Employing Deep Part-Object Relationships for Salient Object Detection

Yi Liu
Xidian University

Qiang Zhang*
Xidian University

Dingwen Zhang
Xidian University

Jungong Han*
University of Warwick

yLiu_89@stu.xidian.edu.cn, qzhang@xidian.edu.cn
zhangdingwen2006yyy@gmail.com, jungonghan77@gmail.com

Abstract

Despite Convolutional Neural Networks (CNNs) based methods have been successful in detecting salient objects, their underlying mechanism that decides the salient intensity of each image part separately cannot avoid inconsistency of parts within the same salient object. This would ultimately result in an incomplete shape of the detected salient object. To solve this problem, we dig into part-object relationships and take the unprecedented attempt to employ these relationships endowed by the Capsule Network (CapsNet) for salient object detection. The entire salient object detection system is built directly on a Two-Stream Part-Object Assignment Network (TSPOANet) consisting of three algorithmic steps. In the first step, the learned deep feature maps of the input image are transformed to a group of primary capsules. In the second step, we feed the primary capsules into two identical streams, within each of which low-level capsules (parts) will be assigned to their familiar high-level capsules (object) via a locally connected routing. In the final step, the two streams are integrated in the form of a fully connected layer, where the relevant parts can be clustered together to form a complete salient object. Experimental results demonstrate the superiority of the proposed salient object detection network over the state-of-the-art methods.

1. Introduction

Salient object detection aims to grab the most attractive object and segment it out from the backgrounds in an image. Serving as a preprocessing step, it has been widely applied for a variety of computer vision applications, including image segmentation [13, 34], image fusion [14], object recognition [36, 41], image and video compression [11, 12, 18], image retrieval [5, 10], etc.

Traditional salient object detection methods [3,26,27,39, 45] are mostly based on hand-crafted features, which are trivial for further improvements. The development of Convolutional Neural Networks (CNNs) has successfully broken the limits of hand-crafted features by learning deep features and thus substantial improvements have been made in the last three years [25, 29, 33].

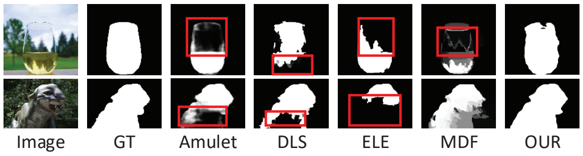


Figure 1. Some problems arose in existing CNNs based salient object detection methods. Inconsistent saliency values or even some "holes" (marked by the red boxes) appear within the salient object.

Existing CNNs based salient object detection methods attempt to learn rich deep features at multiple scales such as the contrast information of the image parts, which in turn infer the saliency of each part in the image. However, this mechanism does not take into account the relationships between the object parts and the complete salient object, thus giving rise to several problems. For example, as shown in Fig. 1, inconsistent saliency values are assigned to different parts within the salient object, thus resulting in a non-uniform segmentation of the salient object. In the worse scenario, as highlighted by the red boxes in Fig. 1, some un-prominent parts within the salient object are mistakenly labeled as non-salient such that a few "holes" appear on the salient object.

As can be observed from Fig. 2, a salient object is usually composed of several associated parts. For instance, the flower in the second row of Fig. 2 consists of two parts including stamens and petals. In turn, the two parts (stamens and petals) can make up an object (flower), which is based on the fact that stamens and petals share the familiar properties of the flower. This reveals that the relationships do exist between parts and objects. In a full image, based on the above discussions, those parts familiar to an object will be clustered together to make a complete object. Inspired by these observations, we introduce the property of

*Equally corresponding authors.

part-object relationships for salient object detection in this paper, which can solve the problem of incomplete segmentation of the salient object.

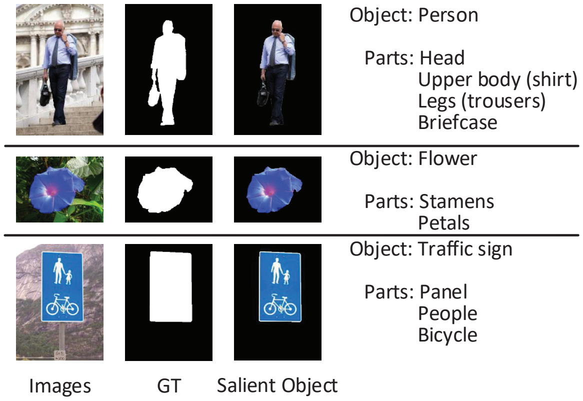


Figure 2. Illustrations of the part-object relationships for salient object detection.

Recently, a new architecture termed as Capsule Network (CapsNet) [15,16,37] has shown promising results in recognizing digits from images. A capsule is a group of neurons whose outputs represent different properties of the entity, such as an object or an object part. In the matrix CapsNet [16], each capsule contains a pose matrix and an activation, which characterize the pose attributes and the existence probability of the capsule, respectively. Each capsule votes for the pose matrix of one capsule in the layer above by multiplying its own pose matrix and trainable view-point transformation matrices, which takes the agreement between these capsule vectors into account to form meaningful part-object relationships. In other words, a familiar object can be detected by looking for agreement between those votes for its pose matrix. Owing to this special property, CapsNet can assign parts to the familiar object based on the part-object relationships, which makes it become a natural platform to implement part-object relationships for salient object detection.

However, using CapsNet for salient object detection does not seem to be that straightforward due to: 1) each low-level capsule essentially belongs to a subset but not a full set of high-level capsules. Allowing each low-level capsule (part) to vote for all the high-level ones (object) will sometimes generate noisy assignment, thus giving rise to performance declines. For example, as shown in Fig. 3, those capsules in the CapsNet are less distinguishable to identify the salient object. 2) as we all know, the original CapsNet has a much heavier computation complexity than CNNs when applied to small digital images classification. It will become unaffordable if we directly apply CapsNet for large-scale dense prediction of salient object detection, which is a much more complicated task, compared to image classification.

To address the above problems, we propose, in this paper, a deep Two-Stream Part-Object Assignment Network (TSPOANet) to detect the salient object. Specifically, the proposed model divides those capsules constructed from the image features into two streams. Within each stream, taking the part-object relationships into account, low-level capsules will be assigned to their familiar ones in the layer above based on the part-object relationships. In such way, the relevant parts will be clustered together to form a salient object. Therefore, the salient object can be predicted and segmented out from the background. Because the proposed TSPOANet assigns each capsule to one stream of high-level capsules but not all the high-level ones, it alleviates redundancy and thus the noisy assignment to some extent.* As shown in Fig. 3, those capsules of the proposed TSPOANet, especially ones marked in red, are much more discriminative when identifying the salient object from the background. Furthermore, due to much less parameters, training TSPOANet is easier than training the original CapsNet. As TSPOANet takes the relationships of part-object into account, the object parts can be naturally linked to its belonged salient object. This enables to overcome the problem of incomplete or non-uniform segmentation of the detected salient object, which is still an unsolved problem in traditional CNNs based methods (see Fig. 2).

Our contributions are summarized as follows:

(1) We incorporate a new property, i.e., part-object relationships, in salient object detection, which is implemented by CapsNet. To the best of our knowledge, this is the first attempt to apply CapsNet for salient object detection.

(2) We propose a deep TSPOANet for salient object detection, which systematically adopts a two-stream strategy to implement the CapsNet. This effectively reduces the searching space when a low-level capsule votes for the high-level capsules. Doing so gets the complexity of CapsNet significantly reduced while diminishing the possibility having noisy assignments.

(3) We compare our approach with 9 state-of-the-art methods on five datasets. The results consistently show the superiority of our algorithm on various datasets.

2. Related Work

2.1. CNNs Based Salient Object Detection

Traditionally, most of salient object detection methods [4, 6, 8, 9, 19, 20, 22, 31, 32, 35, 45] are based on handcrafted features. Readers can gain a comprehensive understanding about these methods from [3]. The development of CNNs has achieved substantial improvements for saliency

*In our experiments, we find that non-convergence occurs for the proposed model with 4 and 8 streams, each of which has too few capsules. However, the model works with 2 streams. This indicates that each stream has enough familiar high-level capsules corresponding to low-level capsules in the case of 2 streams but not enough in the case of 4 or 8 streams.

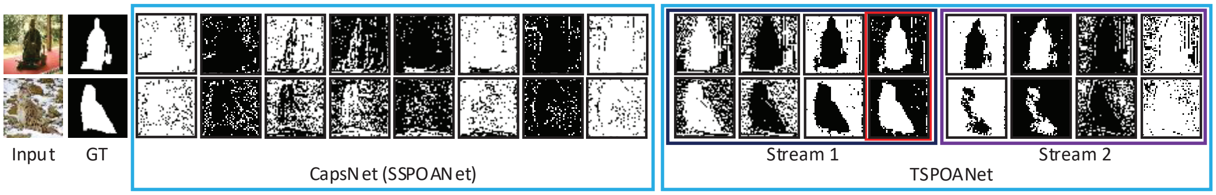


Figure 3. Capsules of the second convolutional capsule layer in TSPOANet and CapsNet (i.e., Single-Stream POANet (SSPOANet)). Due to high redundancy caused by fully connected voting, those capsules of CapsNet are trivial to identify the salient object. In contrast, TSPOANet is able to reduce redundancy by locally connected voting to some extent, leading to more discriminative capsules to identify the salient object from the background.

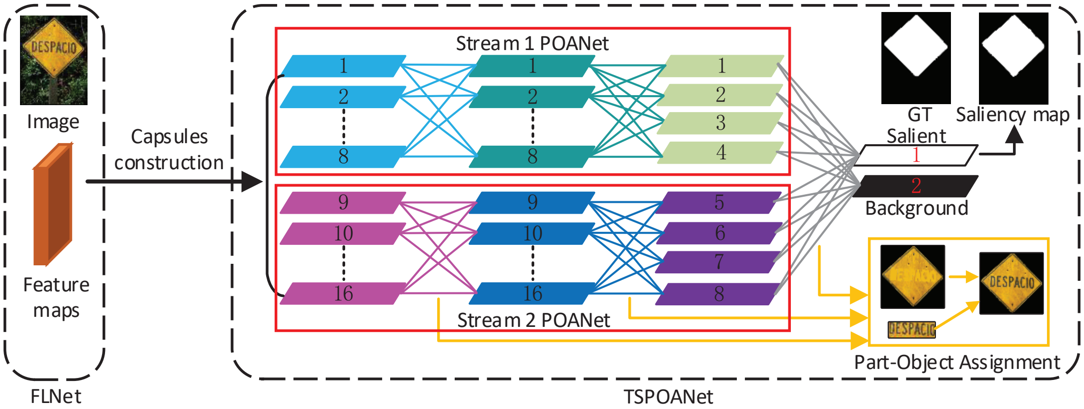


Figure 4. The architecture of the proposed deep salient object detection network consists of two subnetworks, i.e., FLNet and TSPOANet. The image is first input to FLNet to learn deep features (as described in the following Fig. 5), which are then fed to TSPOANet. In TSPOANet, those deep feature maps are first transformed into several capsules. These capsules are divided into two groups, which are fed to two streams to explore the part-object relationships. During the process of part-object assignment in each stream, each low-level capsule is assigned to each high-level capsule with a probability that is learned. Based on the part-object relationships, relevant parts will be assigned to the familiar object. In such way, the salient object will be segmented out from the background, resulting in the saliency map.

detection. Zhao *et al.* [53] modeled a unified deep learning framework by jointly taking into account global context and local context. Li *et al.* [25] used CNNs to learn multiscale deep features for saliency detection. Liu *et al.* [29] proposed an end-to-end deep hierarchical saliency detection framework, which first made a coarse global prediction by learning various global saliency cues, and then refined the coarse prediction by making up the discarded detailed information via a hierarchical recurrent CNN. Zhang *et al.* [50] proposed a multi-level feature aggregation network for salient object detection by integrating multi-level features into multiple resolutions, which well incorporated low-level fine details and high-level semantic knowledge. Liu *et al.* [30] learned to generate a pixel-level contextual attention, which was formulated by incorporating global context and local context. Zhang *et al.* [48] designed a gated bi-directional message passing module to integrate multilevel features in the shallow-to-deep and deep-to-shallow directions, which were complementary and robust for detecting salient objects.

2.2. CapsNet

Hinton *et al.* [15] introduced the concept of capsule. A capsule is a group of neurons and represents the instantiation parameters of a specific type of entity, such as pose (position, size, orientation), deformation, texture, etc. It was a nice idea, but it did not get much attention until Sabour *et al.* [37] implemented a vector CapsNet, in which the output of a capsule is a vector. The length of the activity vector represented the existence probability of the entity while its orientation represented the instantiation parameters. An iterative dynamic routing algorithm was proposed to assign low-level capsules to their familiar high-level capsules via transformation matrices, which were learned to encode the intrinsic spatial relationship between a part and a whole as well as viewpoint invariant knowledge. Therefore, the iterative routing process solved the problem of assigning parts to familiar objects. One year later, Hinton *et al.* [16] consolidated their work by proposing a matrix CapsNet, in which each capsule contained a pose matrix and an activation probability. The pose matrix and the activation probability were used to represent the pose characteristics and the existence probability, respectively. A capsule in one layer voted for the pose matrix of many different capsules in the layer above by multiplying its own pose matrix and trainable viewpoint-invariant transformation matrices that learned part-whole relationships. A familiar object could be detected by looking for agreement between votes for its pose matrix. An iterative Expectation-Maximization (EM) algorithm was proposed to assign low-level capsules to high-level capsules or parts to wholes by finding tight clusters of high-dimensional votes that agreed in a mist of irrelevant votes.

3. Proposed Salient Object Detection Network

Fig. 4 shows the proposed deep salient object detection network. The input image is first input into the designed Feature Learning Network (FLNet) to achieve more primitive features, which are then fed to the proposed Two-Stream Part-Object Assignment Network (TSPOANet). In TSPOANet, those deep feature maps are first transformed to several capsules, which are followed by two streams of POANet. Within each stream, POANet is designed to assign low-level capsules to familiar ones in the higher layer based on the part-object relationships, in which way relevant parts will be clustered together to compose a salient object. Therefore, the salient object can be segmented out from the background.

3.1. FLNet

FLNet is used to learn deep features for the input image. The details of this network are displayed in Fig. 5. As observed from Fig. 5, the input image (352 × 352 × 3) is first fed into five stacked convolutional layers. To capture more image context information, we add four dilation convolutional layers [46] at each stage, which have the same convolutional kernel size of 3×3 with different dilation rates (1, 3, 5, and 7). In such way, we can capture rich context information under various receptive fields at each stage without increasing the kernel scales. Besides, low-level feature maps help to capture fine details such as object boundaries, while high-level feature maps can grab semantic knowledge. To combine their advantages, these five stages of feature maps are integrated together. Specifically, deeper-level feature maps are integrated with shallower-level ones layer by layer until the shallowest stage, resulting in the integrated feature maps (352 × 352 × 128).

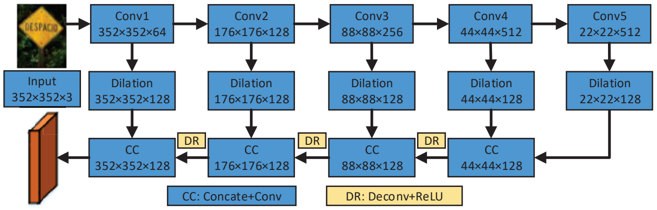


Figure 5. The details of FLNet.

3.2. TSPOANet

TSPOANet is designed to explore the part-object relationships within the input image, which are committed to segmenting the salient object out from the background. It consists of three stages, i.e., capsules construction, two-stream POANet, and capsule classification. The details of TSPOANet will be illustrated as follows.

Capsules construction The feature maps learned by FLNet are first transformed into several capsules (16 capsules in this paper), which is implemented by a Primary Capsule (*PrimaryCaps*) layer. Each capsule consists of a pose matrix (4 × 4) and an activation value, which represent the pose characteristics (such as an object part and an object) and the existence probability of the entity, respectively. Considering the computational memory, we first use two Conv+ReLU layers to transform the integrated feature maps into 88 × 88 × 16. Details of *PrimaryCaps* are shown in Fig. 6.

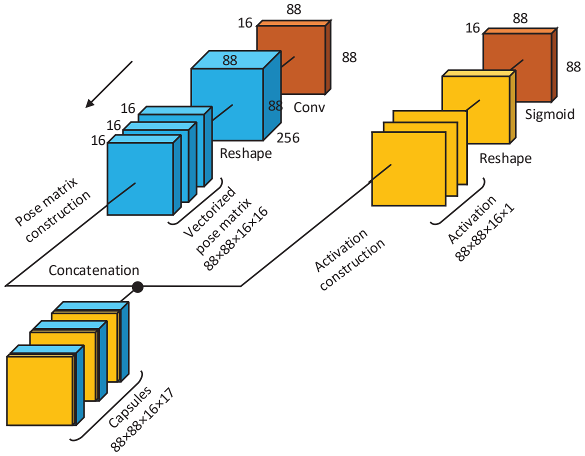


Figure 6. Capsules construction.

Pose matrix construction The 16-channel feature maps (88 × 88 × 16) are first transformed to 256-channel feature maps (88×88×256) via two convolutional layers. The 256-channel feature maps are then reshaped into 88 × 88 × 16 × 16, which is the vectorized pose matrices[†] of 16 capsules.

Activation construction The 16-channel feature maps (88 × 88 × 16) are first transformed to 16-channel feature maps (88 × 88 × 16). The 16-channel feature maps are reshaped into 88 × 88 × 16 × 1, which is the activation information of 16 capsules.

Capsules construction The vectorized pose matrices and activations are concatenated together to construct 16 capsules (88 × 88 × 16 × 17).

Two-stream POANet Those capsules obtained by *PrimaryCaps* are divided into two groups, each of which con-

[†] Here, the pose matrix of each capsule is lengthened as a vector for efficient storage. Dimension 3 is the number of capsules.

tains 8 capsules ($88 \times 88 \times 8 \times 17$). The 8 capsules of each group are reshaped to $88 \times 88 \times 136$. These two groups of capsules are fed to two streams to explore the part-object relationships. This is implemented by two Convolutional Capsule (*ConvCaps1* and *ConvCaps2*) layers. *ConvCaps1* and *ConvCaps2* consist of 8 and 4 capsules in each stream, respectively. Based on the part-object relationships, low-level capsules (parts) will be assigned to familiar high-level capsules (object). The architectures of two streams are the same. We first illustrate one stream of *ConvCaps1* as follows:

Step 1: Enrich the features of capsules. A depth-wise convolution with the stride of 2 and the channel_multiplier of 9 is performed on the output capsules of *PrimaryCaps*, resulting in more rich-feature capsules $44 \times 44 \times 9 \times 136$, which is reshaped into $1936 \times 72 \times 17$. Therefore, the corresponding vectorized pose matrices and activation values are $1936 \times 72 \times [1 : 16]$ and $1936 \times 72 \times [17]$, respectively, where $[\cdot]$ represents the number of channels along the corresponding dimension.

Step 2: Compute the votes of low-level capsules for the adjacent high-level capsules. The vectorized pose matrices are first transformed to the pose matrices M with the dimension of 4×4. Let the pose matrix of the capsule i in layer L be M_i. Between each capsule i in layer L and each capsule j in layer $(L+1)$ is a 4×4 trainable transformation matrix W_{ij}. These W_{ij}s are learned discriminatively. The vote V_{ij} of capsule i in layer L for the capsule j in layer $(L + 1)$ is calculated by multiplying the pose matrix M_i of capsule i and the corresponding transformation matrix W_{ij}, i.e.,

$$V_{ij} = M_i W_{ij}. \tag{1}$$

By Eq. (1), the resulting votes are $1936 \times 72 \times 8 \times 16$.

Step 3: Assign parts (low-level capsules) to wholes (high-level capsules). Assigning parts to wholes can be solved by finding tight clusters of the votes from parts. To achieve this, an iterative Expectation-Maximization (EM) algorithm [16] is used to update the probability, with which a part is assigned to a whole based on the proximity of the vote coming from that part to the votes coming from other parts. This routing algorithm derives segmentation based on the knowledge of familiar shapes, rather than just using low-level cues such as proximity or agreement in color or velocity.

Specifically, the votes and the activation values of low-level capsules are input into the iterative routing algorithm, which will calculate the means ($1936 \times 1 \times 8 \times 16$) and activations ($1936 \times 8$). They are reshaped into vectorized pose matrices ($44 \times 44 \times 8 \times 16$) and activation values ($44 \times 44 \times 8 \times 1$), respectively, which are then concatenated to be the high-level capsules ($44 \times 44 \times 8 \times 17$). Finally, the output is achieved by reshaping the capsules into $44 \times 44 \times 44 \times 136$, which is fed into *ConvCaps2* within the same stream.

ConvCaps2 has the similar architecture with *ConvCaps1* except two points. One difference is that the stride of the depth-wise convolution is 1 in *ConvCaps2* instead of 2 in *ConvCaps1*. Another difference is that *ConvCaps2* reshapes the calculated means and activations by the iterative routing algorithm into $1936 \times 8 \times 16$ and $1936 \times 8 \times 1$ in each stream, respectively.

Capsule classification Those more whole capsules obtained by the two-stream POANet are finally classified to be salient or background, which is implemented by a Class Capsule (*ClassCaps*) layer. The architecture of *ClassCaps* is similar to *Step 2* and *Step 3* in *ConvCaps1*. Through the *ClassCaps* layer, the capsules of two streams will be assigned to two types of capsules corresponding to the salient object and background, in which way some relevant parts will be clustered together to form a salient object. The output of *ClassCaps* is $44 \times 44 \times 2$. After that, three deconvolutional layers are used to transform the detection result into $352 \times 352 \times 2$ that is the same as the resolution of the input image, generating the final saliency map.

3.3. Loss Function

We adopt the cross-entropy loss function used in [48] to train the proposed salient object detection network, i.e.,

$$CE(\mathbf{v}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c \in \{0,1\}} (y(\mathbf{v}_i) = c) \left(\log \left(\hat{y}(\mathbf{v}_i) = c\right)\right), \tag{2}$$

where $\mathbf{v}_i$ represents the location of pixel i. $y(\mathbf{v}_i)$ and $\hat{y}(\mathbf{v}_i)$ represent saliency values of the pixel i in the ground truth and the predicted saliency map, respectively.

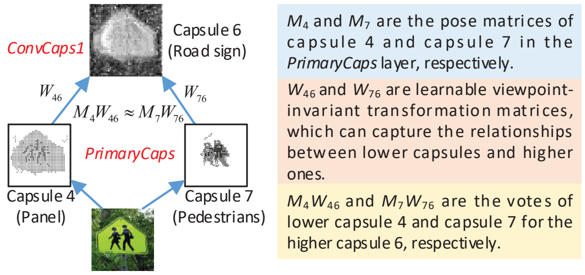
3.4. Insight into TSPOANet

Salient property of part-object relationships The property of part-object relationships for salient object detection is derived from the idea that two low-level capsules will be clustered together to compose a whole if they share familiar properties. In other words, two capsules i and k will be clustered to make the capsule j in the layer above, if

$$M_i W_{ij} \approx M_k W_{kj}. \tag{3}$$

To give a basic and clear insight for the property of part-object relationships employed in salient object detection, we visualize the intermediate layers of a real example (as shown in Fig. 7) based on a Single-Stream POANet (SSPOANet), which is a baseline network by directly adopting the traditional CapsNet after FLNet. Two observations from Fig. 7 are: 1) Capsule 4 and capsule 7 in the *PrimaryCaps* layer indeed capture two parts, i.e., pedestrians and panel, while capsule 6 in the higher *ConvCaps1* layer clearly depicts the whole object; 2) Capsule 4 and capsule 7 vote for capsule 6 by the EM routing algorithm via $M_4 W_{46}$

and M_7W_{76}, where W is a learnable transformation matrix between two capsule layers. W explicitly encodes the relationships between parts and objects. Through voting, capsule 4 and capsule 7 capturing parts make up a higher capsule 6 representing a complete object, i.e., road sign. This way ensures that a complete salient object can be detected in the capsule classification stage, which brings universally high foreground saliency values. In summary, the natural capability of POANet in modeling part-object relationships can address the object part missing problem existing in the CNNs based saliency detectors.



M_4 and M_7 are the pose matrices of capsule 4 and capsule 7 in the *PrimaryCaps* layer, respectively.

W_{46} and W_{76} are learnable viewpoint-invariant transformation matrices, which can capture the relationships between lower capsules and higher ones.

M_4W_{46} and M_7W_{76} are the votes of lower capsule 4 and capsule 7 for the higher capsule 6, respectively.

Figure 7. Illustrations for the part-object relationships. Capsule 4 (panel) and capsule 7 (pedestrians) make up capsule 6 (a whole object) in the higher *ConvCaps1* layer based on their approximately equal votes to the higher capsule 6.

Comparison to CapsNet The differences between our proposed framework and the original CapsNet lie in two folds. Firstly, CapsNet that votes each lower capsule to all higher capsules has a heavy computational complexity. Differently, we apply the two-stream strategy to assign each lower capsule to one stream of higher capsules but not all capsules at the higher layer, which reduces the required computation to some extent. The parameters of the proposed two-stream strategy are 4 times fewer than those of the corresponding two convolutional capsule (*ConvCaps1* and *ConvCaps2*) layers in CapsNet. Secondly, different from using only a Conv+ReLU layer for feature extraction in the original CapsNet, we utilize FLNet to learn better features for TSPOANet. This will improve the performance by a large margin, which is verified in the experiment part.

Comparison to group convolution The major difference between our proposed TSPOANet and group convolution [23, 43, 52] lies in that group convolution performs the convolution operation between low and high groups to achieve more discriminative feature maps, while our proposed TSPOANet performs the vote routing between low and high groups of capsules to explore the part-object relationships.

4. Experiment and Analysis

In this section, numerous experiments and analyses are conducted to verify the effectiveness and superiorities of our proposed deep salient object detection network.

4.1. Benchmark Datasets

We evaluate the performance of our model on five benchmark datasets, details of which are described as follows.

ECSSD [44] contains 1000 images collected from the Internet. These images are with complicated structures. **DUT-OMRON** [45] has 5168 images with different sizes and complex structures. The backgrounds are very complicated to stand out the salient objects. **HKU-IS** [25] consists of 4447 images with multiple disconnected objects. It is divided into 3000 training images and 1447 test images. We evaluate our methods and other state-of-the-arts on the test datasets. **DUTS** [40] contains 10533 training images and 5019 test images. The images in this dataset are with different scenes and various sizes. We use the test dataset to evaluate our model and the compared methods. **PASCAL-S** [28] includes 850 images describing various scenes.

4.2. Evaluation Criteria

We evaluate the performance of our model as well as other state-of-the-art methods from both visual and quantitative perspectives. The quantitative metrics include Precision Recall (PR) curve, average F-measure and Mean Absolute Error (MAE). Given a continuous saliency map S, a binary mask B is achieved by thresholding. Precision is defined as $\Pr ecision = |B \cap G|/|B|$, and recall is defined as $\mathrm{Recall} = |B \cap G|/|G|$, where G is the corresponding ground truth. The PR curve is plotted under different thresholds. The F-measure is an overall performance indicator, which is computed by

$$F_\beta = \frac{\left(1+\beta^2\right) \Pr ecision \times \mathrm{Recall}}{\beta^2 \Pr ecision + \mathrm{Recall}}. \qquad (4)$$

As suggested in [2], $\beta^2 = 0.3$.

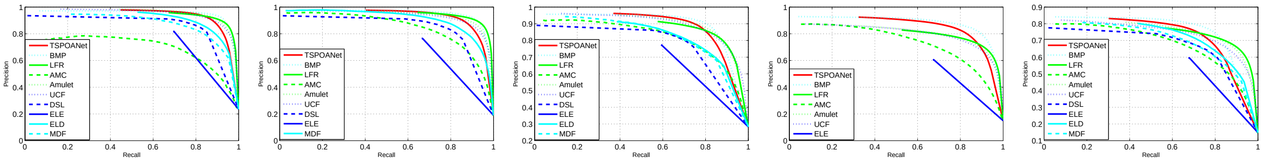
MAE is defined as

$$MAE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |S\left(i,j\right) - G\left(i,j\right)|, \qquad (5)$$

where W and H are the width and height of the image, respectively.

4.3. Implementation Details

The proposed model is implemented in Tensorflow [1]. To avoid over-fitting caused by training from scratch, the five stacked convolutional layers in FLNet are initialized by the Conv1_2, Conv2_2, Conv3_3, Conv4_3, and Conv5_3 of the pretrained VGG16 [38], respectively. The other weights are initialized randomly with a truncated normal ($\sigma = 0.01$), and the biases are initialized to 0. The Adam optimizer [21] is used to train our model with an initial learning rate of 10^6, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The training dataset of DUTS [7] is chosen as the training dataset with horizontal flipping as the data augmentation technique.



ECSSD [44] HKU-IS [25] PASCAL-S [28] DUTS [40] DUT-OMRON [45]

Figure 8. PR curves of different methods.

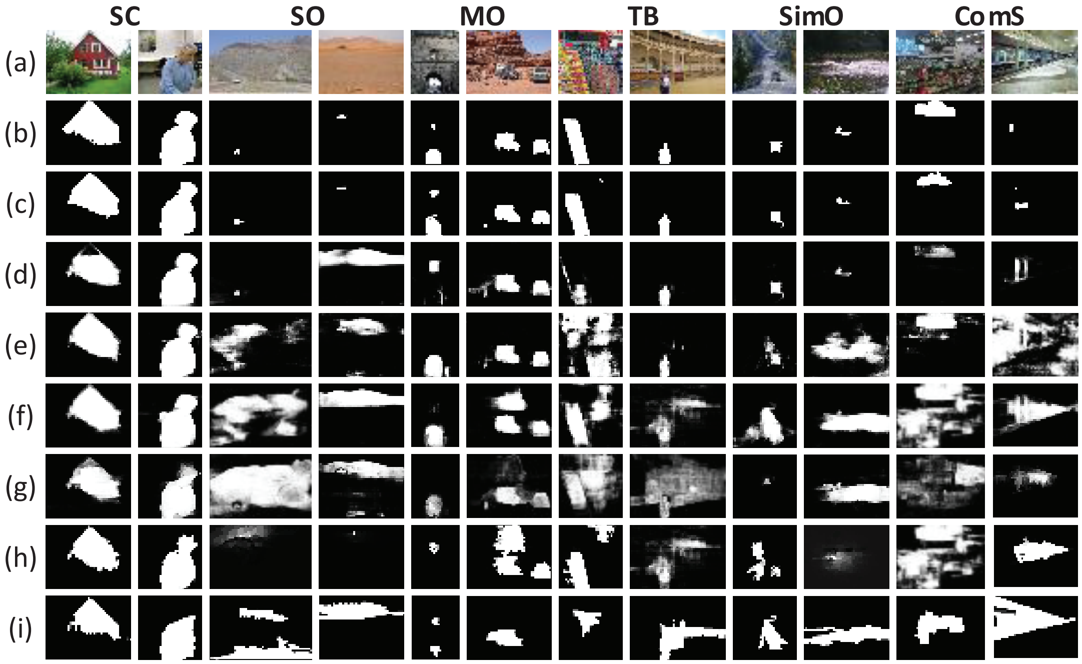


Figure 9. Visual comparisons of some good methods. (a) Image; (b) GT; (c) TSPOANet; (d) BMP [48]; (e) LFR [49]; (f) Amulet [50]; (g) UCF [51]; (h) DLS [17]; (i) ELE [42].

4.4. Performance comparison

In this section, we compare our method with 9 state-of-the-art methods, including BMP [48], LFR [49], AMC [47], Amulet [50], UCF [51], DLS [17], ELE [42], ELD [24], and MDF [25]. Visual and quantitative comparisons are both taken into account to make fair comparisons.

Quantitative Comparisons Fig. 8 shows PR curves of different methods. Table 1 lists the average F-measure values and MAE values of different methods. It is obvious from Fig. 8 that the proposed method achieves better PR curves than most of the compared state-of-the-art methods. Besides, it can be easily seen from Table 1 that our method performs best with respect to the F-measure metric. In terms of MAE metric, the proposed model again performs the best on PASCAL-S [28], DUTS [40], and DUT-OMRON [45], and is the second best on ECSSD [44] and HKU-IS [25]. These quantitative comparisons evidently verify the superiority of our proposed model.

Visual Comparisons Fig. 9 shows some visual comparisons of different methods in various cases, including Simple Case (SC), Small Object (SO), Multiple Objects (MO), Touching Boundary (TB), Similar between Object and backgrounds (SimO), and Complicated Scene (ComS). For the case of SC, most of the mentioned methods get good detection results in general. For the case of SO, most of the compared methods fail to detect the needle-like salient object and wrongly mark backgrounds as salient, while our method is able to accurately locate the needle-like object and well suppress the background. For the case of MO, our method can detect all the salient objects whereas the other methods mostly miss one object or introduce some background noise. For the case of TB, the state-of-the-art methods introduce a lot of background noise, while the proposed network can accurately stand out the whole salient object. For the case of SimO, the compared methods mostly label some backgrounds as salient, while our method is able to accurately distinguish the salient object from the confusing background. For the case of ComS, most state-of-the-art methods are unable to identify the salient object, as opposed to it, our method can still stand out the salient object from

Table 1. Average F-measure values and MAE values of different methods. Top three methods are marked by red, blue, and magenta, respectively. "-" means that the corresponding authors do not provide the detection results of the dataset.

	ECSSD [44]		**HKU-IS [25]**		**PASCAL-S [28]**		**DUTS [40]**		**DUT-OMRON [45]**	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
Ours	0.8873	0.0515	0.8795	0.0391	0.8253	0.0749	0.7993	0.0482	0.7030	0.0628
BMP [48]	0.8682	0.0447	0.8707	0.0389	0.7845	0.0753	0.7505	0.0490	0.6917	0.0635
LFR [49]	0.8799	0.0525	0.8752	0.0396	0.8059	0.1066	0.7064	0.0834	0.6656	0.1030
AMC [47]	0.6516	0.2090	0.7603	0.2160	0.7065	0.1946	0.6374	0.2489	0.5775	0.2693
Amulet [50]	0.8683	0.0589	0.8428	0.0501	0.7956	0.0997	0.6816	0.0846	0.6472	0.0976
UCF [51]	0.8439	0.0691	0.8235	0.0612	0.7675	0.1155	0.6351	0.1119	0.6206	0.1203
DLS [17]	0.8219	0.0860	0.8080	0.0696	0.7344	0.1301	-	-	0.6453	0.0895
ELE [42]	0.7545	0.1201	0.7053	0.1118	0.6705	0.1614	0.5786	0.1272	0.5752	0.1215
ELD [24]	0.8169	0.0790	-	-	0.7413	0.1211	-	-	0.6141	0.0910
MDF [25]	0.8068	0.1050	0.7844	0.1292	0.7113	0.1420	-	-	0.6443	0.0916

complicated backgrounds.

To sum up, compared with the state-of-the-arts, the proposed TSPOANet can accurately locate the salient object in various cases, and segment out the salient object with good wholeness and uniformity.

4.5. Ablation Analysis

TSPOANet To explore the effectiveness of TSPOANet, we compare the entire framework with a baseline, which is implemented by removing TSPOANet from the entire framework. Table 2 and Fig. 10 show the quantitative and visual comparisons, respectively. It can be easily seen from Table 2 that TSPOANet improves the performance to a clear margin. From the left two columns of Fig. 10, it is obvious that TSPOANet helps to grab much better uniformity and wholeness for the salient object. The improvements lie in the part-object relationships provided by TSPOANet.

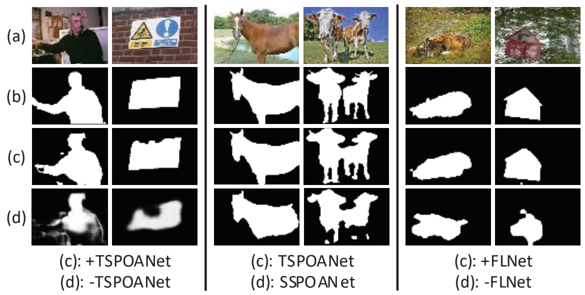


Figure 10. Visual comparisons for ablation analyses. (a) Image; (b) GT.

Two-stream strategy We explore the superiority of the two-stream strategy by comparing the proposed TSPOANet and a baseline, i.e., Single-Stream POANet (SSPOANet), which is implemented by directly adopting the original CapsNet following FLNet. As shown in Table 2, the proposed TSPOANet achieves better performance than SSPOANet. Besides, as illustrated in the middle two columns of Fig. 10, the proposed TSPOANet can detect the whole salient object, while SSPOANet misses some salient parts. The superiority of TSPOANet may be attributed to the two-stream strategy, which alleviates some noisy part-object assignments.

Table 2. Performance evaluations for the ablation analyses on ECSSD [44].

	F_β	MAE
+TSPOANet	**0.8816**	**0.0521**
-TSPOANet	0.8250	0.0694
TSPOANet	**0.8816**	**0.0521**
SSPOANet	0.8706	0.0644
+FLNet	**0.8706**	**0.0644**
-FLNet	0.6545	0.1504

FLNet To explore the validity of FLNet, we compare SSPOANet that learns features through FLNet with its modified version, which learns features of the input image through a Conv+ReLU layer used by the original CapsNet. It can be easily observed from Table. 2 that FLNet promotes the performance significantly. From the right two columns of Fig. 10, it is obvious that FLNet makes the framework possess the ability of identifying the salient object wholly, which is attributed to the rich features learned by FLNet.

5. Conclusions

In this paper, we have proposed a new salient property of part-object relationships provided by the CapsNet for salient object detection. To achieve this, we have presented a deep Two-Stream Part-Object Assignment Network (TSPOANet). The proposed model requires less computation budgets while obtaining better wholeness and uniformity of the segmented salient object.

Acknowledgements This work is supported by the National Natural Science Foundation of China under Grant No. 61773301 and the Funds of China Scholarship Council under Grant No. 201806960044.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *Operating System Design and Implementation*, pages 265–283, 2016.

[2] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, 2009.

[3] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, 2015.

[4] X. Cao, Z. Tao, B. Zhang, H. Fu, and W. Feng. Self-adaptively weighted co-saliency detection via rank constraint. *IEEE Transactions on Image Processing*, 23(9):4175–4186, 2014.

[5] M.-M. Cheng, Q.-B. Hou, S.-H. Zhang, and P. L. Rosin. Intelligent visual media processing: When graphics meets vision. *Journal of Computer Science and Technology*, 32(1):110–121, 2017.

[6] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015.

[7] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. M. Hu. Global contrast based salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 409–416, 2011.

[8] R. Cong, J. Lei, H. Fu, Q. Huang, X. Cao, and C. Hou. Co-saliency detection for rgbd images based on multi-constraint feature matching and cross label propagation. *IEEE Transactions on Image Processing*, 27(2):568–579, 2018.

[9] H. Fu, X. Cao, and Z. Tu. Cluster-based co-saliency detection. *IEEE Transactions on Image Processing*, 22(10):3766–3778, 2013.

[10] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai. 3-d object retrieval and recognition with hypergraph analysis. *IEEE Transactions on Image Processing*, 21(9):4290–4303, 2012.

[11] C. Guo and L. Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Trans. Image Processing*, 19(1):185–198, 2010.

[12] J. Guo, T. Ren, L. Huang, X. Liu, M.-M. Cheng, and G. Wu. Video salient object detection via cross-frame cellular automata. In *IEEE International Conference on Multimedia and Expo*, pages 325–330, 2017.

[13] J. Han, K. N. Ngan, M. Li, and H.-J. Zhang. Unsupervised extraction of visual attention objects in color images. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(1):141–145, 2006.

[14] J. Han, E. J. Pauwels, and P. De Zeeuw. Fast saliency-aware multi-modality image fusion. *Neurocomputing*, 111:70–80, 2013.

[15] G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51, 2011.

[16] G. E. Hinton, S. Sabour, and N. Frosst. Matrix capsules with em routing. In *International Conference on Learning Representations*, pages 3856–3866, 2018.

[17] P. Hu, B. Shuai, J. Liu, and G. Wang. Deep level sets for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2300–2309, 2017.

[18] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10):1304–1318, 2004.

[19] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[20] J. Kim, D. Han, Y.-W. Tai, and J. Kim. Salient region detection via high-dimensional color transform and local spatial support. *IEEE Transactions on Image Processing*, 25(1):9–23, 2016.

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] D. A. Klein and S. Frintrop. Center-surround divergence of feature statistics for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2214–2219, 2011.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[24] G. Lee, Y.-W. Tai, and J. Kim. Eld-net: An efficient deep learning architecture for accurate saliency detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1599–1610, 2018.

[25] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5455–5463, 2015.

[26] N. Li, B. Sun, and J. Yu. A weighted sparse coding framework for saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5216–5223, 2015.

[27] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang. Saliency detection via dense and sparse reconstruction. In *Proceedings of the IEEE international conference on computer vision*, pages 2976–2983, 2013.

[28] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014.

[29] N. Liu and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 678–686, 2016.

[30] N. Liu, J. Han, and M.-H. Yang. Picanet: Learning pixelwise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3089–3098, 2018.

[31] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE*

Transactions on Pattern analysis and machine intelligence, 33(2):353–367, 2011.

[32] H. Lu, X. Li, L. Zhang, X. Ruan, and M.-H. Yang. Dense and sparse reconstruction error based saliency descriptor. *IEEE Transactions on Image Processing*, 25(4):1592–1603, 2016.

[33] Z. Luo, A. K. Mishra, A. Achkar, J. A. Eichel, S. Li, and P.-M. Jodoin. Non-local deep features for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6609–6617, 2017.

[34] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, B. Schiele, et al. Exploiting saliency for object segmentation from image level labels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[35] H. Peng, B. Li, H. Ling, W. Hu, W. Xiong, and S. J. Maybank. Salient object detection via structured matrix decomposition. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 39(4):818–832, 2017.

[36] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition? In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II, 2004.

[37] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representation*, 2015.

[39] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien. Real-time salient object detection with a minimum spanning tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2334–2342, 2016.

[40] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan. Learning to detect salient objects with image-level supervision. In *Proc. IEEE Conference Computer Vision and Pattern Recognition*, pages 136–145, 2017.

[41] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2314–2320, 2017.

[42] C. Xia, J. Li, X. Chen, A. Zheng, and Y. Zhang. What is and what is not a salient object? learning salient object detector by ensembling linear exemplar regressors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4321–4329, 2017.

[43] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[44] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1155–1162, 2013.

[45] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3166–3173, 2013.

[46] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[47] L. Zhang, J. Ai, B. Jiang, H. Lu, and X. Li. Saliency detection via absorbing markov chain with learnt transition probability. *IEEE Transactions on Image Processing*, 27(2):987–998, 2018.

[48] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang. A bi-directional message passing model for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1741–1750, 2018.

[49] P. Zhang, W. Liu, H. Lu, and C. Shen. Salient object detection by lossless feature reflection. In *International Joint Conference on Artificial Intelligence*, 2018.

[50] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 202–211, 2017.

[51] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin. Learning uncertain convolutional features for accurate saliency detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 212–221, 2017.

[52] T. Zhang, G.-J. Qi, B. Xiao, and J. Wang. Interleaved group convolutions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4373–4382, 2017.

[53] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1265–1274, 2015.