Convolutional Networks based Image Saliency Detection

Parth Goyal
2017csb1095@iitrpr.ac.in
Kamal Sharma
2017csb1084@iitrpr.ac.in

Indian Institute of Technology
Ropar, 140001
Punjab, India

Abstract

To predict salient features in an image, high-level visual features at multiple spatial scales must be extracted and augmented with relevant information. In this report, we will explain an approach based on CNN (Convolution Neural Network) that is largely molded for image classification tasks. The implementation of the model explained in the report estimates the fixation of the human eye across complex natural scenes. The architecture forms an encoder-decoder [7] structure consisting of a module with different convolution layers taken at different dilation rates to capture multiscale features in parallel. We combine the results with global scene information in order to predict accurately. Determining characteristic or salient regions of images allows transitioning from low-level pixels to more meaningful high-level regions.

- **Input** : Image
- **Output** : Visual Salient Image

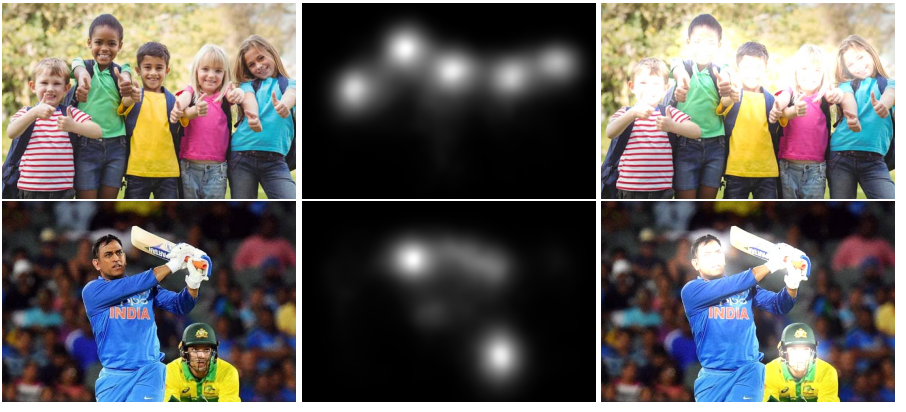
More human attention –> More visual saliency

1 Introduction

Human eye has a great ability to extract the relevant information from a complex scene [5]. Saliency in an image enables eye-brain connection to quickly focus on the most important areas in the image. Saliency detection in an image means to extract the prominently noticeable parts of the image. Saliency detection in Computer Vision is applied to a lot of applications. Some of these are - Object detection, Designig the logo, pop-up and stand-out for a quick glance, Robots with visual systems.

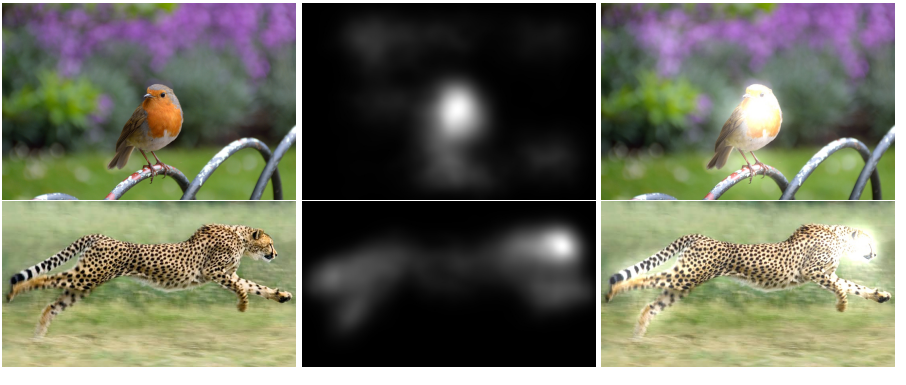
Koch and Ullman [6] have introduced the notion of a central saliency map which combines low-level information to serve as the basis of eye movements. It is likely that the complex representations of multiple spatial scales are necessary for predicting human eye fixation pattern accurately. Hence we use Convolution Neural Networks to extract the salient features from natural images and connect them to a distribution of saliency across scenes. The network is based on a lightweight image classification backbone and hence presents a suitable choice for applications with limited computational resources to estimate human fixations across complex natural scenes.

With the advent of deep neural network solutions for visual tasks such as image classification, saliency modelling has also undergone a paradigm shift from manual feature engineering towards automatic representation learning. In this work, we leveraged the capability of convolutional neural networks (CNNs) to extract relevant features from raw images and decode them towards a distribution of saliency across arbitrary scenes. Compared to the early approach by Itti et al. [4], this approach allows predictions to be based on semantic information instead of low-level feature contrasts.



(a) Input Image (b) Predicted Saliency Map (c) Output Image

Figure 1: Demonstration of the CNN model to predict saliency in images with People. People faces are most salient features of the image (a) shows the Input Images given to the Model. (b) shows the Predicted Saliency Map for each image. (c) shows the Final Output after overlaying Saliency Map on the Input Image



Input Image Predicted Saliency Map Output Images

Figure 2: Demonstration of the CNN model to predict saliency in images with Animals. Here too, animal faces are most salient features of the image

Furthermore, it is likely that complex representations at multiple spatial scales are necessary for accurate predictions of human fixation patterns. We therefore incorporated a contextual module that samples multi-scale information and augments it with global scene features.

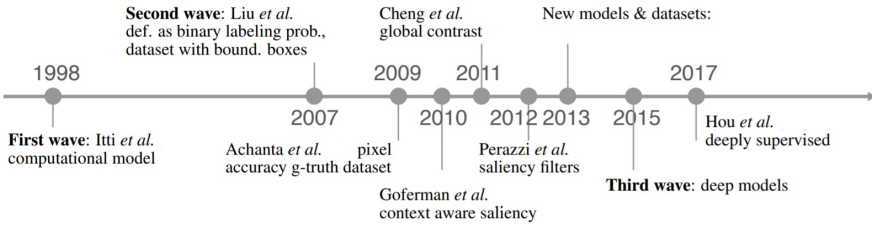


Figure 3: Timeline of Work performed by researchers on Visual-Saliency-Detection

2 Literature Review

In the early approaches to determine visual saliency and attention, computational models were defined in terms of different theoretical frameworks, including Bayesian and graph-based formulations. Work by Itti el al. [4] was inspired more by biological than mathematical principles, there work captured center-surround differences at multiple spatial scales to three basic feature channels: color, intensity, and orientation. Their model successfully captured semantic image content, such as faces and texts. The notion of central saliency map was introduced by Koch and Ullman [6] that combined low-level information for eye movements.

With the large-scale acquisition of eye tracking measurements under natural viewing conditions, data-driven machine learning techniques became more practicable. Early approaches used SVMs, and slowly as development progressed, Deep Neural Networks were used with emergent representations for the estimation of human fixation patterns.

3 Method

The CNN model consists of modules adapted from the semantic segmenation literature to predict fixation density maps of the same resolution of the image. Functionally, it tries to replicate human behaviour under free-viewing conditions to get salient features in the image.

3.1 Architecture [7]

Spatial Features needs to be preserved when image-to-image learning problems are being considered. As a consequence, the network does not include any fully-connected layers and reduces the number of downsampling operations inherent to classification models. The popular VGG16 architecture [8] is used as an image encoder by reusing the pre-trained convolutional layers to extract increasingly complex features along its hierarchy. See Figure (3).

- The encoder of the model consists of a pretrained VGG16 architecture with 13 convolutional layers. All dense layers are discarded and the last 3 layers are dilated at a rate of 2 to account for the omitted downsampling. Finally, the activations from 3 layers are combined.

- The ASPP [1] samples information at multiple spatial scales in parallel via convolutional layers with different dilation factors.

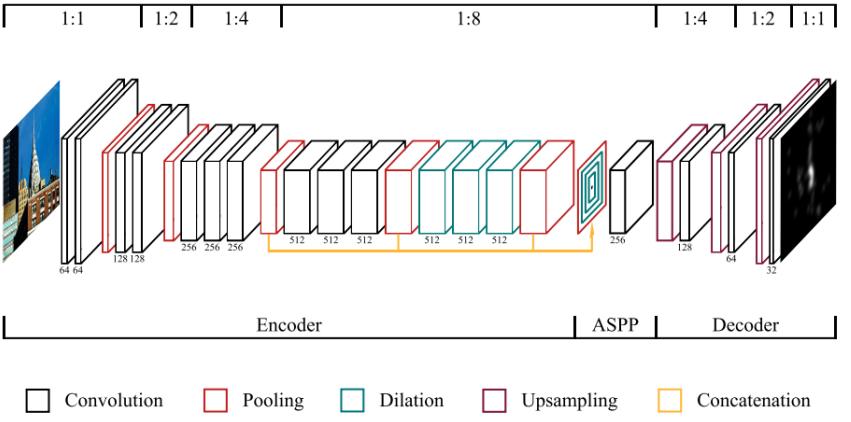


Figure 4: An illustration of the modules that constitute our encoder-decoder architecture. The VGG16 backbone was modified to account for the requirements of dense prediction tasks by omitting feature downsampling in the last two max-pooling layers. Multi-level activations were then forwarded to the ASPP module, which captured information at different spatial scales in parallel. Finally, the input image dimensions were restored via the decoder network. Subscripts beneath convolutional layers denote the corresponding number of feature maps

- The decoder applies a series of 3 upsampling blocks that each performs bilinear upsampling followed by a 3x3 convolution to avoid checkerboard artifacts in the image space.
- *Unlike all other layers, the output of the model is not modified by a ReLU.

Algorithm 1 forward_Model_Architecture(input_image)

encoded_ouput ← *Pass the image* (*input_image*) *through the encoder*
aspp_output ← *Pass the encoded image* (*encoded_output*) *through the aspp*
decoded_output ← *Pass the aspp_output throught the decoder*
final_output ← *normalize the decoded_output*
return final_output

3.2 Training

Weight values from the ASPP module and decoder were initialized according to the Xavier method by Glorot and Bengio [2]. We normalized the model output such that all values are non-negative with unit sum. To determine the difference between an estimated and a target distribution, the Kullback-Leibler (KL) divergence is an appropriate measure rooted in information theory to quantify the statistical distance D. This can be defined as follows:

$$D_{KL}(P \,||\, Q) = \sum_i Q_i \, ln(\varepsilon + \frac{Q_i}{\varepsilon + P_i}) \tag{1}$$

Here, Q represents the target distribution, P its approximation, i each pixel index, and ε a regularization constant. Equation (1) served as the loss function which was gradually

	Batch_size	Dilation rates (in aspp steps)	Architecture	Downsampling
Study 1	16	1,4,16	VGG16	avgpooling
Study 2	16	1,4,8,12	VGG16	maxpooling
Study 3	1	1,4,8,12	VGG16	maxpooling
Study 4	1	1,4,16	VGG16	maxpooling
Study 5	1	1,4,16	VGG19	maxpooling

Figure 5: Details of the 5 studies/experiements that we performed. We train the model on RTX 2080 (8 GB GPU). VGG16 model of batchsize 16 runs in approx. 1hr and VGG16 model of batchsize 1 runs in approx. 1hr 40 mins. *On GTX 1050 (2 GB GPU) VGG16 batchsize 1 model shows expected training time of approx 20 hrs.

minimized via the Adam optimization algorithm. We defined an upper learning rate of 10^{-6} and modified the weights in an online fashion due to a general inefficiency of batch training according to Wilson and Martinez [9]. Based on this general setup, we trained our network for 10 epochs and used the best-performing checkpoint for inference.

3.3 Dataset

Having a well defined annotated data is a prerequisite for the successfull application of deep learning techniques. New data collection methodologies have emerged that leverage webcam-based eye movements or mouse movements instead via crowdsourcing platforms. This approach resulted in the ***SALICON*** *Saliency Prediction Challenge(LSUN'17)* dataset, which consists of 10,000 training and 5,000 validation instances serving as a proxy for empirical gaze measurements. We rescaled and padded all images from the SALICON dataset to 240 * 320 pixels.

4 Experiments and Results

Doing a number of experiments on the dataset is essential for obtaining a good result. Figure (5) shows the results of all the experiments we did with their details. Figure (6) shows the trend of validation loss and training loss for all the 10 epochs and Figure (7) shows plots of those trends. Figure (8) shows you the comparison between the validation losses for the best models that we can obtain through our studies and experiements.

4.1 VGG19 vs VGG16

VGG16 and VGG19 are very famous architectures in the field of Deep Learning and Neural Networks. Our study shows that VGG16 has a good edge over VGG19 over the Visual Saliency detection task. Figure (9) shows the output corresponding to both the models. In the VGG16, we had pretrained weights, trained over 14 million images of the COCO dataset but we did not have pretrained weights of VGG19 network.

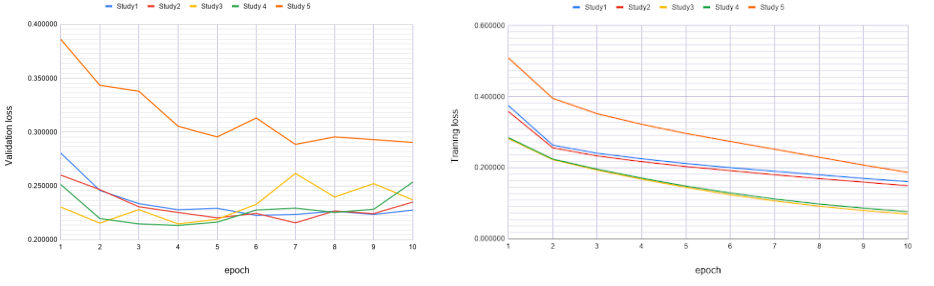
epoch	Study1	Study2	Study3	Study 4	Study 5
1	0.280642	0.260153	0.230425	0.251338	0.386207
2	0.245846	0.246595	0.215516	0.219798	0.343473
3	0.233576	0.230657	0.227983	0.214777	0.337903
4	0.227879	0.225377	0.214803	0.213340	0.305443
5	0.229254	0.220396	0.219034	0.216432	0.295568
6	0.222658	0.224507	0.232956	0.227631	0.313043
7	0.223550	0.215787	0.261693	0.229405	0.288521
8	0.226522	0.226721	0.239779	0.225388	0.295539
9	0.223528	0.224316	0.252155	0.228311	0.293017
10	0.227534	0.235069	0.237158	0.253559	0.290370

(a) Validation Loss

epoch	Study1	Study2	Study3	Study 4	Study 5
1	0.375027	0.358612	0.282207	0.284819	0.509264
2	0.263547	0.256208	0.222864	0.224372	0.395324
3	0.240824	0.233319	0.193384	0.195923	0.352102
4	0.225428	0.217209	0.167803	0.171049	0.322493
5	0.211846	0.203444	0.144828	0.148332	0.296548
6	0.200336	0.191948	0.124025	0.129025	0.274064
7	0.190108	0.180312	0.106666	0.111998	0.252181
8	0.180089	0.169579	0.091923	0.097875	0.229701
9	0.170075	0.159665	0.079779	0.086528	0.207550
10	0.161139	0.149423	0.069869	0.076484	0.186912

(b) Training Loss

Figure 6: An Illustration of Validation and Training Loss for each of the studies for each of the 10 epochs



(a) Validation Loss

(b) Training Loss

Figure 7: An Illustration of Validation and Training Loss for each of the studies plotted for each of the 10 epochs

4.2 Max vs Average Pooling

Average Pooling and Max Pooling are very pooling techniques used for Downsampling in the field of Deep Learning and Neural Networks. Our study shows that Max Pooling has a good edge over Average Pooling over the Visual Saliency detection task. Average Pooling tends to extract all the salient features in the image, even if those are not too salient to be featured. On the other hand, Max Pooling focuses on the most salient features of the image Figure (10) shows the output corresponding to both the Pooling.

4.3 Batch Size

The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters Our studies include models which are trained on different batch sizes, just to see what affect it can bring to the results. Figure (11) shows the difference in output of models trained on batch size 1 and batch size 16. Training on batch-size 16 took significantly less time as compared to batchsize 1 but training and validation loss where better for batchsize 1 because of internal model parameters being updated after going through each sample.

4.4 Dilation Rates

Dilation [10] is a morphological operation used to enhance the features of an image. In the ASPP module of the architecture we have used different sets of dilation rates for experimen-

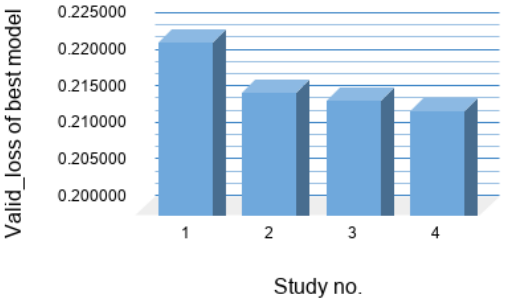


Figure 8: An Illustration of comparison between the Validation losses of the best models for Studies 1-4

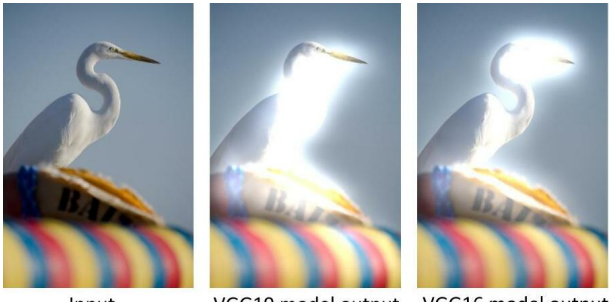


Figure 9: An illustration of final output by VGG19 and VGG16 best models

tation. The original paper [7] used dilation rates 1,4,8,12 but here we used dilation rates 1,4,16 i.e a famous dilation rate sequence and got somewhat better results than that of the original paper which are negligibly small to be recognized by the human eye. Still Figure (12) shows outputs for different dilation rates.

4.5 Categorized Images

The categorical organization of the CAT2000 databases provide 4 individual image classes. The four categories that benefited the most from multi-scale information across all evaluation metrics on the validation set: Noisy, Satellite, Cartoon, Pattern. To understand the

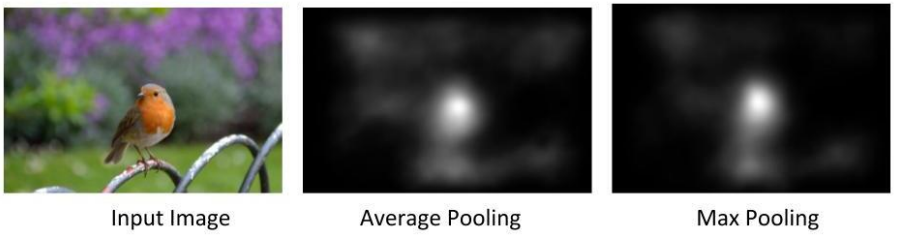


Figure 10: An illustration of final output using Average and Max Pooling for downsampling using the best model of our experiment

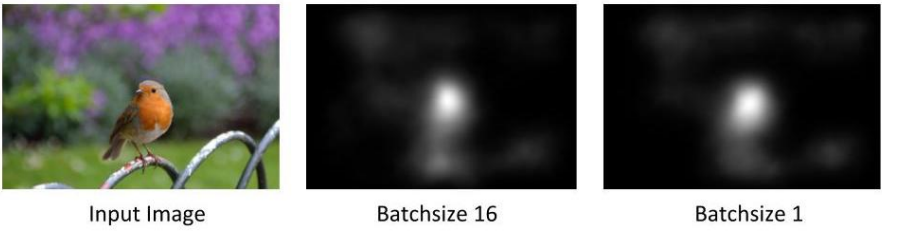


Figure 11: An illustration of final output after training with Batchsize 1 and Batchsize 16. Only a little difference can be observed since the model(VGG16) was same for both

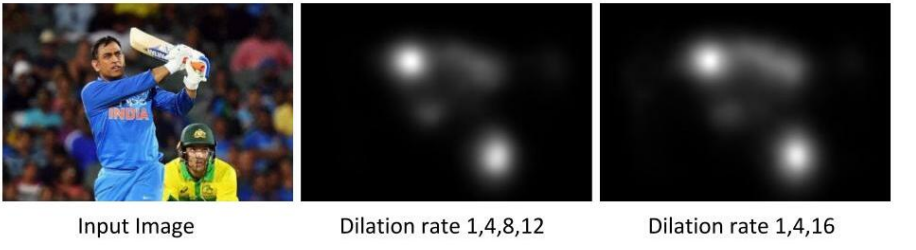


Figure 12: An illustration of final output after training with different Dilation Rates in the ASPP Module of the Model. Only a minute difference visible

measured changes in predictive performance, it is instructive to inspect qualitative results of one representative example for each image category (see Figure (13)). The visualizations demonstrate that large receptive fields allow the reweighting of relative importance assigned to image locations (Noisy, Satellite, Cartoon), detection of a central fixation bias (Noisy, Satellite, Cartoon), and allocation of saliency to low-level features that pop out from an array of distractors (Pattern).

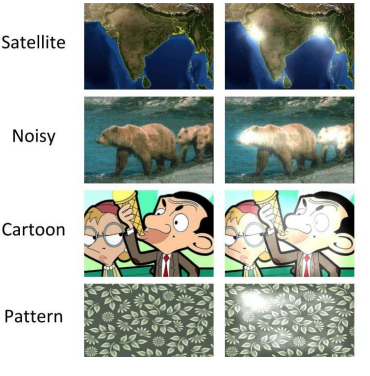


Figure 13: An illustration of final output of saliency, for the 4 kinds of categorization images

5 Discussion

In this section, we discuss what are the difficulties we faced, what design choices we made, what are the future plans, what are our takeaways from this project, and finally what are the resources we took help of.

5.1 Future Plans

We will look forward to implement the model to the Computer Vision applications like scene labeling and categorization and designing logos, pop-ups and stand-outs. We want to explore salient object detection over depth and light field images. One reason behind this is the limited availability of benchmark datasets on these problems. We will try to look the better way to define center bias for the model to reduce the gap between prediction and ground truth. We will like to implement similar model for instance-level saliency detection, which can be helpful in auto image editing applications.

5.2 Difficulties Faced

- Initially, we faced difficulties in training the dataset on our system's gpu Nvidia GTX 1050 2GB.
- Saving, Freezing and Restoring the Tensorflow Model was essential so that we do not miss out on our training due to power failure or any other external factor.
- Trying to come up with better model without having prior knowledge of Visual-Saliency Detection.

5.3 Learnings and Takeaways

- We have learnt different architectures like VGG16 and VGG19 and tensorflow implemntation of these architectures.
- We have understood a new concept of Atrous Spatial Pyramid Pooling that how ASPP is used in processing multiple features in parallel.
- We came to know the role of dilation rate in upsampling.
- We have gained a brief knowledge of Convolutional Neural Networks.
- We have learnt how we can restore trained models on external failures using Techniques of Saving-Freezing-Restoring-Optimizing Tensorflow Models.

5.4 Resources

Resources acquired from the internet and learnings based upon that make a project successfull and well-defined. There are several resources that we took help of to successfully do this awesome project. Here is the list:

- **Tensorflow :** Tensorflow Tutorials link
- **Saving, Freezing, Restoring Tensorflow Models :** Medium Article link

- **VGG16 Architecture :** VGG16 Architecture Github link
- **VGG19 Architecture :** VGG19 Architecture Github link
- **History and Progress Bar :** tf.keras.utils.Progbar link
- **Padding and Resizing Images :** Resizing and Padding Images to keep same Aspect Ratio of image link
- **ASPP Module :** Understanding ASPP Module Youtube link

6 Conclusion

Given that salient regional detection has recently attracted considerable attention, we also conducted a relatively more detailed analysis for this topic. Models can be trained by analysing both low-level and high-level features. Our model includes both low-level features (first two blocks convolutional layers) and high-level features (last three blocks convolutional layers), hence provides competitive results. CNN based model overcomes the limitations of traditional feature detection models, such as insufficient learning and poor robustness [3]. Convolutional layers with large receptive fields at different dilation factors can enable a more holistic estimation of salient image regions in complex scenes. Performance of the model is determined by the parallel expraction of multiple features which is coperated by ASPP. Visual transformations such as mirroring or inversion revealed a low impact on human gaze during scene viewing and could hence form an addition to future work on saliency models.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[2] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[3] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3203–3212, 2017.

[4] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1254–1259, 1998.

[5] J Jonides, DE Irwin, and S Yantis. Integrating visual information from successive fixations. *Science*, 215(4529):192–194, 1982. ISSN 0036-8075. doi: 10.1126/science.7053571. URL https://science.sciencemag.org/content/215/4529/192.

[6] Christof Koch and Shimon Ullman. *Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry*, pages 115–141. Springer Netherlands, Dordrecht, 1987. ISBN 978-94-009-3833-5. doi: 10.1007/978-94-009-3833-5_5. URL https://doi.org/10.1007/978-94-009-3833-5_5.

[7] Alexander Kroner, Mario Senden, Kurt Driessens, and Rainer Goebel. Contextual encoder-decoder network for visual saliency prediction. *CoRR*, abs/1902.06634, 2019. URL http://arxiv.org/abs/1902.06634.

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

[9] D Randall Wilson and Tony R Martinez. The general inefficiency of batch training for gradient descent learning. *Neural networks*, 16(10):1429–1451, 2003.

[10] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.