Integrating Action-aware Features for Saliency Prediction via Weakly Supervised Learning

Jiaqi Feng*†, Shuai Li*, Yunfeng Sui†, Lingtong Meng* and Ce Zhu*
* University of Electronic Science and Technology of China, Chengdu, China
E-mail: fengjiaqi94@163.com,shuaili@uestc.edu.cn,coolmatt1024@outlook.com,eczhu@uestc.edu.cn Tel: +86-15528121731
† Research Center of Second Research Insititute of CAAC, Chengdu, China
E-mail: suiyunfeng@caacsri.com

***Abstract*—Deep learning has been widely studied for saliency prediction. Despite the great performance improvement introduced by deep saliency models, some high-level concepts that contribute to the saliency prediction, such as text, objects of gaze and action, locations of motion, and expected locations of people, have not been explicitly considered. This paper investigates the objects of action and motion, and proposes to use action-aware features to compensate deep saliency models. The action-aware features are generated via weakly supervised learning using an extra action classification network trained with existing image based action datasets. Then a feature fusion module is developed to integrate the action-aware features for saliency prediction. Experiments show that the proposed saliency model with the action-aware features achieves better performance on three public benchmark datasets. More experiments are further conducted to analyze the effectiveness of the action-aware features in saliency prediction. To the best of our knowledge, this study is the first attempt on explicitly integrating objects of action and motion concept into deep saliency models.**

I. Introduction

Saliency prediction has been widely studied and used in many computer vision applications, such as object recognition [1], tracking [2], and image segmentation [3]. In the early phase of saliency prediction, models are developed based on bottom-up stimulus such as contrast of color, orientation and intensity [4]. Later, more high-level factors such as face and object detectors [5] are considered in saliency prediction since low-level factors alone cannot predict image areas of richer contextual information well. Nowadays, with the development of Deep Neural Networks (DNNs), saliency models with Deep Neural Networks (DNNs) [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] have also been studied. While the current DNNs based saliency models greatly improved the performance over the traditional methods, there are still many limitations mostly due to the underexploited high-level concepts such as text, objects of gaze and action, locations of motion, and expected locations of people in images, as analyzed in [16].

This paper focuses on exploiting the concept objects of action and motion in the saliency model. It concerns salient objects interacted with by person or salient regions containing possible action.

This work was supported in part by the Key Project of Sichuan Provincial Department of Science and Technology under Grant 2018JY0035, in part by the National Natural Science Foundation of China under Grant 61571102 and U1633128.

Many works [17], [18] show that channels of high-layer features in neural networks are informative of some high-level semantics and each channel actually represents a kind of feature. These give us insight into providing action-aware features for saliency models. With weakly supervised learning, action-aware features are extracted from an action classification network without collecting extra annotations, and then used for saliency prediction. The contributions of this paper can be summarized as follows.

- To the best of our knowledge, this is the first work attempting to explicitly integrate objects of action and motion concept into deep saliency models.
- A fusion module is developed to combine action-aware features and encoded contextual features from the base saliency network for the final saliency prediction.

The experimental results show that our proposed model with action-aware features improves performance over the existing methods. The rest of the paper is organized as follows. Section 2 briefly reviews related work. The detailed configuration of the proposed method is described in Section 3. Section 4 illustrates implementation details and experimental results on three benchmark datasets.

II. Related Works

DNNs have shown a remarkable performance in saliency detection. The eDN [6] model was an early model learning the representations from neural networks for saliency detection, where only a shallow three-layer network was used. In Deep Gaze I [7], the model was built on AlexNet [19] and some layers of AlexNet were linearly combined for final prediction. Deep Gaze II [8] built its model on the deeper VGG19 [20] and a readout network was followed to learn to combine layers from VGG19 to predict saliency maps. In [9], ML-Net was proposed and multiple VGG layers were directly combined to predict saliency map. The output of ML-Net was centre biased by a learned centre prior. In SALICON [10], two input images with coarse and fine resolutions are first fed to the neural network, then their encoded features are combined for prediction. In DeepFix [11], inception module and global context are explored in the saliency models. In [12], [13], LSTM is adopted to refine final feature maps for saliency prediction.

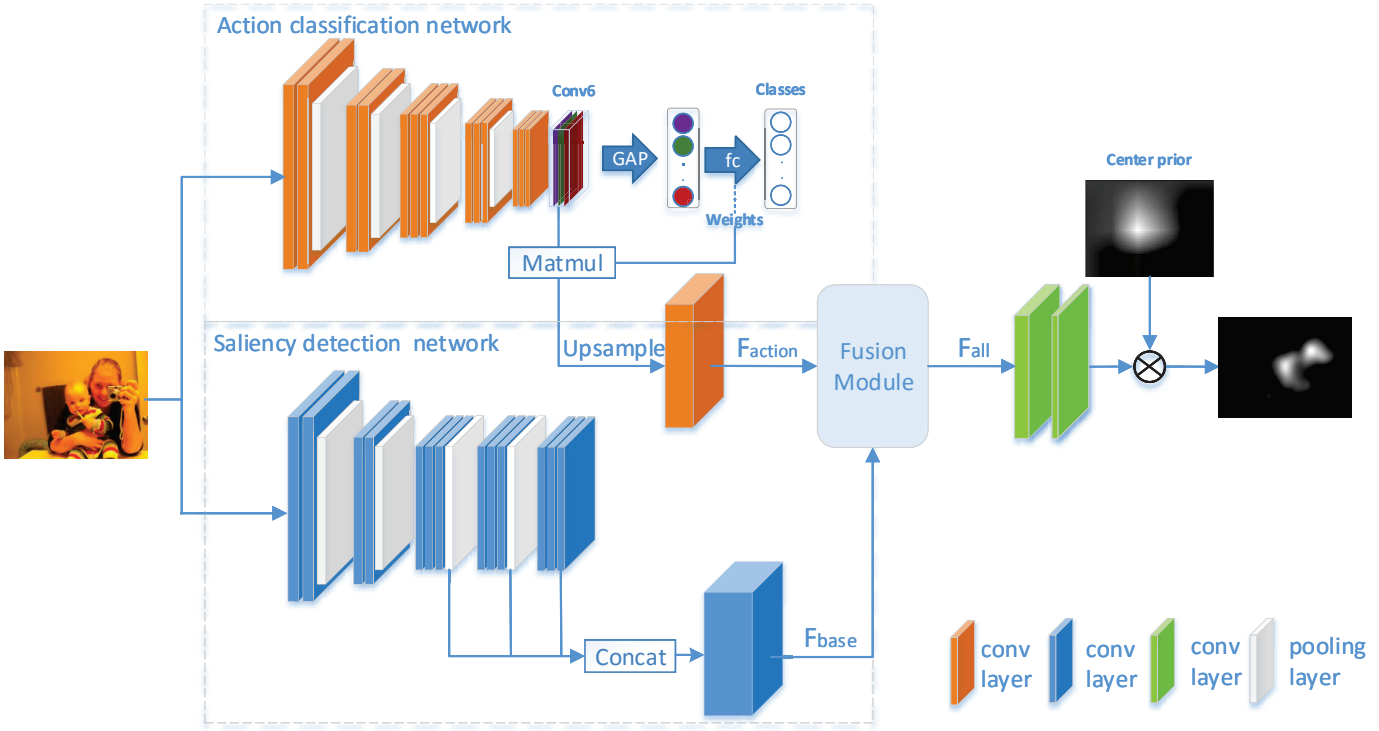

Fig. 1. The architecture of the proposed model. The upper input branch is for generating action-aware features and the lower input branch is for generating contextual features for saliency detection. Then features are fused to predict the saliency map.

On the other hand, instead of improving the architectures of neural networks to obtain more effective features for saliency prediction, the saliency map is formulated as a generalized Bernoulli distribution in [21] and a novel loss functions developed based on the distribution is introduced to train deep saliency models. In [22], adversarial training is introduced to train saliency models to obtain better performance.

While these models improve the performance, the high-level concepts including objects of action and motion have not been explicitly considered. The most closely related work to ours is the work of [23] focusing on the objects of gaze, where a dataset containing the head and gaze information is used to augment the saliency map. By contrast, our work proposes to explicitly integrate objects of action and motion concept into deep saliency models via weakly supervised learning without collecting any dataset with extra annotations.

III. Proposed Method

The architecture of the proposed model is illustrated in Fig.1. It consists of three parts: an action classification network for generating action-aware features, a base deep saliency model for generating contextual features, and a feature fusion module to combine the action-aware features and contextual features for the final saliency prediction. The details of each component are explained in the following.

A. Action-aware Features

It has been shown [24], [25] that the trained CNNs for classification can be informative of the image areas about the labels used for training. In CAM [24], class activation maps from classification network are used for visualization and localization. Therefore, to generate the action-aware features, an action classification network is used. Similarly as the CAM [24], it adopts a global average pooling (GAP) based classification network with the simple VGG16 model as the backbone. Differently, it treats class activation maps as features. As shown in the upper part of Fig.1, the five convolution blocks of VGG16 are used while the pool5 layer and the rest layers are removed. Instead, an extra convolutional layer, a GAP layer, and a fully connected layer with softmax function for the final classification are added. This network can be trained with any existing image based action dataset. In this paper, the Standford 40 [26] with 40 action classes is used. After the network is trained, it can be fixed and used as a feature extractor to generate the action-aware features.

Since for action classification, only the image-level labels are provided without annotations such as bounding box indicating detailed location information of the action, the action-aware features are obtained based on the class activation map similarly as CAM [24]. The action-aware feature of one action class can be obtained by a weighted sum of features before the GAP layer:

$$F_{action}(c) = \sum_{k=1}^{N} W_{k,c} Conv6_k \quad (1)$$

where k denotes the channel index of conv6 layer, c denotes the action class, N denotes the channels of conv6 layer, W

Fig. 2. Visualization of action-aware features. Action-aware feature maps are shown as heat maps where red means high values in action-aware features and blue means low values.

denotes the weights of the final fully connected layer, which encodes relative importance of channels of conv6 layer. In experiments, the kernel size of conv6 layer is set to 3×3 and N is set to 1024. The action-aware features of an image is able to highlight objects of action and motion of corresponding action class, as shown in Fig.2.

The final action-aware feature for the saliency prediction is obtained by concatenating the features of all action classes to make the features response to all the action classes.

Considering that for different features of each class, the classification probability of its action class is usually different, the feature for each class is further enhanced by multiplying the class probability of its corresponding action class as follows.

$$F_{action}(c) = p(c) \times \sum_{k=1}^{N} W_{k,c} Conv6_k \tag{2}$$

where p denotes corresponding softmax class probability.

B. Contextual Features

In addition to the action-aware features proposed above, the image features used in the conventional deep saliency models are also used for saliency prediction. The ML-Net [9] is adopted as a base deep saliency model for its simplicity. In the model, the feature maps of the last three blocks of VGG16 are first concatenated to generate contextual features for saliency detection, then two additional convolutional layers are added to predict the saliency map. In order to produce features from three blocks with the same resolution, the stride of pool4 layer in VGG16 is set to one and the pool5 layer in block5 is removed.

Context information has been shown [12] to play an important role in saliency prediction. However, in ML-Net, after directly setting stride of pool4 layer to 1, the receptive field of block5 layer is decreased and consequently, the context information contained in block5 layer is reduced. In order to increase the receptive field, the atrous convolutional layers [27] with atrous rate 2 is proposed to be used in the block5 layers.

C. Feature Fusion

With the contextual features and action-aware features extracted from the corresponding networks shown above, a fusion module is further proposed to combine them for the final saliency prediction. Since the contextual features of different layers are usually of different scales and action-aware features obtained from a different sub-network is also of different scales, features are normalized with batch normalization (BN) [28].

The simple concatenation and element-wise addition are used as the fusion operations. For the action-aware feature, an extra convolutional layer is added to improve its robustness. In experiments, we found a 5×5 convolutional layer works well and is added on the action-aware features, then all features are concatenated in channels after batch normalization as follows.

For the concatenation based feature fusion, all features are concatenated in channels as follows.

$$F_{all} = BN(Concat(F_{base}, Conv(F_{action}))) \tag{3}$$

where $F_{base} \in \mathbb{R}^{h*w*c_1}$ and $F_{action} \in \mathbb{R}^{h*w*c_2}$ represent the contextual features and the action-aware features, respectively. Specially, BN(F) means BN layer is added respectively on multi-layer features of F if F is composed of multi-layer features.

For the element-wise addition based feature fusion, action-aware features are followed by a one-channel convolutional layer, then an action-aware mask is obtained and is added on contextual features as follows.

$$F_{all} = BN(F_{base} \oplus Conv(F_{action})) \tag{4}$$

IV. Experiments

A. Datasets

In our experiments, four datasets are used including one dataset used for training the action classification network and three used for training and testing the proposed deep saliency model.

Stanford 40 [26]: The Stanford 40 Action dataset contains 40 classes of human daily actions, such as brushing teeth, reading book, walking the dog, etc. There are 9532 images in dataset, where 6800 images with 170 images per action class are used for training and the rest are used for testing. This dataset is used for training the action classification network to generate the action-aware features.

SALICON [29]: Saliency in Context (SALICON) is the biggest saliency dataset so far and is commonly used for training saliency models. It contains 10000 images for training and 5000 images for validation.

MIT1003 [30]: It is an eye-tracking dataset containing 1003 images obtained with 15 viewers. There are 779 landscape images and 228 portrait images of different resolutions. 900 images are chosen randomly in MIT1003 to fine-tune the model and extra 103 images are used for evaluation as in [11].

CAT2000 [31]: CAT2000 dataset is a collection of 4000 images of 20 different categories covering different types of

scenes such as Cartoons, Art, Action, etc. The training set contains 2000 images with 100 images per class. To better illustrate the effect of the action-aware features in saliency prediction, images in Action category are used to evaluate the performance of our models.

B. Evaluation Metrics

There are several saliency metrics used in the literature. AUC-Judd proposed in [30] is a commonly used traditional metric. In addition, Normalized Scanpath Saliency(NSS) and Pearsons Correlation Coefficient(CC) metrics are recommeded by [32]. In this work, we choose NSS, CC and AUC-Judd to evaluate our models.

C. Training Details

Tensorflow and Keras were used as the experiment platform. The training process is performed in two steps. Firstly, the action classification network of our model is trained on Stanford 40 dataset. The VGG layers are initialized by pre-trained parameters and the following layers are initialized randomly. The weights in the VGG layers are first fixed to train the following layers with learning rate of 1×10^{-4} and weight decay of 5×10^{-4}, then the whole network is fine-tuned with a dropout layer of a dropping probability 0.5 inserted after GAP. Exponential decay is applied on learning rate with decay rate 0.95 and decay step 100. The cross entropy loss is used and the model is trained using Stochastic Gradient Descent (SGD) with a momentum of 0.9 and batch size 64. Input images are rescaled to the resolution 240×320. Secondly, with the action classification network fixed and used as a feature extractor, the rest of our saliency model is trained on the SALICON training dataset. A similar training process as ML-Net [9] is used, but the input image size is rescaled to 240×320 due to limited GPU memory. The learning rate is decayed using the inverse time decay with steps 50 and rate 5×10^{-4} . Other training details can be found in [9]. When evaluating on MIT1003, the saliency model is further finetuned on MIT1003 training set as in [11] using the same training process. When evaluating on CAT2000 dataset (Action category), 1000 images with 50 images per class are used to fine-tune the model and extra 50 images in Action category are used to evaluate the performance of our models.

D. Experimental Results

Evaluations are conducted on SALICON validation dataset, MIT1003 validation dataset and CAT2000 dataset (action category).

Quantitative results

The proposed model is evaluated with extensive comparison to the ML-Net, where three variants are designed for ablation study, as shown in Table 1.

The quantitative results on SALICON and MIT1003 validation datasets are presented in Table 2. The results between model A and model B reflect that context information plays an important role in saliency prediction. The results between our proposed models and model B show that action-aware features are useful in both concatenation and element-wise addition form, the proposed method with explicitly integrating objects of action and motion concept achieves better performance. Moreover, it is worth noting that improvements on AUC-Judd are relatively smaller which also agrees with the argument that AUC-Judd has begun to saturate on saliency dataset as mentioned in [32].

TABLE I
DESIGNED MODEL VARIANTS BASED ON ML-NET

Model Variants	Model description
A	raw ML-Net
B	ML-Net with atrous convolution layers
Proposed I	Our final model, features are fused in concatenation form as equation (3)
Proposed II	Our final model, features are fused in element-wise addition form as equation (4)

To further demonstrate the effectiveness of action-aware features, we conduct experiments on CAT2000 dataset (action category) with three saliency models: ML-Net, SALICON and DeepFix. We reimplement SALICON and DeepFix models following their papers as the base saliency models except that center prior is removed for simplicity. Action-aware features are fused in the same forms except that BN is not used since only one-layer contextual feature is used in SALICON and DeepFix. In Table 3, validation results on CAT2000 dataset (action category) are presented. The bold numbers also denote model gets better results with action-aware features. With action-aware features integrated, both in concatenation and element-wise addition form, better overall performance on the dataset is obtained on three different saliency models. It seems concatenation form performs more stable.

However, the improvements on SALICON and DeepFix are relatively smaller than ML-Net, which is mostly because of two points: these models learn more powerful contextual information that action-aware features compensate little for them, and difference exists in terms of the training data since action-aware features are trained from an action dataset instead of saliency dataset. Consequently, more powerful action classification network and providing image labels about action classes for the saliency dataset to support multitask learning are promising next steps.

Although integrating action-aware features into saliency model intuitively makes sense, we attempt to explain how action-aware features help in predicting saliency maps. High-layer features capture more semantics and are closely related to ground truth data which is saliency map in our work. Therefore, we design a method to analyze the effect of action-aware features by the CC between last-layer convolutional features (F) and ground-truth saliency maps (layer CC) as follows.

Firstly, for every input image, calculate CC between each channel of F and corresponding ground-truth saliency map. Secondly, input all test images and get mean CC of each channel (channel CC). Finally, get layer CC by averaging all

TABLE II
QUANTITATIVE RESULTS ON SALICON AND MIT1003 VALIDATION DATASET. THE BOLD NUMBERS DENOTE PROPOSED METHOD GETS BETTER RESULT ON THE METRIC.

Model	Dataset					
	SALICON			MIT1003		
	Auc-Judd	NSS	CC	AUC-Judd	NSS	CC
A	0.8543	1.7807	0.8270	0.8638	2.3114	0.5932
B	0.8605	1.8257	0.8543	0.8505	2.3898	0.6145
Proposed I	0.8593	**1.8487**	**0.8624**	**0.8668**	**2.4351**	**0.6411**
Proposed II	**0.8606**	**1.8542**	**0.8638**	**0.8792**	**2.4428**	**0.6468**

TABLE III
QUANTITATIVE RESULTS ON CAT2000 DATASET (ACTION CATEGORY). W1 DENOTES ACTION-AWARE FEATURES ARE COMBINED AS EQUATION (3), W2 DENOTES ACTION-AWARE FEATURES ARE COMBINED AS EQUATION (4), W/O DENOTES WITHOUT ACTION-AWARE FEATURES.

Base model	action-aware features	metrics		
		Auc-Judd	NSS	CC
ML-Net [9]	w/o	0.8460	2.0989	0.7293
	w1	**0.8600**	**2.1038**	**0.7422**
	w2	**0.8619**	**2.1056**	**0.7452**
SALICON [10]	w/o	0.8469	2.1218	0.7602
	w1	**0.8476**	**2.1301**	**0.7632**
	w2	0.8446	**2.1481**	**0.7689**
DeepFix [11]	w/o	0.8430	2.3056	0.8113
	w1	**0.8500**	**2.3179**	**0.8162**
	w2	**0.8506**	2.2894	**0.8151**

channel CC of F.

The layer CC reflects the correlation coefficient between the last-layer convolutional feature and ground-truth saliency maps. In Table 4, with or without action-aware features, layer CC in ML-Net on two test datasets are presented. It can be seen that layer CC with action-aware features is larger, which corresponds to better results in Table 2 and Table 3. In other words, the action-aware features indeed help provide better features for saliency prediction.

Qualitative results

Visualization results of ML-Net on some test images are shown in Fig.3. It can be seen that the objects of action and motion are more highlighted with action-aware features integrated.

V. CONCLUSION

In this work, we explicitly integrate objects of action and motion concept into deep saliency models. Without collecting extra annotations, we propose to extract action-aware features from an action classification network by weakly supervised learning and develop a fusion module to combine action-aware features and contextual features from the base deep saliency model for the final saliency prediction. Both in concatenation form and element-wise addition form, the proposed action-aware features improves saliency prediction. Our quantitative and qualitative results demonstrate that explicitly integrating objects of action and motion concept improves performance of deep saliency models.

TABLE IV
EXPERIMENTAL RESULTS EXPLAINING HOW ACTION-AWARE FEATURES HELP IN PREDICTING SALIENCY MAPS. W1 DENOTES ACTION-AWARE FEATURES ARE COMBINED AS EQUATION (3), W2 DENOTES ACTION-AWARE FEATURES ARE COMBINED AS EQUATION (4), W/O DENOTES WITHOUT ACTION-AWARE FEATURES.

Dataset	action-aware features	layer CC
SALICON	w/o	0.297
	w1	**0.430**
	w2	**0.434**
CAT2000(action category)	w/o	0.242
	w1	**0.304**
	w2	**0.391**

Fig. 3. Row 1 are raw images. Row 2 are ground-truth saliency maps. Row 3 are outputs without action-aware features. Row 4 are outputs with action-aware features.

REFERENCES

[1] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 989–1005, 2009.

[2] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International conference on machine learning*, 2015, pp. 597–606.

[3] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele, "Exploiting saliency for object segmentation from image level labels," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5038–5047.

[4] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 11, pp. 1254–1259, 1998.

[5] A. Borji, "Boosting bottom-up and top-down visual features for saliency estimation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 438–445.

[6] E. Vig, M. Dorr, and D. Cox, "Large-scale optimization of hierarchical features for saliency prediction in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2798–2805.

[7] M. Kümmerer, L. Theis, and M. Bethge, "Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet," *arXiv preprint arXiv:1411.1045*, 2014.

[8] M. Kümmerer, T. S. Wallis, and M. Bethge, "Deepgaze ii: Reading fixations from deep features trained on object recognition," *arXiv preprint arXiv:1610.01563*, 2016.

[9] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "A deep multi-level network for saliency prediction," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 3488–3493.

[10] X. Huang, C. Shen, X. Boix, and Q. Zhao, "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 262–270.

[11] S. S. Kruthiventi, K. Ayush, and R. V. Babu, "Deepfix: A fully convolutional neural network for predicting human eye fixations," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4446–4456, 2017.

[12] N. Liu and J. Han, "A deep spatial contextual long-term recurrent convolutional network for saliency detection," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3264–3274, 2018.

[13] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an lstm-based saliency attentive model," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5142–5154, 2018.

[14] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (indrnn): Building a longer and deeper rnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5457–5466.

[15] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "A fully trainable network with rnn-based pooling," *Neurocomputing*, 2019.

[16] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, and F. Durand, "Where should saliency models look next?" in *European Conference on Computer Vision*. Springer, 2016, pp. 809–824.

[17] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[18] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.

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

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] S. Jetley, N. Murray, and E. Vig, "End-to-end saliency mapping via probability distribution prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5753–5761.

[22] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," *arXiv preprint arXiv:1701.01081*, 2017.

[23] D. Parks, A. Borji, and L. Itti, "Augmented saliency model using automatic 3d head pose detection and learned gaze following in natural scenes," *Vision research*, vol. 116, pp. 113–126, 2015.

[24] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[26] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1331–1338.

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

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[29] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1072–1080.

[30] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 2106–2113.

[31] A. Borji and L. Itti, "Cat2000: A large scale fixation dataset for boosting saliency research," *arXiv preprint arXiv:1505.03581*, 2015.

[32] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 3, pp. 740–757, 2019.