed for the video domain, showing the relevant regions for perceptual demand estimation.

Acknowledgements

We would like to thank Freya Marijatta, Mahmoud El Bahnasawi and Tim Sandhu for their assistance in data collection and labelling.

References

[1] T. Althoff, H. O. Song, and T. Darrell. Detection bank: an object detection based video representation for multimedia event recognition. In *International Conference on Multimedia*. ACM, 2012.

[2] J. Bergstra, D. Yamins, and D. D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *ICML*, 2013.

[3] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4), 1952.

[4] I. D. Brown. *Review of the 'Looked but failed to see' accident causation factor*. Dept. for Transport, London, 2005.

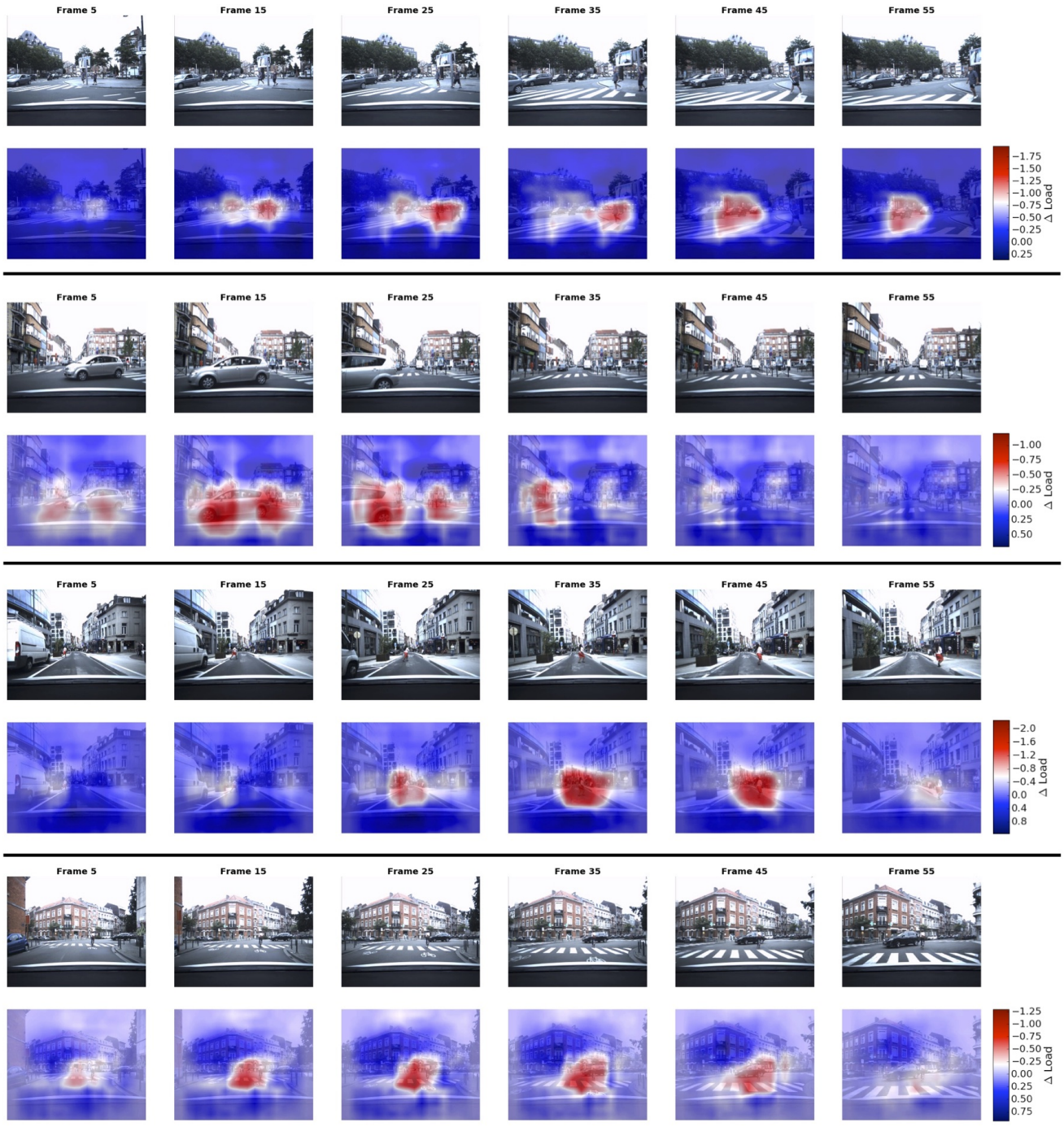

Figure 10: Examples of perceptual demand maps through video frames. The top row of each example contains the original frames for each video, while the lower row superimposes the map. The colours indicate how the predicted load varies when occluding different regions of the video with respect to the original un-occluded video's perceptual load prediction. Red corresponds to regions which, when removed from the video (replaced with a grey cuboid), result in relatively lower load estimates, indicating that the spatio-temporal area contains visual information demanding high levels of perceptual processing during driving.

[5] N. Cowan, E. M. Elliott, J. S. Saults, C. C. Morey, S. Mattox, A. Hismjatullina, and A. R. Conway. On the capacity of attention: Its estimation and its role in working memory and cognitive aptitudes. *Cognitive psychology*, 51(1), 2005.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, 2009.

[8] A. Dubey, N. Naik, D. Parikh, R. Raskar, and C. A. Hidalgo. Deep learning the city: Quantifying urban perception at a global scale. In *ECCV*. Springer, 2016.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9), 2010.

[10] Y. Fu, T. M. Hospedales, T. Xiang, J. Xiong, S. Gong, Y. Wang, and Y. Yao. Robust subjective visual property prediction from crowdsourced pairwise labels. *TPAMI*, 38(3), 2016.

[11] M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, and L. Van Gool. The interestingness of images. In *ICCV*, 2013.

[12] R. Herbrich, T. Minka, and T. Graepel. Trueskill^{TM}: A bayesian skill rating system. In *NIPS*, 2006.

[13] A. B. Hillel, R. Lerner, D. Levi, and G. Raz. Recent progress in road and lane detection: a survey. *Machine vision and applications*, 25(3), 2014.

[14] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2013.

[15] Y.-G. Jiang, Y. Wang, R. Feng, X. Xue, Y. Zheng, and H. Yang. Understanding and predicting interestingness of videos. In *AAAI*, 2013.

[16] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg. Hipster wars: Discovering elements of fashion styles. In *ECCV*, 2014.

[17] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, 2012.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

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

[20] N. Lavie. Distracted and confused?: Selective attention under load. *Trends in cognitive sciences*, 9(2), 2005.

[21] N. Lavie and Y. Tsal. Perceptual load as a major determinant of the locus of selection in visual attention. *Perception & Psychophysics*, 56(2), 1994.

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004.

[23] R. D. Luce. On the possible psychophysical laws. *Psychological review*, 66(2), 1959.

[24] A. Mack and I. Rock. *Inattentional blindness*, volume 33. MIT press Cambridge, MA, 1998.

[25] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3), 2001.

[26] Z. J. Roper, J. D. Cosman, and S. P. Vecera. Perceptual load corresponds with factors known to influence visual search. *Journal of experimental psychology: human perception and performance*, 39(5), 2013.

[27] G. P. Stein, O. Mano, and A. Shashua. A robust method for computing vehicle ego-motion. In *IEEE Intelligent Vehicles Symposium*, 2000.

[28] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[29] Y. Wang, E. K. Teoh, and D. Shen. Lane detection and tracking using b-snake. *Image and Vision computing*, 22(4), 2004.

[30] K. Yamaguchi, T. Kato, and Y. Ninomiya. Vehicle ego-motion estimation and moving object detection using a monocular camera. In *ICPR*, 2006.

[31] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*. Springer, 2014.