IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 12, 2020, accepted November 26, 2020, date of publication December 1, 2020,
date of current version December 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041606

Interpretable Global-Local Dynamics for the Prediction of Eye Fixations in Autonomous Driving Scenarios

JAVIER MARTÍNEZ-CEBRIÁN, MIGUEL-ÁNGEL FERNÁNDEZ-TORRES, AND FERNANDO DÍAZ-DE-MARÍA, (Member, IEEE)
Department of Signal Theory and Communications, Universidad Carlos III de Madrid, 28911 Leganés, Spain
Corresponding author: Javier Martínez-Cebrián (javiermcebrian@gmail.com)

This work was supported by the Spanish Ministry of Economy and Competitiveness under National Grant TEC2017-84395-P.

ABSTRACT Human eye movements while driving reveal that visual attention largely depends on the context in which it occurs. Furthermore, an autonomous vehicle which performs this function would be more reliable if its outputs were understandable. Capsule Networks have been presented as a great opportunity to explore new horizons in the Computer Vision field, due to their capability to structure and relate latent information. In this article, we present a hierarchical approach for the prediction of eye fixations in autonomous driving scenarios. Context-driven visual attention can be modeled by considering different conditions which, in turn, are represented as combinations of several spatio-temporal features. With the aim of learning these conditions, we have built an encoder-decoder network which merges visual features' information using a global-local definition of capsules. Two types of capsules are distinguished: representational capsules for features and discriminative capsules for conditions. The latter and the use of eye fixations recorded with wearable eye tracking glasses allow the model to learn both to predict contextual conditions and to estimate visual attention, by means of a multi-task loss function. Experiments show how our approach is able to express either frame-level (global) or pixel-wise (local) relationships between features and contextual conditions, allowing for interpretability while maintaining or improving the performance of black-box related systems in the literature. Indeed, our proposal offers an improvement of 29% in terms of information gain with respect to the best performance reported in the literature.

INDEX TERMS Top-down visual attention, eye fixation prediction, context-based learning, interpretability, capsule networks, convolutional neural networks, autonomous driving.

I. INTRODUCTION

The way contemporary Computer Vision systems represent our world seems progressively further from being understood by humans. Both the performance and the complexity of feature learning methods, which derives from the application of Deep Learning (DL) and Convolutional Neural Networks (CNNs) to compelling but challenging vision tasks such as object recognition [1] and tracking [2], or anomaly detection in video surveillance scenarios [3], increase at the same time. These powerful recent techniques lead to an apparent paradox: Although CNN-based models are inspired by how visual processing works in living organisms [4], which should bring us to a closer insight of this function, its intricacy, either in humans or automated systems, places us in an even more distant position.

Here arises one of the reasons why interpretability is a paramount property for automated applications, especially in those that involve several cognitive operations of the brain, as the visual attention task [5]. Humans visually attend to every situation based on its context, perceiving only relevant information like a perfect synthesis machine [6]. Contextual information could be expressed to a greater or lesser extent attending to several features: size, texture, motion, semantics, etc. [7]. The relationship between these visual patterns is not always the same, but depends on each scene as a whole, so it occurs in a dynamic way. This information framework, as modeled by humans, can be defined as the combination of latent representations of high level concepts, which are characterized by their visual structure [8], [9]. It is also

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

noteworthy the role of eye movements in visual perception [10]. The position and duration of eye fixations highlight the most relevant elements of the scene when performing tasks such as cooking or chess-playing, and can even constitute a way to interpret thoughts [11].

Interpretability is also a desirable property for DL systems when the degree of robustness of the application domain to errors is low [12]. The cost of making wrong predictions in legal, healthcare or transportation domains can be very high. Consequently, it is necessary to develop new methodologies which enable humans to understand, in a straightforward way, the decisions made by models with a black-box nature. This would also provide an enhanced control over unexpected events, at the same time it helps for efficiently discovering extra useful information to improve the systems behavior. For that purpose, the possibilities of eye tracking for assistance in real applications such as industry control [13] and video surveillance [14] have been assessed, coming to the conclusion that there is a strong correlation between fixation sequences of different users carrying out the same task. Autonomous driving systems could be more expressive if they were able not to only estimate but also report about visual attention, providing both the users and the monitorization system with the most significant spatio-temporal locations (e.g. pedestrians, traffic signs and lights, walls, etc.) in safety-critical situations (e.g. pedestrians suddenly crossing the street, overtakings, stopped vehicles on the road or the hard shoulder, etc.) [15]. Drivers would feel safer, at the same time the system would be more transparent and reliable [16].

Capsule Networks [17], [18] have been recently shown to be a promising technology which allows us to explore new possibilities in the Computer Vision field. They make the assumption that the visual world can be modeled by a parse-tree like structure. This latent representation is dynamically instantiated for each new input stimulus, for example, at each new frame of a video sequence. The information flows through the dynamic structure by means of routing elements called capsules. This mechanism could be used to model every hierarchical structure designed by the human being with the aim of understanding the visual world: contexts or scenarios, as well as intermediate concepts. Dynamic routing coefficients between capsules would be designed in this regard to model both global and local spatial relationships.

Contexts could be broken down into multiple intermediate level concepts called conditions. Conditions, being representative of the contextual information, could be understood as the combination of several visual features. Visual attention for autonomous driving is such an attractive challenge to solve due to its complexity and diversity of scenarios. A driving scenario can be defined as the combination of some contextual conditions as follows: a sunny morning in downtown, a cloudy evening in the countryside or a rainy night on a highway. We can model these conditions by considering their global and local relationship with some features, which can be extracted using CNN-based architectures.

Following the previous statements, we propose a multi-source encoder-decoder network with a fusion stage based on Capsule Networks, which aims to estimate visual attention in autonomous driving scenarios. Our hierarchical system, which we have called Global-Local Capsule Network (GLCapsNet) and is shown in Figure 1, makes use of a dynamic routing mechanism similar to the one presented in [19] in order to define contextual conditions as the combination of latent representations of visual features. The use of a multi-task loss function allows the system to learn either to model conditions presence, both locally and globally, or to estimate visual attention, based not only on the previously determined contextual structure, but also in human fixations, which are recorded using wearable eye tracking glasses under different real driving conditions. In summary, this article makes the following contributions:

1) We introduce for the first time, to the best of our knowledge, a hierarchical framework based on capsules for context-aware visual attention modeling. Its structure is designed at both concept (global) and pixel (local) levels. This implies a new definition of capsules, as well as modifying the dynamic routing mechanism and defining the contextual conditions' presence.
2) We propose an emphasis function to enhance conditions prediction, which improves the generation of the internal structure of the model by emphasizing the difference between local routing coefficients.
3) We carry out an in-depth analysis of our proposal, comparing it with several methods in the literature for autonomous driving. In contrast to these approaches, the global-local definition of capsules enables, for the first time, the interpretability of the model. For this reason, we additionally analyze its results discovering, on the one hand, at global level, the contribution of visual features to contextual conditions and, on the other hand, at local level, how features are at the spatial locations highlighted for a given condition.

The rest of the paper is organized as follows. First, work related with visual attention, autonomous driving and Capsule Networks is summarized in Section II. Then, the architecture and formulation of the GLCapsNet model proposed is described in Section III. Later, results on visual attention estimation, together with the interpretability analysis provided by our system for autonomous driving scenarios are covered in Section IV. Finally, conclusions and future lines of research are exposed in Section V.

II. RELATED WORK

A. VISUAL ATTENTION

Visual attention is an appealing line of research which applies to many Computer Vision applications [20]: image segmentation, image matching, super-resolution and object detection, among others. Its main objective can be summarized as searching space regions in an image or a video sequence which are useful or relevant to observers.

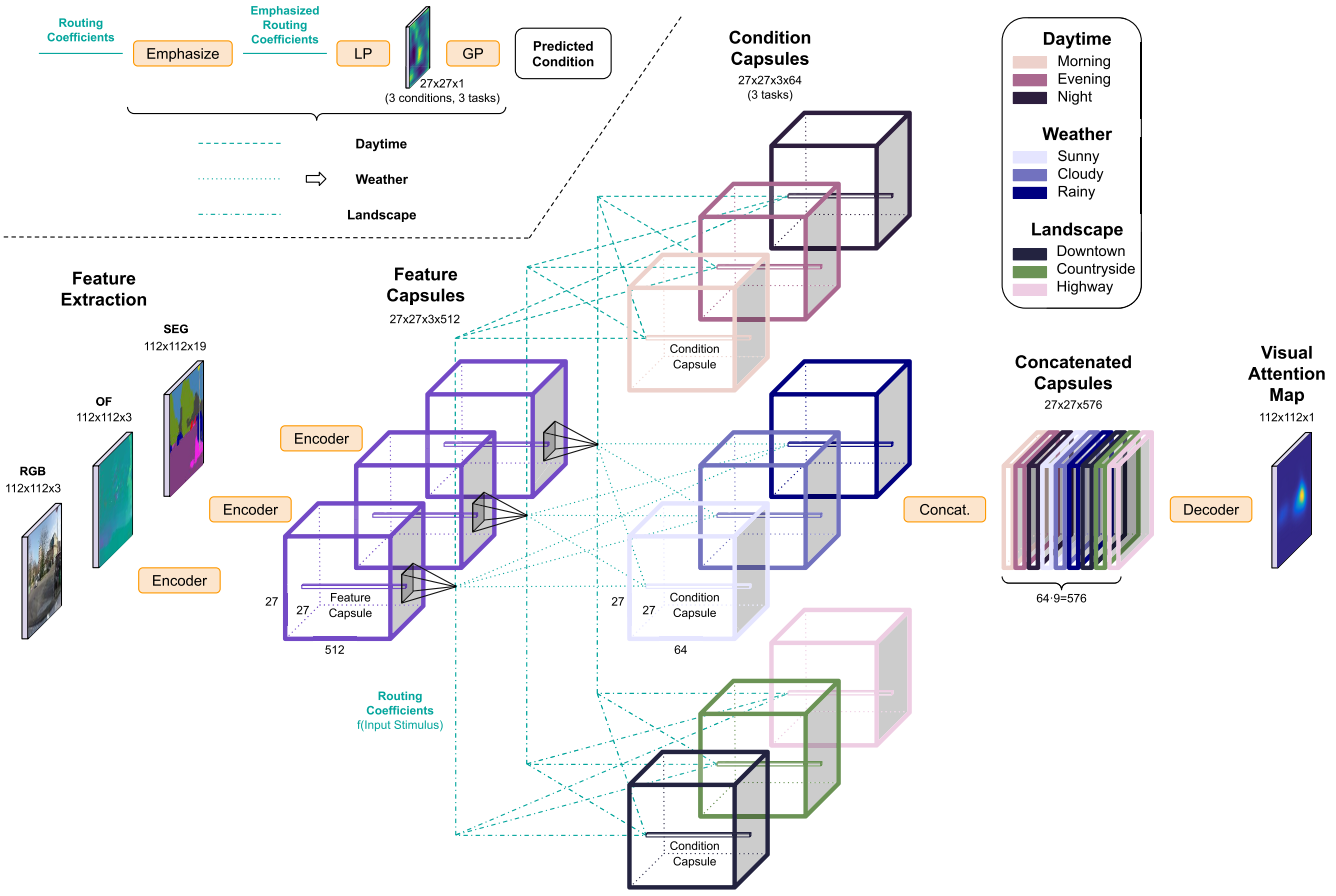


FIGURE 1. Global-Local Capsule Network (GLCapsNet) block diagram. It models visual attention based on several contextual conditions of the scene, which are represented as combinations of several spatio-temporal features (RGB, Optical Flow and Semantic Segmentation). Its hierarchical multi-task approach routes Feature Capsules to Condition Capsules both globally and locally. On the one hand, the term "global" in Global-Local Capsules is referred to the conditions' prediction endpoints as they are defined for the whole scene, as well as their intermediate building blocks (feature-condition relationships at frame level) and the conditions' masking applied during training. On the other hand, "local" is referred to pixel-wise relationships between capsule layers, which could be considered as specialized visual attention maps for each feature-condition pair. Non-capsule transformations are in orange, capsule types (blocks of vector capsules) are cube-shaped and Condition Capsules are colored based on the legend, pyramids stand for constrained matrix multiplication (Section III-D), routing coefficients distinguish between conditions' prediction tasks via different line patterns (dashed, dotted, etc.), *Emphasize* is defined in (5), LP stands for *LocalPresence* (8) and GP for *GlobalPresence* or conditions' predictions (9).

Different factors, types and applications are considered when classifying visual attention models. The most common distinction is Bottom-Up (BU) [21] or *stimulus driven* v.s. Top-Down (TD) [22] or *goal-driven*. Some models are spatial-based, while others are both spatial and time-based. We can also differentiate between models oriented to the search of salient regions [23] or objects [24], [25], consider different features to model visual attention [26], [27], or even design models tailored to particular tasks [28].

We can distinguish between classical and modern visual attention approaches. The former refers to feature engineering models, while the latter usually makes use of DL architectures. Borji and Itti describe in [20] an extensive state-of-the-art in classical visual attention modeling. By contrast, the reader is referred to [29] for references on modern approaches. There are lots of visual attention models proposed in the literature, and many of them achieve good results in eye fixation prediction. However, most of them are BU approaches without a specific goal, which require a higher level visual understanding to reach human-level accuracy. Indeed, they are not able to determine the semantic meaning of concepts or action cues in rich scenes yet, or even the relative importance of image regions and objects [30]. Moreover, almost none of them provide interpretability properties, which would allow us to understand not only how we predict where people look, but also how visual attention is deployed in humans [31]. Last but not least, there is still a lack of task-oriented databases annotated with eye-tracking data, which would enable to assess the usefulness and performance of existing architectures in potential applications such as medical imaging or video surveillance. Incorporating additional ground truth (GT) information to these datasets, such as image-level labels for classification, pixel-level labels for segmentation or frame-level labels for action recognition, would significantly increase their value when pursuing the design of explainable models.

In an effort to contribute to visual attention understanding in real settings, we propose a TD system to carry out an

autonomous driving task, which is able to offer interpretation about its predictions by means of Capsule Networks [17], [18]. In addition, since the autonomous driving task takes place in diverse scenarios, it is required a hierarchical model that first understands the context in which attention will be guided. Fernández-Torres *et al.* present in [32] their hypothesis about visual attention, similar to the one stated in this work: when looking at particular contexts, visual attention may be attracted by different events or elements in the scene. They try to discover latent sub-tasks that guide the later processing to the areas where those occur, which are modeled as a combination of spatio-temporal features.

B. VISUAL ATTENTION FOR AUTONOMOUS VEHICLES

Autonomous driving is a complex task where four subsystems can be distinguished [33]. First, the sensing field, which involves several onboard vehicle sensors to monitor the environment, such as LIDARs, mono or stereo video cameras, or even short/long-range RADAR systems and ultrasonic sensors. In this work, we assume the use of video cameras. Second, the perception module [34], which processes and fuses the measurements coming from the sensors in order to provide the vehicle with relevant information about the driving context (e.g. velocities, free drivable areas, obstacles' locations, etc.). Third, the path planning module [35], which determines the motion and the optimal path that the vehicle has to follow in order to avoid obstacles and reach a target location, based on the outputs of the perception module. Finally, the control module [36], which commands the vehicle to execute planned actions, such as accelerating, braking and steering, among others.

Many computer vision algorithms are part of the perception subsystem [37]: object detection [38] and semantic segmentation [39], 2D [40] and 3D reconstruction [41], [42], optical flow [43], [44], tracking [45], etc. Specifically, visual attention can play a relevant role in most of the essential functions performed by autonomous vehicles since it has the purpose of filtering the huge amount of data accessing the perception module, thus guiding the car's path planning and control modules. However, there are hardly any autonomous driving modules dedicated to the prediction of human eye fixations, what motivates even more our proposal. From the point of view of the region proposal, Vijay John *et al.* in [46] propose a saliency map based on a GPS sensor, which identifies regions of interest to help a CNN-based traffic signal detector in low illumination conditions. In addition, center bias is a common problem in autonomous driving scenarios. Tao Deng *et al.* propose in [47] and [48] a TD approach which tries to detect where the drivers look, guiding the visual attention process by means of a vanishing point tracker, in order to deal with the aforementioned bias.

It is difficult to build video datasets for autonomous driving applications, at the same time it is expensive to label them with human fixations. Andrea Palazzi *et al.* describe in [49], [50] the creation of DR(eye)VE, a video dataset for autonomous driving composed of 74 video sequences (approx. 6 hours) and fixations from 8 observers. They also propose a CNN architecture based on 3D convolutions to predict the driver's focus of attention using this dataset. Ye Xia *et al.* present in [51] an in-lab dataset called Berkeley DeepDrive Attention (BDD-A), together with a system which includes a convolutional LSTM network and a method to show relevant frames to the model more frequently. Tao Deng *et al.* [52] provide a traffic driving video dataset with fixations, together with a saliency detection model based on compact convolutional-deconvolutional neural networks (CDNN). Either BDD-A [51] or CDNN [52] databases have more diverse fixations than DR(eye)VE [49], but recorded in a laboratory and not under real driving conditions. Moreover, DR(eye)VE [49] provides labels for contextual conditions, which enables us to demonstrate our model's capabilities. For these two reasons, we have decided to use the latter in our experiments. Last but not least, it should be stressed that none of the solutions found in the literature for the prediction of eye fixations in autonomous driving scenarios allow for interpretation, which is the core contribution of our approach based on capsules.

C. CAPSULE NETWORKS

A notable increase in Capsule Networks proposals has been observed in the literature during the last two years. A capsule is a group of neurons, which constitutes a routable item through a dynamic network. Hinton, Sabour and Frosst propose in [17] and [18] the first versions of Capsule Networks. The former routes vectors and the latter routes pose matrices and activations, both through a CNN. Moreover, they both define an iterative process to solve the problem of assigning parts to wholes, assuming that a parse tree like structure is dynamically carved out of a Neural Network.

In 2018, LaLonde and Bagci propose SegCaps [19], a Capsule Network for image segmentation. It consists in an encoder-decoder fully capsular architecture, which improves the computational and memory efficiency by introducing spatial constraints in vector routing, as well as in shared weights. Zhang *et al.* propose in [53] a Capsule Network for multi-instance multi-label learning, which makes use of attention mechanisms while routing. Another interesting approach is shown in [54], where McIntosh *et al.* use text queries in order to try to find actors and actions in video sequences. Both video and text are encoded as capsules, which provide a more effective representation than convolutional layers, and there is a mechanism to fuse them. DeepCaps [55] is proposed in 2019, which uses a 3D convolution-based dynamic routing algorithm. Attention Routing CapsNet [56] is also released in 2019, where dynamic routing and squash activations are replaced by a new routing algorithm and a convolutional activation function, respectively. In addition, Tsai *et al.* recently propose a new routing algorithm for Capsule Networks, which performs at par with ResNet-18 with 4x fewer parameters [57].

We aim to solve visual attention given a context and to provide interpretability for model predictions. For that

purpose, Capsule Networks allow to group information in blocks of capsule types and to build relationships between them. On the basis of the previously cited work, we design our own version of Capsule Network, GLCapsNet, where capsule types constitute visual features and contextual conditions. This enables the modularization of the architecture of the model, which computes condition's predictions and feature-condition relationships at both global and local visual levels.

III. GLOBAL-LOCAL DYNAMICS OF VISUAL ATTENTION

A. MODEL OVERVIEW

In this work, we propose a hierarchical multi-task approach which is able to estimate eye fixations based on several contextual conditions of the scene. Following the assumption made by Fernández-Torres *et al.* in [32], our system begins with the following hypothesis: *Context-driven visual attention in video can be modeled by considering several conditions that define scenarios which, in turn, are represented as combinations of several spatio-temporal features*. For instance, an autonomous driving context can be defined as a cloudy morning across downtown, and each of these conditions can be represented as combinations of features (e.g. color, motion, semantic categories, etc.).

The whole architecture proposed is shown in Figure 1. As can be appreciated, the model is built by means of a latent structure, relating some intermediate blocks (routing Feature Capsules to Condition Capsules) both locally and globally in the space dimension. This is done dynamically for each input stimulus, which allows for interpretation. In subsequent sections, we explain what globally and locally mean in this model. The whole system first encodes the feature representations into Feature Capsules, to be converted into Condition Capsules via the routing algorithm, and finally to be concatenated and decoded into the visual attention map. In addition, it leverages routing coefficients to predict the conditions and to provide interpretability capabilities.

Particularly, with the aim to evaluate our system in autonomous driving scenarios, conditions are based on the ones provided by the dataset considered for the experiments. The dataset used is DR(eye)VE [49], which involves the following groups of conditions:

- Daytime: Morning, Evening, Night.
- Weather: Sunny, Cloudy, Rainy.
- Landscape: Downtown, Countryside, Highway.

The following subsections describe in detail the different stages of the system proposed.

B. VISUAL FEATURE BRANCHES

Here we describe the feature branches that encode the feature level capsules. Three features are extracted from each video frame: color in RGB space; motion estimation, using an optical flow algorithm; and visual entities, using a semantic segmentation algorithm.

First, color channels are normalized by subtracting their mean and dividing by three times their standard deviation (which covers the $\sim 99.7\%$ of the data samples). In addition, values are clipped to the range $[-1, 1]$. Second, the optical flow algorithm used for motion estimation is an efficient implementation[1] of TVL1 [58]. Once motion vectors are computed, the motion feature is made of 3 channels: horizontal and vertical components, together with the motion magnitude (calculated using the Euclidean norm). Horizontal and vertical components are clipped to the range $[-20, 20]$. Furthermore, motion channels are normalized according to the procedure described for color channels above. Finally, the semantic segmentation algorithm used is DeepLabv3+[2] pretrained on Cityscapes dataset [39]. Each output pixel is normalized to sum to 1 along class channels (19 classes for Cityscapes). Dimensions for each feature described are $112 \times 112 \times 3$, $112 \times 112 \times 3$ and $112 \times 112 \times 19$, respectively.

Figure 2 shows the encoder-decoder structure used for training each branch. On the one hand, the encoder block constitutes the branch itself and encodes visual features into latent Feature Capsules. On the other hand, the decoder block decodes information into a visual attention map. Once each branch is trained, the decoder part is removed from the system. Both encoder and decoder consist of convolutional structures with several max pooling or bilinear upsampling layers, respectively, including dropout layers behind these to avoid overfitting. A ReLU activation function is introduced behind each convolutional layer, except at the final layer of the decoder block, which is a 1×1 Conv2D layer and performs a linear combination of the 2D activation maps at its input, being followed by a linear activation. The latter is usually done to solve visual attention tasks; otherwise, the resulting attention map might be distorted. Dashed and dotted borders in some layers indicate that weights are obtained from a pre-trained network during a Transfer Learning phase. These weights are extracted from the first 3 layers of the VGG-CNN-M model [59] trained on the ImageNet dataset [60]. Color and optical flow branches use these 3 layers, while the semantic segmentation branch does not use the first one (dotted line), due to its number of input maps is 19 instead of 3. The structure has also shortcut connections, in an attempt to preserve the spatial resolution of the down-sampled input features. These connections are removed once each branch is transferred to the final system. Finally, we apply a data augmentation technique while training each feature branch. The augmentation consists of cropping and mirroring video frames with a certain probability. For random cropping, frames and GT fixation maps are first resized between reasonable bounds (original and double the dimensions of the images, keeping their aspect ratio). Then, a slack ratio is defined, which determines the crop search space and is ranged from 0 (crop must be tight to the center of the resized

[1] Source code taken from https://github.com/feichtenhofer/gpu_flow

[2] Source code taken from https://github.com/bonlime/keras-deeplab-v3-plus

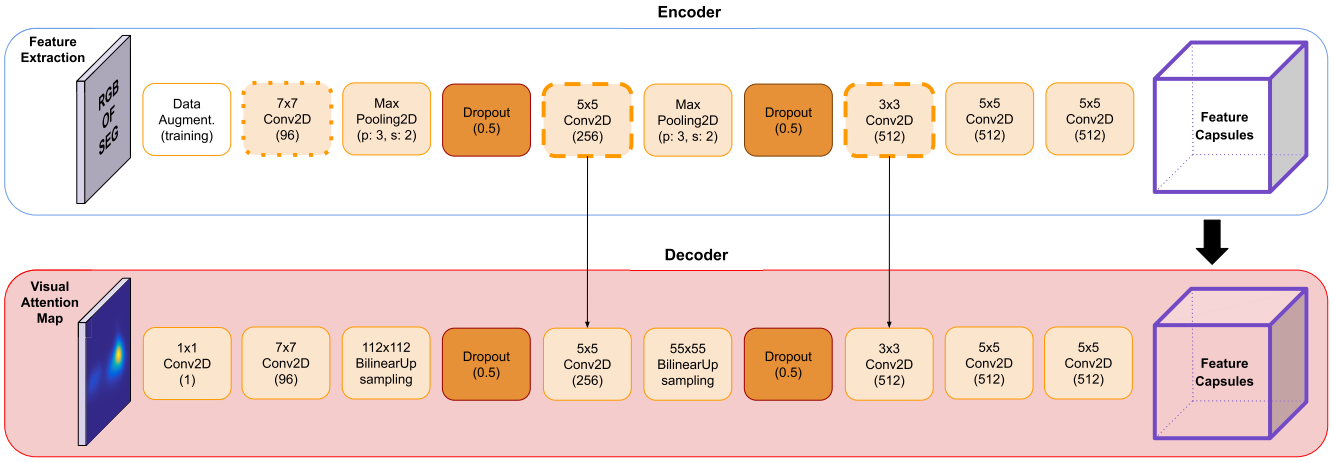


FIGURE 2. Encoder-decoder architecture for branch training. Convolutional layers are defined by their kernel size and number of filters; in max pooling layers, *p* refers to padding and *s* to strides; the rate of units to drop is determined in dropout layers, and the output spatial size is indicated in bilinear upsampling layers. Dashed and dotted layers are initialized using the pre-trained weights from the first 3 layers of the VGG-CNN-M model [59]. In the case of the semantic segmentation branch, pre-trained weights are not used in the first dotted layer. Skip connections are incorporated to improve training, starting at dashed layers and ending at their mirror ones from decoder. Data augmentation stage is defined as the first step, and only applies for training.

image) to 1 (all search space available to locate the crop). The random crop size is equal to 112 × 112 (original dimensions of frames and GT fixation maps).

C. GLOBAL-LOCAL CAPSULE NETWORK

Once visual feature branches are pre-trained, their weights are used in the whole architecture of the visual attention system proposed, which we have called Global-Local Capsule Network (GLCapsNet) and is shown in Figure 1. As can be seen, the last layers of the encoders or branches constitute the Feature Capsules, which are fused using a specific routing mechanism. On the basis of the Feature Capsules, 3 independent tasks are built, one for each group of contextual conditions introduced above (Daytime, Weather and Landscape). This is due to conditions belonging to the same group are mutually exclusive. We define Condition Capsules as the union of each group of conditions. Each group of conditions is defined as the union of capsule types (convolutional volumes), being each capsule type a group of vector capsules. Each of these vector capsules represents a unique condition defined by the capsule type to which they belong. This is because we require routing and presence (routing aggregations as LP and GP in Figure 1) to be both locally and globally defined, so there is a 2D matrix for each pair feature-condition to relate their capsules, a 2D matrix per condition to express its *LocalPresence*, and a single probability value per condition to define the *GlobalPresence*. Additionally, we define representational capsule types as the ones that only represent latent information (in this particular case, Feature Capsules, which represent visual information at the input of the system), and discriminative capsule types as those which allows to predict the presence of a concept based on aggregated routing values from the previous stage (in this particular case, Condition Capsules). Finally, all capsule types are concatenated along the filters dimension and the last stage of the system is a convolutional decoder, which generates a context-driven

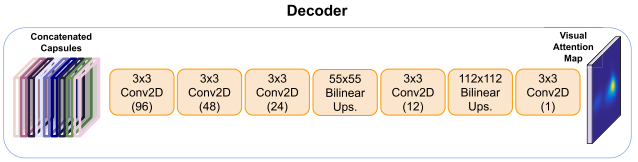


FIGURE 3. Decoder block for the whole architecture, which converts Concatenated Capsules into the visual attention map. Convolutional layers are defined by their kernel size and number of filters, and the output spatial size is indicated in bilinear upsampling layers.

visual attention map for eye fixation prediction based on the conditions predicted for a given scenario. Figure 3 shows in detail the structure of this decoder, which is similar to the one defined above for visual feature branches. Unlike during visual feature branches pre-training, we do not consider a data augmentation procedure in the whole architecture. Moreover, pre-trained weights are not frozen but updated during the whole system training stage.

D. LOCAL ROUTING DYNAMICS

Here are detailed the GLCapsNet routing equations. Formulation is similar as in [19], despite both approaches have some critical differences. Leaving the autonomous driving application aside, the main difference is the purpose of the capsule modules. The authors of [19] try to encode better representations of the convolutional information and to decode it into capsules, whose modulus are able to perform the main task (object segmentation), including regularization with masking (similar to [17]) to improve internal representations. In contrast, our goal is to provide an internal concept-level self-organized structure for each input stimulus or video frame, incorporating intermediate endpoints which improve internal representations, at the same time they predict contextual conditions enabling interpretability. If we go into the formulation details, both approaches impose local spatial constraints to the original dynamic routing [17], performing an operation

Algorithm 1 Dynamic Routing Between Capsules in GLCapsNet

procedure Routing($\hat{u}_{xy|t_i^l t_j^{l+1}}, d, l, k_h, k_w$)

  for all capsule types t_i^l within a $k_h \times k_w$ kernel centered at position (x, y) (layer l) and capsule xy in capsule type t_j^{l+1} (layer $l+1$): $b_{t_i^l t_j^{l+1}|xy} \leftarrow 0$

  for d iterations **do**

    for all capsule types t_i^l (layer l): $r_{t_i^l t_j^{l+1}|xy} \leftarrow \texttt{softmax}(b_{t_i^l t_j^{l+1}|xy})$

    for all capsules xy in t_j^{l+1} (layer $l+1$): $p_{xy|t_j^{l+1}} \leftarrow \sum_i r_{t_i^l t_j^{l+1}|xy} \hat{u}_{xy|t_i^l t_j^{l+1}}$

    for all capsules xy in t_j^{l+1} (layer $l+1$): $v_{xy|t_j^{l+1}} \leftarrow \texttt{squash}(p_{xy|t_j^{l+1}})$

    for all capsule types t_i^l (layer l) and capsules xy in t_j^{l+1} (layer $l+1$): $b_{t_i^l t_j^{l+1}|xy} \leftarrow b_{t_i^l t_j^{l+1}|xy} + \texttt{cosine}(\hat{u}_{xy|t_i^l t_j^{l+1}}, v_{xy|t_j^{l+1}})$

  return $v_{xy|t_j^{l+1}}$

that behaves as a convolution (parameter sharing across spatial locations within a capsule type), while keeping routing operation across all vector capsules. In [19], this convolution operator is shared by all capsule types in the previous layer, but there is a different one for each capsule type in the next layer. Here we remove this sharing property for the previous layer and build a different convolutional kernel for each pair of capsule types between consecutive layers, while keeping parameter sharing across spatial locations. This is necessary due to GLCapsNet is required first to merge three feature branches with distinct behaviours, and then to build groups of capsules that represents quite different contextual conditions. This is not the same as in the encoder-decoder fully capsular system in [19], where they merge groups of capsules from the same layer and without conceptual interpretation, therefore with similar properties.

First, let us introduce a child capsule layer l, whose output is a set of n capsule types $T^l = \{t_1^l, t_2^l, \ldots, t_n^l\}$. Each capsule type $t_i^l \in T^l$ consists of a $h^l \times w^l$ grid of z^l-dimensional child capsules $C = \{c_{11}, \ldots, c_{1w^l}, \ldots c_{h^l 1}, \ldots, c_{h^l w^l}\}$, being $h^l \times w^l$ the spatial dimensions of the output of layer $l-1$. Parent capsule layer $l+1$ outputs a set of m capsule types $T^{l+1} = \{t_1^{l+1}, t_2^{l+1}, \ldots, t_m^{l+1}\}$. Each capsule type $t_j^{l+1} \in T^{l+1}$ consists of a $h^{l+1} \times w^{l+1}$ grid of z^{l+1}-dimensional parent capsules $P = \{p_{11}, \ldots, p_{1w^{l+1}}, \ldots p_{h^{l+1}1}, \ldots, p_{h^{l+1}w^{l+1}}\}$, being $h^{l+1} \times w^{l+1}$ the spatial dimensions of the output of layer l.

Given a capsule type $t_j^{l+1} \in T^{l+1}$, its associated parent capsules $p_{xy} \in P$ receive a set of *prediction vectors* $\{\hat{u}_{xy|t_1^l t_j^{l+1}}, \hat{u}_{xy|t_2^l t_j^{l+1}}, \ldots, \hat{u}_{xy|t_n^l t_j^{l+1}}\}$ from each spatial location (x, y) in each child capsule type $t_i^l \in T^l$. In order to compute this set of *prediction vectors*, a transformation matrix $M_{t_i^l t_j^{l+1}}$ of shape $k_h \times k_w \times z^l \times z^{l+1}$ is learned for each pair t_i^l and t_j^{l+1}, being $k_h \times k_w$ the shape of a user-defined kernel. Thus, we have $|T^l| \times |T^{l+1}|$ transformation matrices, where $|T^l|$ and $|T^{l+1}|$ are the number of capsule types in layer l and $l+1$, respectively. Our visual attention system for autonomous driving scenarios considers three Feature Capsules to perform three prediction tasks. Each task involves three contextual conditions. Therefore, assuming $k_h = k_w = 3$ and being $z^l = 512$ and $z^{l+1} = 64$, we define $3 \times 3 = 9$ transformation matrices of shape $3 \times 3 \times 512 \times 64$ for each task. Each matrix is applied to a sub-grid of child capsules outputs $U_{xy|t_i^l}$ of shape $k_h \times k_w \times z^l$, which is centered at location (x, y) in layer l. Then, the whole set of *prediction vectors* is computed as $\hat{u}_{xy|t_i^l t_j^{l+1}} = M_{t_i^l t_j^{l+1}} \times U_{xy|t_i^l}$, $\forall t_i^l \in T^l$ and $\forall t_j^{l+1} \in T^{l+1}$. Each $M_{t_i^l t_j^{l+1}}$ does not depend on the spatial location (x, y), as is shared across all spatial locations given a pair of capsule types t_i^l and t_j^{l+1}. Re-defining parent capsules as $p_{xy|t_j^{l+1}}$ to distinguish between parent capsule types, final input to each parent capsule is computed as the weighted sum of the *prediction vectors* as follows:

$$p_{xy|t_j^{l+1}} = \sum_i r_{t_i^l t_j^{l+1}|xy} \hat{u}_{xy|t_i^l t_j^{l+1}} \tag{1}$$

where $r_{t_i^l t_j^{l+1}|xy}$ are the routing coefficients determined by the dynamic routing procedure summarized in Algorithm 1. In total, $|T^l| \times h^{l+1} \times w^{l+1} \times |T^{l+1}|$ routing coefficients are computed, which allow to express relationships between each pair of child and parent capsule types t_i^l and t_j^{l+1} both locally $(h^{l+1} \times w^{l+1})$ and globally by applying the following equation (only for the sake of interpretation):

$$R_{t_i^l t_j^{l+1}} = \frac{1}{h^{l+1} w^{l+1}} \sum_{xy} r_{t_i^l t_j^{l+1}|xy} \tag{2}$$

Local routing coefficients are computed from the log prior probabilities $b_{t_i^l t_j^{l+1}|xy}$ that prediction vector $\hat{u}_{xy|t_i^l t_j^{l+1}}$ should be routed to parent capsule $p_{xy|t_j^{l+1}}$, then applying a softmax function along the m parent capsule types:

$$r_{t_i^l t_j^{l+1}|xy} = \frac{\exp\left(b_{t_i^l t_j^{l+1}|xy}\right)}{\sum_j \exp\left(b_{t_i^l t_j^{l+1}|xy}\right)} \tag{3}$$

This means that given a fixed spatial location (x, y), a child capsule is weighted before being sent to each parent capsule, with sum equal to 1 along parents. First, as in [19], the creation of *prediction vectors* is locally constrained and,

second, routed child capsules are only those within the user-defined kernel, due to *prediction vectors* creation constraints. Activation applied to each vector capsule during routing is the non-linear squashing function proposed in [17]:

$$v_{xy|t_j^{l+1}} = \frac{\left\| p_{xy|t_j^{l+1}} \right\|^2}{1 + \left\| p_{xy|t_j^{l+1}} \right\|^2} \frac{p_{xy|t_j^{l+1}}}{\left\| p_{xy|t_j^{l+1}} \right\|} \tag{4}$$

This activation is also applied to the output of both representational and discriminative capsule types. An agreement function is used to update log prior probabilities while routing, and here we propose to use the cosine similarity $a_{xy|t_i^l t_j^{l+1}} = cosine(v_{xy|t_j^{l+1}}, \hat{u}_{xy|t_i^l t_j^{l+1}}) = \frac{v_{xy|t_j^{l+1}} \cdot \hat{u}_{xy|t_i^l t_j^{l+1}}}{\left\| v_{xy|t_j^{l+1}} \right\| \cdot \left\| \hat{u}_{xy|t_i^l t_j^{l+1}} \right\|}$ as opposed to the scalar product defined in previous works [17], [19]. We assume $d = 3$ routing iterations to update log prior probabilities.

E. MULTI-TASK LEARNING

Global-Local Capsule Network proposed constitutes a multi-task learning framework, where four different tasks are performed: three contextual conditions prediction tasks and a visual attention estimation task. Discriminative capsule types are required to predict contextual conditions. During the training phase, objective labels for these first classification tasks are encoded in one-hot. The function that provides this functionality is called *Presence* and has 2 main requirements: 1) To express both local and global presence of the conditions at the scenario, which allows for visual attention interpretation, and 2) To improve the final visual attention estimation. For that purpose, *Presence* is a probability distribution computed over local routing coefficients $r_{t_i^l t_j^{l+1}|xy}$, which range from 0 to 1 (see softmax function in (3)), for each of the three conditions prediction tasks separately. These coefficients are first preprocessed by the *Emphasize* function. This function emphasizes the difference between local routing coefficients:

$$e_{t_i^l t_j^{l+1}|xy} = \texttt{sigmoid}\left(\beta' \cdot \left(r^{\alpha}_{t_i^l t_j^{l+1}|xy} - 0.5\right)\right) \tag{5}$$

$$\alpha = \frac{\ln\left(\ln\left(\frac{\frac{1}{|T^{l+1}|}}{1-\frac{1}{|T^{l+1}|}}\right) \cdot \frac{1}{\beta'} + 0.5\right)}{\ln\left(\frac{1}{|T^{l+1}|}\right)} \tag{6}$$

$$\beta' = 10 \cdot (\beta + 0.5), \quad where \;\; 0 \leq \beta \leq 1 \tag{7}$$

Emphasize is designed as a configurable sigmoid function which is also in the range [0, 1], and it is applied only during the training stage. Parameter α controls the node point, which is the intersection between this function and a linear function between 0 and 1. Routing values below this point are decreased while routing values above it are increased. β' is the strength of the sigmoid and is linearly projected to a space where β values from 0 (weak) to 1 (strong) offer reasonable emphasis without distortion. Variable α is thus a constant defined by β' and $|T^{l+1}|$, which depends on the system architecture. Therefore, β is a hyper-parameter which requires validation during the experiments. This formulation enforces the node point to be $\frac{1}{|T^{l+1}|}$, which is the equiprobability point for all parent capsule types' *Presence*.

Once routing coefficients are emphasized, *Presence* function is applied, first locally as *LocalPresence* (LP), by computing the average of the routing coefficients at each spatial location over the number of input capsules (feature capsules), and then globally as *GlobalPresence* (GP), by computing the average of LP over all spatial locations. Only the global endpoint is controlled by a loss function, since condition labels are provided at frame level. The formulation is as follows:

$$LP_{t_j^{l+1}|xy} = \frac{1}{|T^l|} \sum_i e_{t_i^l t_j^{l+1}|xy} \tag{8}$$

$$GP_{t_j^{l+1}} = \frac{1}{h^{l+1} w^{l+1}} \sum_{xy} LP_{t_j^{l+1}|xy} \tag{9}$$

The global function enforces routing coefficients to make the best effort at driving child capsules information to the direction of the discriminative capsule type associated to the condition which is present at the scene. In addition, during the training phase, capsule types associated to conditions which are not present in the scene, according to GT labels, are masked with 0's, which constitutes a learning guide. Finally, below is detailed the multi-task loss function, which is comprised of a Kullback-Leibler Divergence (KL) for the eye fixation estimation task and the Spread Loss (SL) defined at [17] for contextual conditions prediction:

$$KL(\hat{g}_t, g_t) = \sum_{xy} g_{t_{xy}} \log\left(\epsilon + \frac{g_{t_{xy}}}{\epsilon + \hat{g}_{t_{xy}}}\right) \tag{10}$$

$$SL_{t_j^{l+1}} = \max(0, m - (GP_{t_o^{l+1}} - GP_{t_j^{l+1}}))^2 \tag{11}$$

$$SL = \sum_{j \neq o} SL_{t_j^{l+1}} \tag{12}$$

$$Loss = KL(\hat{g}_t, g_t) + \gamma \cdot \sum_n SL(GP^n, O^n, m) \tag{13}$$

where g_t and $\hat{g}_t$ are the GT fixation map and the estimated visual attention map, respectively; ϵ is a regularization parameter of the KL function; $GP_{t_o^{l+1}}$ is the GP value at the one-hot activated position; GP^n and O^n are the GP values and the one-hot encoded labels for the n discriminative task of the model, respectively; and finally m (margin) and γ are design parameters, which are empirically determined.

IV. EXPERIMENTAL RESULTS

A. EXPERIMENTAL DESIGN

1) DATASET

Our main goal is to demonstrate that GLCapsNet is able to model and interpret visual attention in autonomous driving scenarios, attending to several contextual conditions. For that purpose, results are given for the DR(eye)VE dataset [49], which contains car driving video sequences. GT maps are

based on observer's eye fixations recorded with wearable eye tracking glasses under the different real driving conditions defined in Section III. Training, validation and test sets are defined as in the original paper: videos from 1 to 37 for training, videos from 38 to 47 for test, and frames from the training set which range from 3501 to 4000 are left for validation.

2) EXPERIMENTAL SETUP AND BASELINES

The experimental setup is as follows. First, we train the feature branches using data augmentation, pre-trained weights from the VGG-CNN-M model [59], a decoder guide and mirror shortcuts. Then, branches' decoders are removed and remaining encoders are merged by GLCapsNet using Transfer Learning. More details are described in Section III-B. We define some baselines to compare with in an ablation study, as well as the proposed approach:

- Feature-based models: We consider one model per feature branch (RGB, OF, Segmentation) by using the architecture defined in Figure 2, but without removing the decoder block once it is trained. Subsequent baselines defined below, which are built on top of these encoding branches, do not use data augmentation, pre-trained weights from VGG-CNN-M [59] again, branches' decoders or shortcut connections.
- Fusion-based models: They fuse the 3 feature-based models by considering 2 direct approaches:
  - Simple Fusion (SF): We compute the average of the visual attention maps predicted at the output of the decoder block of the 3 feature-based models described above. This constitutes the most straightforward fusion.
  - Generic Fusion (GF): Given the GLCapsNet proposed, we replace the capsule blocks, dynamic routing algorithm, *Presence* function and the final concatenation by a classical convolutional pipeline with an analogue configuration. That is, a fully convolutional network with $N \times |T^{l+1}| \times z^{l+1} = 3 \times 3 \times 64 = 576$ filters, being $N = 3$ the number of auxiliary tasks as it is stated in Section III-E, $|T^{l+1}| = 3$ the number of contextual conditions per conditions group considered in GLCapsNet and $z^{l+1} = 64$ the number of filters per condition capsule type, as it is stated in Section III-D.
- Capsule-based models: They fuse the output at the encoder block of the 3 feature-based models via capsule layers, with the aim of evaluating the contribution of each of the components of our proposal, drawing from the SegCaps modules presented in [19]:
  - SegCaps module (SC): It merges branches using the capsule module proposed in [19], defining 9 output capsule types, one per condition in GLCapsNet. Weights are shared between all the input capsule types, as in [19].
  - Non Shared (NS-SC): SC but stopping sharing weights between input capsule types.
  - 3 Pathway (3-NS-SC): It divides the NS-SC into 3 parallel branches, one per group of conditions considered in GLCapsNet (Daytime, Weather, Landscape), but without explicit knowledge about conditions yet.
  - Mask Conditions (Mask-3-NS-SC): It applies the masking procedure described in Section III-E to 3-NS-SC. This baseline is the first one specialized by conditions.
  - Multi-Task (MT-Mask-3-NS-SC): It defines the multi-task framework by adding *Presence* function to Mask-3-NS-SC, but without applying *Emphasize* function (5) yet. This baseline is even more specialized by conditions.
  - GLCapsNet: It applies the *Emphasize* function to the MT-Mask-3-NS-SC baseline, obtaining the final proposed system.

Moreover, we report the results obtained by the three related approaches in the *state-of-the-art* introduced in Section II-B: BDD-A [51], CDNN [52] and DR(eye)VE [49]. In order to provide a fair comparison, all methods have been re-trained by using the DR(eye)VE [49] database.

3) EVALUATION METRICS

On the one hand, accuracy is used as the evaluation metric for conditions prediction. On the other hand, due to lack of consensus in how to evaluate visual attention, many metrics are proposed in the literature, each with different properties. Bylinskii *et al.* [61] expose an exhaustive analysis about them, being used in this work Kullback-Leibler Divergence (KL), Information Gain (IG), Pearson's Correlation Coefficient (CC) and shuffled Area Under Curve (sAUC). Moreover, we take from Leboran *et al.* [62] the shuffled variant of Normalized Scanpath Saliency (sNSS). The lower KL the better the results, unlike with the rest of the metrics used. KL and CC are distribution based metrics, while IG, sAUC and sNSS are location based. CC and sAUC are bounded in the range $[0, 1]$ and KL is only lower bounded by 0. IG and sNSS are not bounded. For IG, a value over 0 tells that prediction is better than baseline. For sNSS, a value less, equal or greater than 0 indicates anti-correspondence, chance or correspondence between maps above chance, respectively. IG, sNSS and sAUC require the computation of a shuffle-map (commonly called baseline or bias for IG), which is obtained as the average of fixations from the training set (Monte Carlo estimation of bias). This shuffle-map guides IG, sNSS and sAUC to provide a higher penalty to biased predictions. Particularly, sNSS and sAUC use it as a distribution probability map to gather negative samples for their computation.

4) TRAINING AND IMPLEMENTATION DETAILS

Code is developed in Python using TensorFlow and Keras, and it is publicly available on GitHub.[3] Hardware is composed of a CPU (8 cores and 16 GB of RAM) and a NVIDIA

[3]GLCapsNet code is publicly available on https://github.com/javiermcebrian/glcapsnet

TABLE 1. Ablation study for the GLCapsNet model proposed using the DR(eye)VE dataset [49].

Model	KL ↓	IG ↑	CC ↑	sAUC ↑	sNSS ↑	Daytime acc. ↑	Weather acc. ↑	Landscape acc. ↑
RGB	1.67	0.14	0.53	0.68	0.31	-	-	-
OF	1.65	0.19	0.53	0.65	0.25	-	-	-
Segmentation	1.64	0.16	0.53	0.66	0.27	-	-	-
SF	1.61	0.25	0.54	0.66	0.27	-	-	-
GF	1.63	0.31	0.54	0.68	0.32	-	-	-
SC	1.63	0.05	0.54	0.59	0.16	-	-	-
NS-SC	1.61	0.30	0.55	0.68	0.32	-	-	-
3-NS-SC	1.62	0.34	0.55	0.69	0.33	-	-	-
Mask-3-NS-SC	1.60	**0.41**	0.56	0.69	0.33	-	-	-
MT-Mask-3-NS-SC	1.65	0.38	0.55	0.68	0.32	0.83	0.67	0.80
GLCapsNet	**1.60**	0.40	**0.56**	**0.69**	**0.34**	**0.83**	**0.70**	**0.83**

TABLE 2. Results obtained in the DR(eye)VE dataset [49] for the GLCapsNet model proposed and related methods in the *state-of-the-art*.

Model	KL ↓	IG ↑	CC ↑	sAUC ↑	sNSS ↑	Daytime acc. ↑	Weather acc. ↑	Landscape acc. ↑
BDD-A [51]	1.78	0.18	0.53	0.68	0.31	-	-	-
CDNN [52]	1.68	0.10	0.55	0.69	0.33	-	-	-
DR(eye)VE [49]	1.64	0.31	0.55	0.69	0.34	-	-	-
GLCapsNet	**1.60**	**0.40**	**0.56**	**0.69**	**0.34**	**0.83**	**0.70**	**0.83**

TABLE 3. Inference computational time and frame rate obtained in the DR(eye)VE dataset [49] for the GLCapsNet model proposed and related methods in the *state-of-the-art*. Hardware used is a NVIDIA GTX 1070 GPU. Models and input frames are assumed to be already loaded in memory. BDD-A [51] has a feature extraction (f.e.) stage, while DR(eye)VE [49] and our approach GLCapsNet has to compute optical flow (of) and semantic segmentation (seg) maps first.

Model	Inf. time (s/frame)	Frame rate (fps)
BDD-A [51]	0.0110 (f.e.) + 0.0012 (model) = 0.0122	82
CDNN [52]	0.0039	258
DR(eye)VE [49]	0.0071 (of) + 0.0845 (seg) + 1.0887 (model) = 1.1803	<1
GLCapsNet	0.0071 (of) + 0.0845 (seg) + 0.0329 (model) = 0.1245	8

GTX 1070 GPU. All networks are trained using Adam optimizer with a learning rate of 10^{-4} and an exponential decay of 0.99. Batch size is 32 for feature-based models and SF model, and 8 for GF and capsule-based models. Models are trained during 50 epochs, but saving checkpoints only when validation loss is improved, and the number of batches per epoch is 512. Multi-task parameters are empirically determined according to the validation set defined in DR(eye)VE [49]: β for *Emphasize* function is set to 1, namely the highest strength, as described in Section III-E; margin m is set to 0.9 as in [17], but without scheduling; and γ is set to 0.1. The sensitivity of the latter, which balances the visual attention estimation and contextual conditions prediction losses, is higher than for the first two parameters, so we exhaustively try values in the range [0.1, 1]. This configuration takes approximately 3 hours to train.

B. ABLATION STUDY

First of all, it is required to justify some design decisions of the proposed system. For that purpose, we conducted the following ablation study by considering the baselines described in Section IV-A2. Results are summarized in Table 1. As can be appreciated, individual feature-based models allow for a good prediction of eye fixations, even their CC, sAUC and sNSS scores are not so far from the best performing models, especially in the case of RGB. However, their performance is significantly lower when we look at KL and IG, which means that probably CC, sAUC and sNSS are more saturated metrics, being KL and IG more expressive for our analysis. Fusion-based models perform better than their feature building blocks separately, except for SF when being compared according to sNSS, which may indicate that a too simple fusion strategy, such as an average over the feature-based maps, could deteriorate the individual predictions. This fact positions GF as the best straightforward and convolutional baseline, which could compete with capsule-based approaches.

SC baseline is equivalent to apply the proposal in [19], achieving the worst results out of the capsule-based models because the weights are shared between feature capsules, which limits the power of our approach when merging feature branches of different nature. This approach is even worse than feature-based and fusion-based baselines for most of the metrics. NS-SC is the first capsule-based baseline which is comparable to GF, as it can merge feature patterns properly in the capsule framework. Results improve even more when splitting the capsule module into 3 pathways (3-NS-SC), due to the addition of structure and hierarchy to the model. The critical point comes at Mask-3-NS-SC, which achieves almost the best results in the study thanks to the implicit specialization of the condition capsules, which are masked with 0's for the first time during the training stage when they are

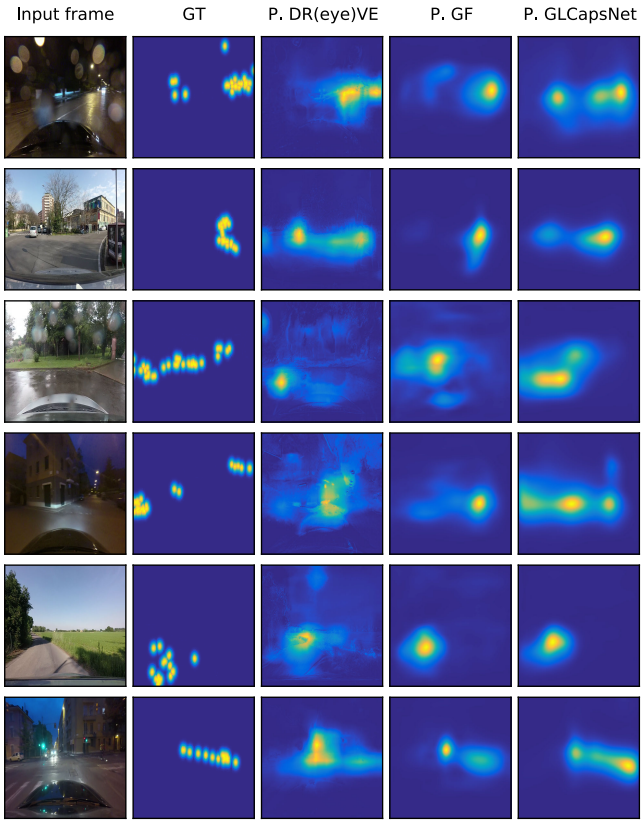


FIGURE 4. Visual attention maps obtained for some example frames in DR(eye)VE dataset [49] by three different approaches: DR(eye)VE [49] (the best competitor), GF (the best fusion-based baseline) and our proposed GLCapsNet.

not present in the scene. The insertion of the *Presence* function, which establishes the multi-task framework (MT-Mask-3-NS-SC), significantly drops the performance according to KL and IG scores, but offers a new output, the prediction of the conditions, which allows for the interpretation of the system. This drop in performance is solved by making use of the *Emphasize* function (GLCapsNet, our final proposal), which even improves the conditions prediction.

We can conclude that the most important insight of this analysis is that the specialization by conditions is improving performance. This is more notable in the usage of masking, but not negligible in the usage of multi-task via *Presence-Emphasize*, as *Emphasize* clearly improves the conditions prediction and provides stability to the visual attention metrics. Finally, this specialization is even more important due to the interpretability capabilities of the model, which are explained in detail in Sections IV-D and IV-E and would not be allowed by a separate simple model to predict conditions.

C. RESULTS ON THE DR(EYE)VE DATASET

1) QUANTITATIVE RESULTS

Table 2 summarizes the results obtained in the DR(eye)VE dataset [49] for the proposed system and other existing solutions in the *state-of-the-art*. In general, as can be appreciated, contextual conditions prediction task is easier to learn than the eye fixation prediction task. Nonetheless, in terms of visual attention estimation, GLCapsNet matches or outperforms the results obtained by related approaches in the literature, offering a huge improvement for the IG metric. It can be seen that DR(eye)VE [49] model is the best competitor compared to GLCapsNet, but there is still a significant margin in terms of KL and IG metrics. In addition, GLCapsNet predicts the contextual conditions and provides interpretability of the results. Regarding contextual conditions, Daytime and Landscape predictions are quite accurate but Weather prediction achieves lower results. Our experiments demonstrate the noteworthy ability of convolutional capsules to effectively learn hierarchical representations of visual attention. This opens the door to their extension using temporal modules, such as Conv3D or ConvLSTM, which were considered for the design of DR(eye)VE [49] and BDD-A [51] architectures, respectively. Indeed, although our system already receives as input a motion feature based on optical flow, temporal modules could be useful to model drivers' dynamic behavior during the training stage, taking into account the spatio-temporal information gathered by eye fixation sequences.

2) QUALITATIVE RESULTS

Figure 4 shows the visual attention maps obtained by DR(eye)VE [49] (the best approach in the literature) and GF (the best fusion-based baseline), together with our proposed GLCapsNet (the best capsule-based model) for some example frames in the DR(eye)VE dataset [49]. The frames correspond to different contexts, with the aim to analyze these models in a wide range of situations. In the first frame, we can see that the observer is going to steer to the right while there is an inbound traffic lane, which is well modeled by GLCapsNet. In the second frame, only the proposed system is able to accurately focus the attention to the right while keeping slight and smooth attention cues from the previous situation (the main road where the driver comes from). In the third example, the observer is going to turn to the left, but in the right side there is something unusual that is part of the attentional focus, as the proposed system is capturing better than the other systems. In the fourth example, GLCapsNet is the best model that tries to attend to the left steering while expressing slight intentions for the right side, from where unexpected vehicles could come, but DR(eye)VE [49] model is blurring the predicted attention map. In the fifth row, all the models perform in a similar way, but our approaches are more accurate than DR(eye)VE [49]. Finally, in the last row, the proposed system follows the GT pattern much better than the baselines. In addition, DR(eye)VE predictions seem to be too textured in general, allowing us to see the frame's details. Therefore, we can conclude that GLCapsNet provides the best qualitative results.

3) INFERENCE COMPUTATIONAL TIME AND FRAME RATE

Table 3 shows the inference computational time and frame rate for the proposed approach and related methods in the *state-of-the-art*, given a system with a NVIDIA GTX

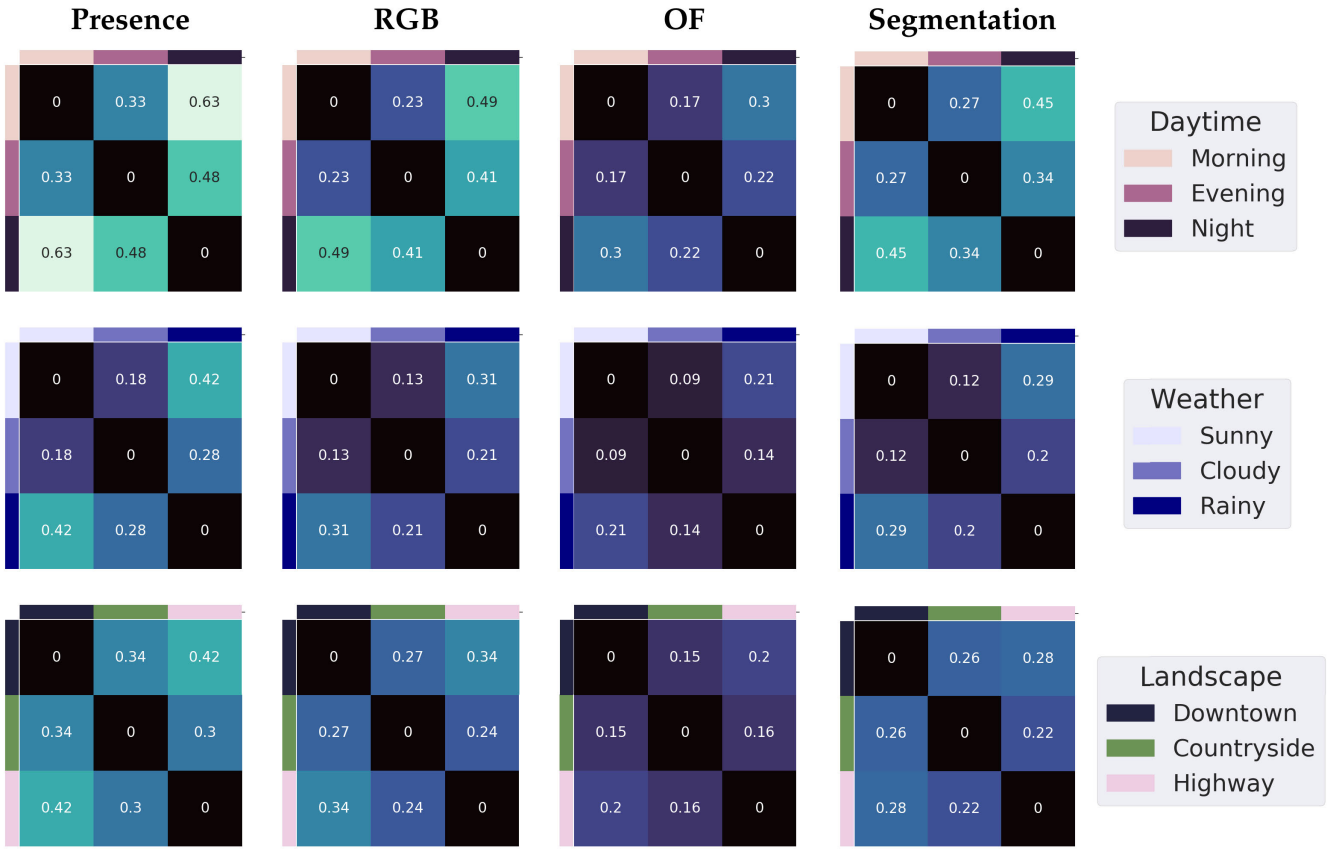


FIGURE 5. Global dynamics shown as dendrograms for each group of conditions in DR(eye)VE dataset [49]. Dynamics are explained as *Presence* does at the first column. Then, they are decomposed into routing coefficients, providing the features point of view: RGB, OF and Segmentation.

1070 GPU. Models and input frames are assumed to be already loaded in memory. We take into account the computational time of additional stages prior to some models: BDD-A [51] has a feature extraction (f.e.) stage, while DR(eye)VE [49] and our approach GLCapsNet have to compute optical flow (of) and semantic segmentation (seg) maps first, using the same algorithms. While BDD-A [51] and CDNN [52] achieve the lowest average time per frame, their performance in terms of eye fixation prediction is significantly lower than ours, as was discussed in Section IV-C1. CDNN is faster than BDD-A, as it only makes use of a single encoder-decoder architecture without temporal dependencies (BDD-A uses ConvLSTM modules). In contrast, DR(eye)VE and our GLCapsNet are combining different feature extractors, which improves their quality but drops their efficiency in comparison with the other methods in the table. When contrasting our approach to the best competitor in quantitative terms, DR(eye)VE, we can observe that our approach is more efficient, even offering a reasonable frame rate of 8 fps while running in a not extremely powerful hardware, compared to the one that could be placed in an autonomous vehicle to deal with computer vision tasks.

D. INTERPRETABILITY: GLOBAL DYNAMICS

For each video frame, GLCapsNet is able to route features' information structures to conditions' ones by means of contextual dynamics, building a hierarchical structure with the goal to represent visual attention. In this section, dynamics are summarized to analyze their general behavior, so we make use of global information about routing (2) and *GlobalPresence* (9). These mechanisms provide interesting interpretability capabilities to the system proposed, which allow to determine what is the contribution of each visual feature to each contextual condition, which conditions are difficult to model, what their most common mistakes are, if we should extend the dataset for these difficult cases or re-label them properly, etc.

Figure 5 shows 12 dendrograms which serve to represent global dynamics. Each row of dendrograms corresponds to each group of conditions (Daytime, Weather, Landscape), being each condition defined by a color legend for the sake of interpretation. These colors are placed at dendrogram's left and up sides. Each cell represents the divergence of each pair of conditions by a numeric value and a color, where the numeric value is normalized between 0 and 1 and the color is normalized along all dendrograms' numeric values. In this way, each dendrogram shows how different each group of mutually exclusive conditions are. This difference is explained from multiple points of view (see columns): *Presence* and features. As *Presence* is based on routing values (see (5), (8) and (9)), we can see *Presence* dendrograms as the resulting combined behavior from features' dendrograms.

Each dendrogram is computed as follows. First, routing coefficients and *Presence* are generated for each frame in the test set. Then, the values that apply for mutually exclusive groups of conditions and each particular feature are collected. For example, for the top-left dendrogram we select Daytime-*Presence* only, while for the one on its right we select the routing coefficients for the relationship Daytime-RGB. Once we have collected the data, as we do not want the distribution of the values themselves, but the distribution of when they are correctly predicting the conditions, we group the values by considering the real condition labels and, after that, we average the results to estimate the probability distributions. Finally, numeric values are computed using Jensen-Shannon Divergence (JSD) between each pair of conditions:

$$JSD(P, Q) = \frac{1}{2}KL(P, M) + \frac{1}{2}KL(Q, M) \tag{14}$$

where P and Q are the data defined above as probability distributions for each case, $M = \frac{1}{2}(P + Q)$ and KL is Kullback-Leibler Divergence defined in (10), but assuming 1D probability distributions. JSD values behave as a soft metric (before hard predictions), based on *Presence* and routing values, which can serve for interpretability measurement.

As can be seen in Figure 5, OF is less discriminative than the other features, being RGB the best in these terms. This means that RGB is able to route information better than the other features, guiding the main flow to where is needed (i.e. the condition defined for each frame). This guiding property is due to the specialization of the model by conditions, and is desirable since information flow gets structured and inefficiencies are reduced. As an example, the RGB capsule type of the system is able to guide information efficiently with a 0.49 discriminative power between Morning and Night. For a better understanding, if all dendrogram values were 0, we could say that each feature's capsule type sends information to each conditions' capsule types group as the Generic Fusion would do. We could see these values as a structural efficiency gain measure over the Generic Fusion baseline. *Presence* column, as an aggregation of features behavior, represents the global structural efficiency gain. If we compute the mean of that measure for each group of conditions, namely by rows of Figure 5, the results explain why Daytime and Landscape accuracies are better than Weather's, as it is shown in Table 2.

Changing the point of view to the rows in Figure 5, we could analyze specific cases of confusion. When divergence is low, this means that the pair of conditions share visual properties (similar colors, driving speed and semantic concepts) and, when divergence is high, it means that they have different ones. In other words, the presence of these concepts at the scene is similar or different, respectively, conceptually speaking. It can be observed for Daytime row that the pair Morning-Evening has the lowest divergence, while the pair Morning-Night has the highest one. In the Weather row, the pair Sunny-Cloudy has the lowest values, while Sunny-Rainy achieves the highest one. Finally, in the Landscape row, we can see that the pair Countryside-Highway has the lowest divergence, while Downtown-Highway has the highest one, except for OF, which obtains its lowest value for Downtown-Countryside. These insights make sense and could be explained from a human perspective in the scope of visual attention for autonomous driving. However, some of them could express the need of more video sequences with other variations of the most degraded conditions, a recalibration of the condition labels, or that human language (which defines conditions labels) is limited as it defines frontiers on the information which are useful for communications but not for this task.

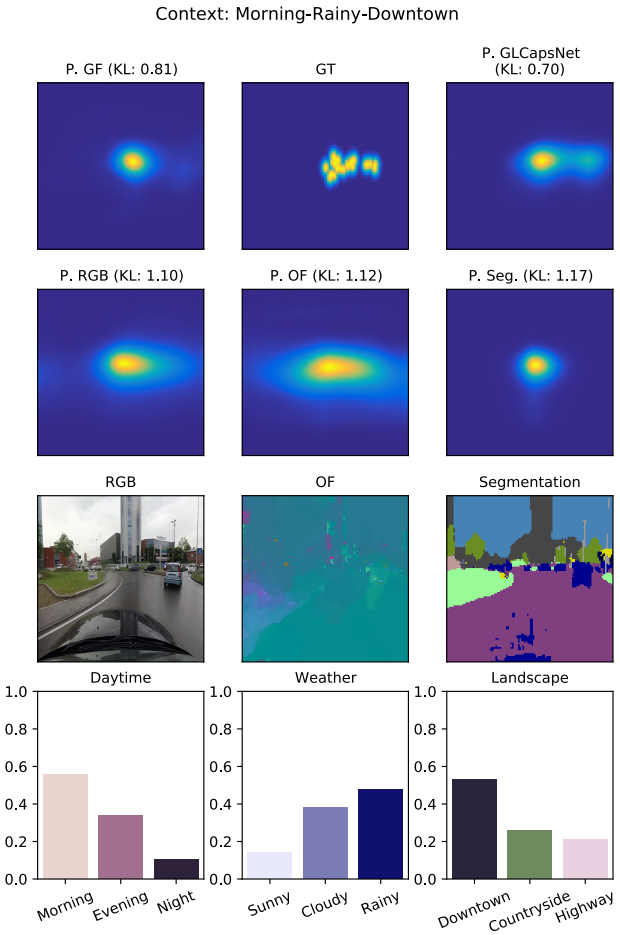


FIGURE 6. Results obtained for frame 4153 in video 47 from DR(eye)VE dataset [49], where a decision at a crossroad is made. Video 47 context is defined by the conditions Morning, Rainy and Downtown. In the first row, GT fixations and visual attention maps are shown for the most relevant baselines taken from fusion-based (GF) and capsule-based (GLCapsNet model proposed). Baselines are introduced in section IV-A2. In the second row, visual attention maps for the feature-based models are shown, to be compared with the features themselves in row 3. Finally, on the last row, *GlobalPresence* values (conditions prediction scores) determined by GLCapsNet for each condition $GP_{t_j^{l+1}}$ (9) are included.

E. INTERPRETABILITY: LOCAL DYNAMICS

In this section we expose how contextual dynamics work, as they are defined in Section IV-D, but going from global to

local perspective. Here, routing coefficients and *Presence* are expressed in a 2D form, giving us the opportunity to explore how the system works at spatial locations. Our purpose is to illustrate local dynamics for a better understanding, trying to answer the following question: *How are the visual features at the spatial locations highlighted by the system for a given contextual condition?* This can be done by assuming that routing coefficients constitute visual attention maps particularized for each pair feature-condition.

Figures 6, 7 and 8 show an example frame taken from video 47 in DR(eye)VE [49] database, where a decision at a crossroad is made, together with the information provided by the GLCapsNet model proposed. Additional example figures (Figures SM.1-SM.18) and video sequences are included in the supplementary material. In the case of Figure 6, the driver is focusing the attention on the road but with an additional interest in the car that is nearby. This final attention is the result of combining cues from visual features (the car as a color pattern, moving object and a semantic concept) guided by 2D context representations (which information is relevant given that it is a rainy morning in the downtown). First, as can be appreciated in the figure, GLCapsNet model is the one that best captures that behaviour, at the same time it correctly predicts the context.

Second, in Figure 7, local dynamics are represented thanks to the local routing coefficients, which provide a 2D map for each feature-condition relationship. Moreover, *LocalPresence* constitutes another 2D map for each condition. Each of the 12 visualizations of the figure is built by concatenating maps which belong to the same group of conditions. For instance, the 3 Daytime *LocalPresence* maps are considered at the top-left graph, similarly to the graph represented on its right using the three Daytime-RGB routing coefficient maps (Morning-RGB, Evening-RGB and Night-RGB). Then, an argmax function is applied pixel-wise along the conditions dimension to classify each spatial location according to them. Each row of sub-figures is dedicated to each group of conditions (Daytime, Weather, Landscape), being each contextual condition represented by a color for the sake of interpretation.

Third, if we revert the argmax function and expand the maps horizontally, each column of the Figure 7 is represented as each sub-figure in Figure 8, where all the local dynamics' maps are visualized for the example frame. By construction, when aggregating maps from sub-figures 8b, 8c and 8d, we obtain the *LocalPresence* maps of sub-figure 8a. Therefore, when applying the argmax function, the results of Figure 7 are obtained (keeping aggregation properties between the routing coefficient columns and the *Presence* column as mentioned). Finally, *GlobalPresence* values shown at Figure 6 are built by aggregating pixel's values before argmax function in the *Presence* column from Figure 7.

Considering each local map from subfigures 8b (RGB), 8c (OF) and 8d (Segmentation) as visual attention maps, input features from Figure 6 can be filtered to answer the question declared at the beginning of this section. *LocalPresence* maps in sub-figure 8a would behave as the combination of the

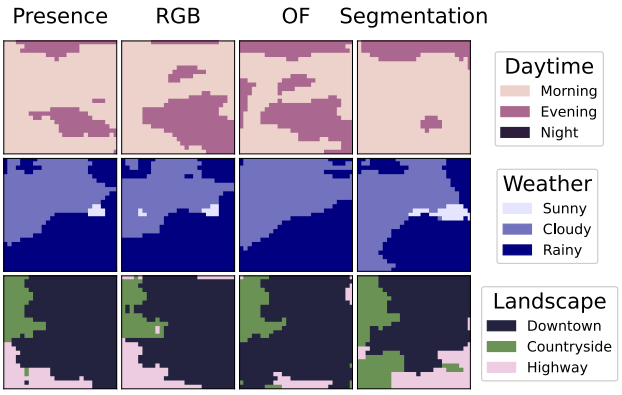
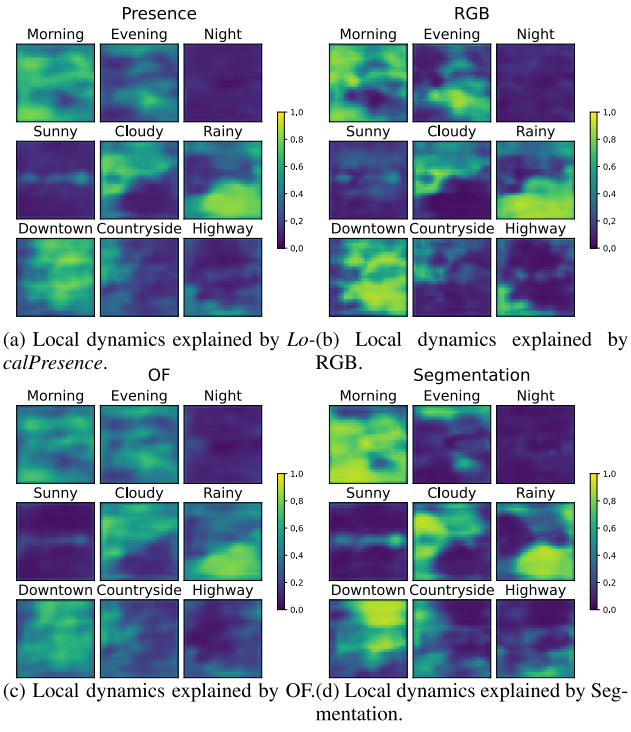


FIGURE 7. Local dynamics for frame 4153 in video 47 from DR(eye)VE dataset [49]. They are represented as classification maps for each group of conditions. At first column, dynamics are explained as *Presence* does. Then, they are decomposed into routing coefficients from the features point of view: RGB, OF and Segmentation.



(a) Local dynamics explained by *LocalPresence*. (b) Local dynamics explained by RGB. (c) Local dynamics explained by OF. (d) Local dynamics explained by Segmentation.

FIGURE 8. Local dynamics for frame 4153 in video 47 from DR(eye)VE dataset [49]. The information is the same as in Figure 7 but, in this case, each condition group is unrolled into 3 columns (one per condition). Hence, there are 4 unrolled blocks: *Presence*, RGB, OF and Segmentation.

visual features' ones. To simplify the analysis, the classification maps from Figure 7 can be used. As some examples, Daytime maps explain that some darker regions of the road and darker clouds belong to Evening, while the clearest regions and clouds belong to Morning. Weather maps mainly split the scene into the clear car (Sunny), the wet road (Rainy) and the Cloudy sky. Landscape maps perceive that vegetation is related to Countryside, while road and buildings are related to Downtown, and it seems that the bottom side has shared patterns with the ones that occur in Highway conditions.

Based on this analysis, one more step in depth would be to aggregate that information for multiple frames and

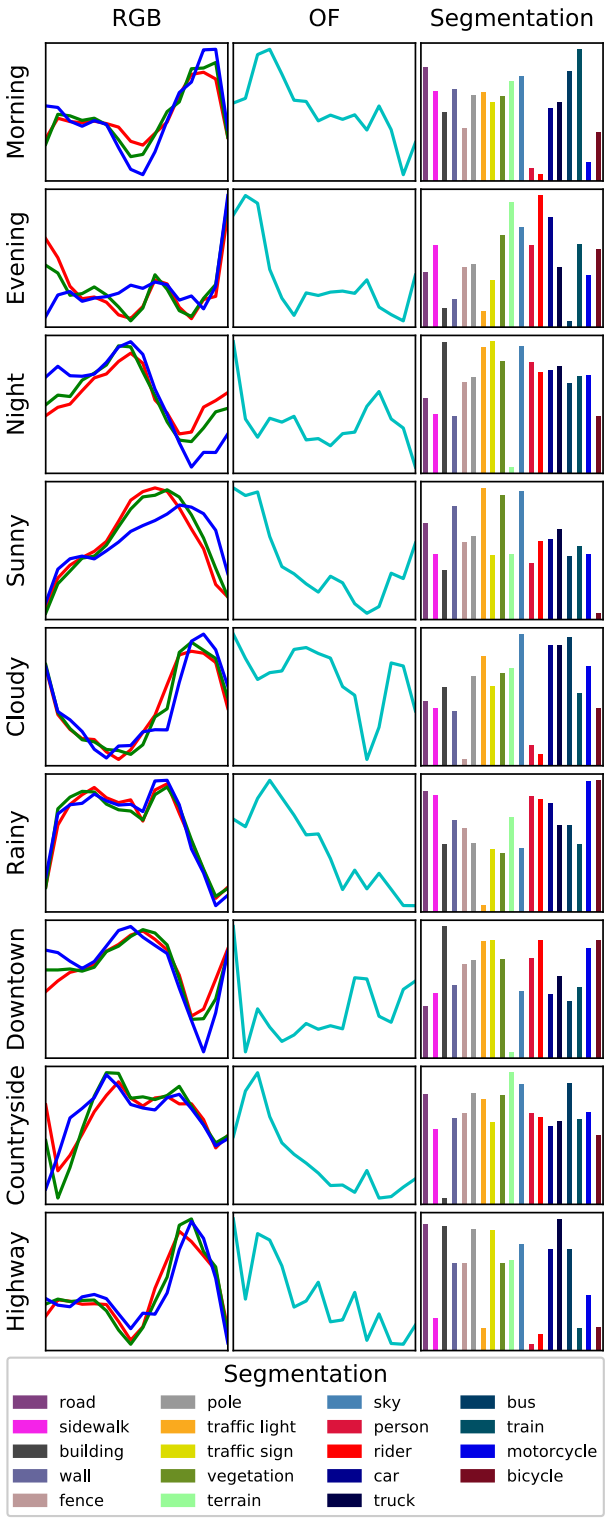


FIGURE 9. Local dynamics for DR(eye)VE [49] shown as estimations of the posterior probability distribution for the action of attending given a feature and a condition. Each histogram has been built by filtering the 2D representation of each input feature using the local routing coefficients which correspond to each pair feature-condition.

study the general behaviour of GLCapsNet. For that purpose, Figure 9 summarizes the aforementioned information, distinguishing between contextual conditions (rows) and visual features (columns) point of view. In total, the information coming from 27 maps is summarized (one per cell in the figure).

Nine thresholds defined between 0 and 1 are applied to each routing coefficients map, resulting in 27 binary masks for each threshold. These thresholds emulate the human visual attention focusing at some location. As the absolute contributions are described in Section IV-D, routing coefficient maps can change their scale to express relative results and interpretations. Therefore, each map is normalized between 0 and 1, as the thresholds are in that range.

Masks are 27×27 pixels resolution, so they are first upsampled to the original input size (112×112 pixels) and used then to select the visual regions from which to gather information. In addition, the frames' features are only taken into account at the conditions (rows from Figure 9) to which the frame belongs (based on the condition GT, not on the predicted condition). A histogram is computed separately for each channel in RGB. Another histogram is built for OF, but in this case the channel used is the Euclidean norm of the optical flow. Segmentation requires a histogram along the Cityscapes categories [39]. For each pair feature-condition (channel-condition in the case of RGB), this process results in 9 histograms, one per threshold. These histograms are averaged to obtain an estimation of the joint probability distribution between the feature and the action of attending, given a condition. To achieve an estimate of the posterior probability distribution for the action of attending given a feature and a condition, the prior probability distribution of the features given a condition is estimated as a histogram computed by following the same process, except for the application of the binary masks (the whole input feature map is used). This posterior probability distribution is the one which shows relevant information. The formulation is as follows:

$$Pr(att|f, c) = \frac{Pr(att, f|c)}{Pr(f|c)} \tag{15}$$

where att is the action of attending, f refers to a feature and c refers to a condition.

According to Figure 9, it is easy to realize that discriminative patterns are extracted for each pair feature-condition. Focusing on RGB column, histograms are interpretable from a light intensity point of view. Morning is represented by brighter regions than for Night, being Evening in the middle. Sunny locations are brighter than Rainy ones, and Cloudy achieves high peaks (maybe due to raindrops). In terms of occluded regions from the sunlight, assuming that Downtown > Countryside > Highway as an hypothesis, RGB column is showing that behavior.

From the OF perspective, an interesting insight is that very different OF patterns are achieved for each condition. It should be noted that these patterns are not modeling car speed, as the attention maps (routing coefficients) focus on relevant space regions but not in the absolute velocity. As it is shown, slow motions are salient for most of the conditions, but fast motions peaks are only appearing for some

particular cases. Night has a fast motion peak because the observer must drive with caution and, therefore, attend to fast movements. Sunny and Cloudy have also fast motion peaks. This might be because there are more moving elements in these conditions than in Rainy, where social activity is reduced. The same behaviour is observed for Downtown, where social activity is higher than in Countryside and Highway, guiding the attention to dangerous areas.

Finally, regarding Segmentation histograms, which are in log scale for the probability axis, some conclusions are extracted. It seems that the model gives less attention to people and riders in Morning, Cloudy and Highway conditions, probably because these semantics are not so surprising or meaningful for the model to build attention maps at the described conditions (we expect them with high probability in Morning, and with low probability in the other two conditions, but they do not constitute conspicuous elements). There seems to be a lot of cars in all conditions, but this actually refers to the observer's car. As interesting examples of conditions, it is shown that traffic lights are less important in Highway than in the other landscapes. Moreover, buildings are less important in Countryside than in Downtown or Highway (since vegetation covers the scene). Finally, the sky is sometimes occluded in Downtown and Rainy conditions, being less attended by the model.

F. TOWARDS REAL-TIME EYE FIXATION PREDICTION: MODEL STRENGTHS AND LIMITATIONS

With regard to the interpretability achieved by our proposal, the great advantages of capsules are undeniable. They result in specialized sets of filters based on particular conditions or properties of a scene, which allow for context-aware visual attention understanding and, at the same time, could signify a step forward to safer and more user-friendly autonomous driving systems. However, even though our results have shown the potential capability of capsules when predicting eye fixations in several challenging situations, such as making decisions at crossroads (e.g. Figures 6 and SM.7) or steering when multiple vehicles are present (e.g. Figure SM.16), we are still a long away from reaching the accuracy of human drivers' attention and, consequently, additional research should be accomplished towards real-time visual attention estimation.

When considering the deployment of such a system in a real scenario, it is worth mentioning some implications. First, it would be necessary to define a more diverse set of contexts, according to the ones expected in a real setting. This labelling task would not require a significant human effort when compared to the video recording and eye-tracking annotation tasks, as context annotation would be probably done by default, given that each driving session corresponds to a single context.

Second, the quality of GT eye fixations used for training the system has to be measured and monitored, in order to compensate for bias effects derived from the expertise of drivers conducting the eye-tracking annotations. Hybrid approaches for eye fixation prediction based on the ensemble of future BU saliency models, which are able to determine the conspicuousness of high-level concepts and actions, and TD architectures such as GLCapsNet, which can be extended by using temporal modules for dynamic behaviour modeling, could enable to reach this goal, notably enhancing the performances reported in the literature so far.

Third, focusing on the implementation of the system in the perception module of an autonomous vehicle, we highlight the following two considerations: 1) The physical integration would need a capable GPU, which could be already required by the autonomous vehicle itself, thus this is not an additional requirement, and 2) Sub-systems could be easily integrated, as they only require the video sequence as input, providing as output the interpretable visual attention maps to filter the scene information, along with the context predictions to condition subsequent autonomous decisions. Last but not least, we have shown that the inference speed of GLCapsNet is quite fast and not so far from offering a real-time smooth video sequence of visual attention maps.

V. CONCLUSION

In this article, we have presented Global-Local Capsule Networks, which introduces for the first time, to the best of our knowledge, the use of capsules for context-aware visual attention modeling. This type of networks allows to learn concepts and relationships between them at both global and local levels. Our approach has been validated in an autonomous driving scenario, using DR(eye)VE [49] dataset. The system's formulation is based on [19], but with important differences. While their purpose is to decode capsules using their modulus for the main task, our goal is to give semantic meaning to the latent and hierarchical capsular structure. This structure self-organizes relationships between internal concepts (latent representations of visual features and contextual conditions) dynamically for each new input frame, at the same time it provides an endpoint for each discriminative concept (contextual conditions). These endpoints are called *Presence* and are used to predict these conditions. In addition, the specialization of the model by conditions serves as a guide for the CNN to efficiently route information between layers.

We have demonstrated that the proposed GLCapsNet matches or outperforms several baselines and similar approaches in the literature in terms of eye fixation prediction. *Emphasize* function enhances conditions prediction improving the results obtained by the specialized capsule network proposed. Finally, the model has interpretability capabilities: it is able to express feature-condition relationships both globally and attending to particular visual regions.

Adding capsules at context level, modeling motion patterns and temporal sequences of fixations with LSTM, ConvLSTM or Conv3D units [63]–[65], or sampling data with better strategies would be some interesting future lines of research. New horizons: *What if my data or my system is required to change its information structure fast, dynamically and to other complex structures?* This is the real case of a

situation where data structure could change based on particular needs, affecting to the dataset construction (new video recordings and context labels) and model re-training (for new information maps). This is a challenge proposed for One Shot Learning line of research, where new capsule types could be learned fast and using only a few more data.

ACKNOWLEDGMENT

The authors would like to thank the authors from DR(eye)VE Project [49] for the support provided during this work, as well as the Multimedia Processing Group from the Universidad Carlos III de Madrid for their entire personal and academic implication.

REFERENCES

[1] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019.

[2] G. Ciaparrone, F. Luque Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, Mar. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231219315966

[3] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, and R. S. M. Goh, "AnomalyNet: An anomaly detection network for video surveillance," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2537–2550, Oct. 2019.

[4] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition Cooperation Neural Nets*. Cham, Switzerland: Springer, 1982, pp. 267–285.

[5] J. M. Wolfe, "Approaches to visual search: Feature integration theory and guided search," in *The Oxford Handbook of Attention*. New York, NY, USA: Oxford Univ. Press, 2014, pp. 11–55.

[6] M. M. Chun, "Contextual cueing of visual attention," *Trends Cognit. Sci.*, vol. 4, no. 5, pp. 170–178, May 2000.

[7] J. M. Wolfe, M. L.-H. Võ, K. K. Evans, and M. R. Greene, "Visual search in scenes involves selective and nonselective pathways," *Trends Cognit. Sci.*, vol. 15, no. 2, pp. 77–84, Feb. 2011.

[8] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," in *Matters of Intelligence*. Cham, Switzerland: Springer, 1987, pp. 115–141.

[9] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Rev. Neurosci.*, vol. 2, no. 3, pp. 194–203, 2001.

[10] A. L. Yarbus, *Eye Movements and Vision*. Cham, Switzerland: Springer, 2013.

[11] A. Borji and L. Itti, "Defending Yarbus: Eye movements reveal observers' task," *J. Vis.*, vol. 14, no. 3, p. 29, 2014.

[12] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[13] C. Sharma, P. Bhavsar, B. Srinivasan, and R. Srinivasan, "Eye gaze movement studies of control room operators: A novel approach to improve process safety," *Comput. Chem. Eng.*, vol. 85, pp. 43–57, Feb. 2016.

[14] C. J. Howard, I. D. Gilchrist, T. Troscianko, A. Behera, and D. C. Hogg, "Task relevance predicts gaze in videos of real moving scenes," *Exp. Brain Res.*, vol. 214, no. 1, p. 131, 2011.

[15] L. Pomarjanschi, M. Dorr, and E. Barth, "Gaze guidance reduces the number of collisions with pedestrians in a driving simulator," *ACM Trans. Interact. Intell. Syst.*, vol. 1, no. 2, pp. 1–14, Jan. 2012.

[16] N. Du, J. Haspiel, Q. Zhang, D. Tilbury, A. Pradhan, J. Yang, and L. Robert, "Look Who's talking now: Implications of AV's explanations on driver's trust, AV preference, anxiety and mental workload," *Transp. Res. C, Emerg. Technol.*, vol. 10, pp. 428–442, Jul. 2019.

[17] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3856–3866.

[18] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–15.

[19] R. LaLonde and U. Bagci, "Capsules for object segmentation," 2018, *arXiv:1804.04241*. [Online]. Available: http://arxiv.org/abs/1804.04241

[20] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 185–207, Jan. 2013.

[21] J. M. Wolfe and W. Gray, "Guided search 4.0," in *Integrated Models of Cognitive Systems*. New York, NY, USA: Oxford Univ. Press, 2007, pp. 99–119.

[22] M. Hayhoe and D. Ballard, "Eye movements in natural behavior," *Trends Cognit. Sci.*, vol. 9, no. 4, pp. 188–194, Apr. 2005.

[23] J. W. Bisley, "Neuronal activity in the lateral intraparietal area and spatial attention," *Science*, vol. 299, no. 5603, pp. 81–86, Jan. 2003.

[24] Y. Sun and R. Fisher, "Object-based visual attention for computer vision," *Artif. Intell.*, vol. 146, no. 1, pp. 77–123, May 2003.

[25] A. Borji, M. N. Ahmadabadi, B. N. Araabi, and M. Hamidi, "Online learning of task-driven object-based visual attention control," *Image Vis. Comput.*, vol. 28, no. 7, pp. 1130–1145, Jul. 2010.

[26] R. Desimone and J. Duncan, "Neural mechanisms of selective visual attention," *Annu. Rev. Neurosci.*, vol. 18, no. 1, pp. 193–222, Mar. 1995.

[27] L. Itti, N. Dhavale, and F. Pighin, "Realistic avatar eye and head animation using a neurobiological model of visual attention," *Appl. Sci. Neural Netw., Fuzzy Syst., Evol. Comput. VI*, vol. 5200, pp. 64–79, Dec. 2003.

[28] R. J. Peters and L. Itti, "Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[29] A. Borji, "Saliency prediction in the deep learning era: Successes, limitations, and future challenges," 2018, *arXiv:1810.03716*. [Online]. Available: http://arxiv.org/abs/1810.03716

[30] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, and F. Durand, "Where should saliency models look next?" in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 809–824.

[31] Q. Lai, S. Khan, Y. Nie, S. Hanqiu, J. Shen, and L. Shao, "Understanding more about human and machine attention in deep neural networks," *IEEE Trans. Multimedia*, early access, Jun. 6, 2020, doi: 10.1109/TMM.2020.3007321.

[32] M.-A. Fernandez-Torres, I. Gonzalez-Diaz, and F. Diaz-de-Maria, "Probabilistic topic model for context-driven visual attention understanding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1653–1667, Jun. 2020.

[33] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, p. 6, Feb. 2017.

[34] M. P. Muresan, I. Giosan, and S. Nedevschi, "Stabilization and validation of 3D object position using multimodal sensor fusion and semantic segmentation," *Sensors*, vol. 20, no. 4, p. 1110, Feb. 2020.

[35] H. Kim, J. Cho, D. Kim, and K. Huh, "Intervention minimized semi-autonomous control using decoupled model predictive control," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 618–623.

[36] A. Arikan, A. Kayaduman, S. Polat, Y. Simsek, I. C. Dikmen, H. G. Bakir, T. Karadag, and T. Abbasov, "Control method simulation and application for autonomous vehicles," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Sep. 2018, pp. 1–4.

[37] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state of the art," 2017, *arXiv:1704.05519*. [Online]. Available: http://arxiv.org/abs/1704.05519

[38] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[39] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[40] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, "DrivingStereo: A large-scale dataset for stereo matching in autonomous driving scenarios," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 899–908.

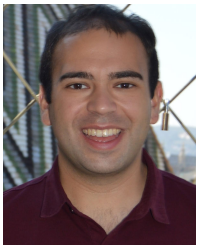
[41] C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, P. Furgale, T. Sattler, and M. Pollefeys, "3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection," *Image Vis. Comput.*, vol. 68, pp. 14–27, Dec. 2017.

[42] C. Zhou, F. Guney, Y. Wang, and A. Geiger, "Exploiting object similarity in 3D reconstruction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2201–2209.

[43] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3061–3070.

[44] V. Vaquero, A. Sanfeliu, and F. Moreno-Noguer, "Hallucinating dense optical flow from sparse lidar for autonomous vehicles," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1959–1964.
[45] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. Change Loy, "Robust multi-modality multi-object tracking," 2019, *arXiv:1909.03850*. [Online]. Available: http://arxiv.org/abs/1909.03850
[46] V. John, K. Yoneda, B. Qi, Z. Liu, and S. Mita, "Traffic light recognition in varying illumination using deep learning and saliency map," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 2286–2291.
[47] T. Deng, A. Chen, M. Gao, and H. Yan, "Top-down based saliency model in traffic driving environment," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 75–80.
[48] T. Deng, K. Yang, Y. Li, and H. Yan, "Where does the driver look? Top-down-based saliency detection in a traffic driving environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2051–2062, Jul. 2016.
[49] A. Palazzi, D. Abati, S. Calderara, F. Solera, and R. Cucchiara, "Predicting the driver's focus of attention: The DR(eye)VE project," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1720–1733, Jul. 2019.
[50] S. Alletto, A. Palazzi, F. Solera, S. Calderara, and R. Cucchiara, "DR(eye)VE: A dataset for attention-based tasks with applications to autonomous and assisted driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 54–60.
[51] Y. Xia, D. Zhang, J. Kim, K. Nakayama, K. Zipser, and D. Whitney, "Predicting driver attention in critical situations," 2017, *arXiv:1711.06406*. [Online]. Available: http://arxiv.org/abs/1711.06406
[52] T. Deng, H. Yan, L. Qin, T. Ngo, and B. S. Manjunath, "How do drivers allocate their potential attention? Driving fixation prediction via convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 2146–2154, May 2020.
[53] N. Zhang, S. Deng, Z. Sun, X. Chen, W. Zhang, and H. Chen, "Attention-based capsule networks with dynamic routing for relation extraction," 2018, *arXiv:1812.11321*. [Online]. Available: http://arxiv.org/abs/1812.11321
[54] B. McIntosh, K. Duarte, Y. S. Rawat, and M. Shah, "Multi-modal capsule routing for actor and action video segmentation conditioned on natural language queries," 2018, *arXiv:1812.00303*. [Online]. Available: http://arxiv.org/abs/1812.00303
[55] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, "DeepCaps: Going deeper with capsule networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10725–10733.
[56] J. Choi, H. Seo, S. Im, and M. Kang, "Attention routing between capsules," 2019, *arXiv:1907.01750*. [Online]. Available: http://arxiv.org/abs/1907.01750
[57] Y.-H. Hubert Tsai, N. Srivastava, H. Goh, and R. Salakhutdinov, "Capsules with inverted dot-product attention routing," 2020, *arXiv:2002.04764*. [Online]. Available: http://arxiv.org/abs/2002.04764
[58] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L 1 optical flow," in *Proc. Joint Pattern Recognit. Symp.* Cham, Switzerland: Springer, 2007, pp. 214–223.
[59] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," 2014, *arXiv:1405.3531*. [Online]. Available: http://arxiv.org/abs/1405.3531
[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[61] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 740–757, Mar. 2019.
[62] V. Leboran, A. Garcia-Diaz, X. R. Fdez-Vidal, and X. M. Pardo, "Dynamic whitening saliency," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 893–907, May 2017.
[63] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5142–5154, Oct. 2018.
[64] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," 2016, *arXiv:1603.08199*. [Online]. Available: http://arxiv.org/abs/1603.08199
[65] L. Jiang, M. Xu, and Z. Wang, "Predicting video saliency with Object-to-Motion CNN and two-layer convolutional LSTM," 2017, *arXiv:1709.06316*. [Online]. Available: http://arxiv.org/abs/1709.06316

JAVIER MARTÍNEZ-CEBRIÁN received the degree in telecommunications engineering and the double master's degree in telecommunications engineering and in multimedia and communications from the Universidad Carlos III de Madrid, Spain, in 2014 and 2016, respectively. He has more than four years of experience in deep learning and computer science in general, working on visual attention, object detection, text classification, named entity recognition, and social network analysis and scheduling. His other research interests include explainable AI, meta-learning, self-attention, and self-supervised strategies. He has also worked at bringing AI software applications to production using cloud and microservices technologies.



MIGUEL-ÁNGEL FERNÁNDEZ-TORRES received the degree in audiovisual systems engineering and the master's and Ph.D. degrees in multimedia and communications from the Universidad Carlos III de Madrid, Spain, in 2013, 2014, and February 2019, respectively. During the Ph.D. period, his research has been related to spatio-temporal visual attention modeling and understanding, applying both Bayesian networks and deep learning. He is currently an Assistant Professor and a Researcher with the Multimedia Processing Group, Signal Theory and Communications Department, Universidad Carlos III de Madrid. In addition to his work on visual attention, he has participated in projects related to some of his other interests within the computer vision field, which include image and video analysis, and medical image analysis and classification. He had also the opportunity to study at the Technische Universität Wien, Vienna, Austria, during the bachelor's degree, in 2013, and to do a Ph.D. stay at the Visual Perception Laboratory, Purdue University, West Lafayette, IN, USA, in 2016.

FERNANDO DÍAZ-DE-MARÍA (Member, IEEE) received the degree in telecommunication engineering and the Ph.D. degree from the Universidad Politécnica de Madrid, Madrid, Spain, in 1991 and 1996, respectively. Since October 1996, he has been an Associate Professor with the Department of Signal Processing and Communications, Universidad Carlos III de Madrid, Madrid. He is a coauthor of numerous articles in peer-reviewed international journals, several book chapters, and a number of papers in national and international conferences. His current research interests include deep learning, image and video processing, and computer vision. He has led numerous projects and contracts in the fields mentioned.