Saliency Prediction using GAN

Divyam Khanna
Viterbi School of Engineering, University of Southern California, Los Angeles, California 90089, USA
(Dated: May 1, 2021)

Visual saliency is vital for variety of applications including region of interest extraction and medical imaging. This is why it is important to have techniques to have accurate saliency maps. This work explores and analyzes the methodology proposed in [1] which provides a novel method to predict saliency maps using Generative Advsersarial Networks and Adversarial Loss. This method outperforms models which train with simple content loss like Binary Cross Entropy due to addition of Adversarial Loss. This work proposes modifications to both the architecture and of SalGAN and further compares the effect of the modifications with different saliency metrics.

Keywords: Generative Adversarial Networks, Saliency Maps, Computer Vision

I. INTRODUCTION

The term saliency of an object referes to the property of that object which stands out from its neighbours. Here, visual saliency refers to the parts of an image that stands out to humans when we look at it. In other words, it is the parts of images that attracts human attention. A map which enables us to see this visually on an image is called a saliency map. These are very important for various applications. It can be used to find interesting parts of an image or a video. It can be used to do smart cropping of images to reduce size or change aspect ratio. In machine learning pipelines, these maps can be used as an intermediate transformation for a different task for better representation. Another big use case is to make models more interpretable. These maps can allow us to highlight features in input which is supposedly a good image for prediction in a specific model.

The data for these maps are collected by different methods. These include eye trackers and mouse clicks of unbiased human observers not given any specific task. All of this data is then converted to a "heatmap" format which we then call saliency maps (Figure 1).

The original paper has proposed a new method of predicting these saliency maps accurately by using a Generative Adversarial model to make the generated saliency map indistinguishable from the ground truth. Generative Adversarial Networks are a combination of two networks where one part of the network generates from a distribution and tries to replicate samples from the true distribution. The discriminator network on the other hand tries to tell if the generated sample is real or fake. Both of these train simultaneously and each model tries to better at its own job. This slowly makes the generator network good enough to generate close to true sample. They have also proposed a different type of loss called the Adversarial Loss which performs much better than the classic Binary Cross Entropy loss. The goal of this work is to improve upon their methodology and analyse the decisions taken while deciding on the pipeline. This paper replicates their results with an omission and talks about the importance of that.

FIG. 1. Example of Saliency Map produced from an image on Unsplash using the model trained in this study.

II. LITERATURE REVIEW

There has been a lot of research in the field of saliency. The idea of saliency maps was proposed in [2] and [3] articles where they have used low level feature maps and combining them in different ways to predict saliency maps for feature extraction. [4] and [5] presented bottom up and top down models which were vital to the field. Both of these papers introduced important metrics namely AUC-Judd and AUC-Borji. These metrics

are location based metrics where every pixel acts like a classifier of it being fixated or not hence we can use Area Under the Curve metrics.

There also have been other deep learning approaches in this field.[6] blended feature maps from different layers to predict these maps. [7] made important contribution in the field by using a pre-trained AlexNet model on ImageNet. This model was named DeepGaze I. The output of the network is edited to output to give us a saliency map. [8], also known as Salicon Net did the same but used a better model like VGG with pretrained weights. The authors also introduced a dataset for Saliency in Context named SALICON which is used in this paper. [9].

Further advancements like [10] are using LSTM Convolutional networks to achieve state of the art performance. The models used here are extremely complex and unique. [11] is an extension of [7] called DeepGaze II. This performs much better than its previous version and can explain 87% of information gain and is said to be a strong test of transfer learning.

Regarding metrics [12] explains the different types of metrics and what importance they hold for saliency maps.

III. PIPELINE

A. Dataset

The dataset used in this paper is the SALICON [9] dataset. This dataset provides images and saliency maps for those images. The images used in this data come from the MS COCO dataset. The dataset includes 10000 training examples, 5000 validation examples and 5000 test examples. The test samples do not contain the respective saliency maps. This dataset was collected on a large scale by allowing humans to "free view" images and use mouse clicks. The dataset is also attached to a competition which is used to calculate the metrics on the test set. The original images are of dimension 640x480 but have been down-sampled to 256x192 for the purposes of training in reasonable time and develop a proof of concept.

B. Architecture

The architecture is similar to any GAN model as seen in Figure 2. Here there are two networks playing a min max games with each other.

The generator network is responsible for taking a RGB image as input and generating a saliency map for that image. The generator model has two parts, the first part contains convolutional and max pooling layers which acts as a feature extractor or encoder. In the original paper, this part of the model is actually the feature extractor part of the VGG16 model pretrained on ImageNet. The second part of the generator model is a decoder model which has deconvolutional and upsampling layers which outputs an image as the same size as input layer. To produce the saliency map, a 1x1 convolution is used at the end to join all the feature maps and sigmoid activation is applied on it to get the saliency map.

The discriminator model on the other hand is a simple convolutional classifier network. This network takes as input the 1 channel saliency map on top of the 3 channel input image. This makes the input a 4 channel image. It takes in both generated maps and ground truth maps and classifies them as either real or fake. This model contains convolutional and max pool layers with ReLU activation. After the convolutional layers, there are fully connected layers with tanh activation and finally the last layer having sigmoid activation for binary classification.

The job of the generator network is to fool the discriminator model by generating images which are indistinguishable from the real saliency maps. The job of the discriminator network is to catch the generator network and make the right decision if the image is fake or real. In this way, both model try to improve simultaneously which in turn results in saliency maps which are extremely accurate and similar to the ground truths.

1. Architecture Modification

In the original paper, the authors have pre trained the encoder part of the generator network using VGG16 weights on ImageNet. This means that the encoder part of the generator is already pre-trained on ImageNet. In this work, the encoder is not pretrained and has been initialized with random weights. Metrics calculated on the model have been compared with the original paper and some inferences regarding the importance of this are included in the Results and Discussion section below.

C. Training

This model has been defined and trained using the PyTorch framework using an RTX 2070 Super with 8GB of VRAM. The training process took 16 hours. The model has been trained on 100 epochs. While training, we need to make sure that the correct loss function is used to capture the difference between model predictions and ground truth. In the original paper the authors have proposed two types of losses. Content Loss - This is a loss where we compare our original ground truth to our saliency map by per pixel basis. For every pixel we assume it to be an independent binary variable, which will mean each pixel corresponds to the probability of it being attended or not. This loss averages all of the binary cross entropies between every pixel. The loss function will be (1) given S_j and $\hat{S}_j$ are ground truth and predicted saliency maps respectively.

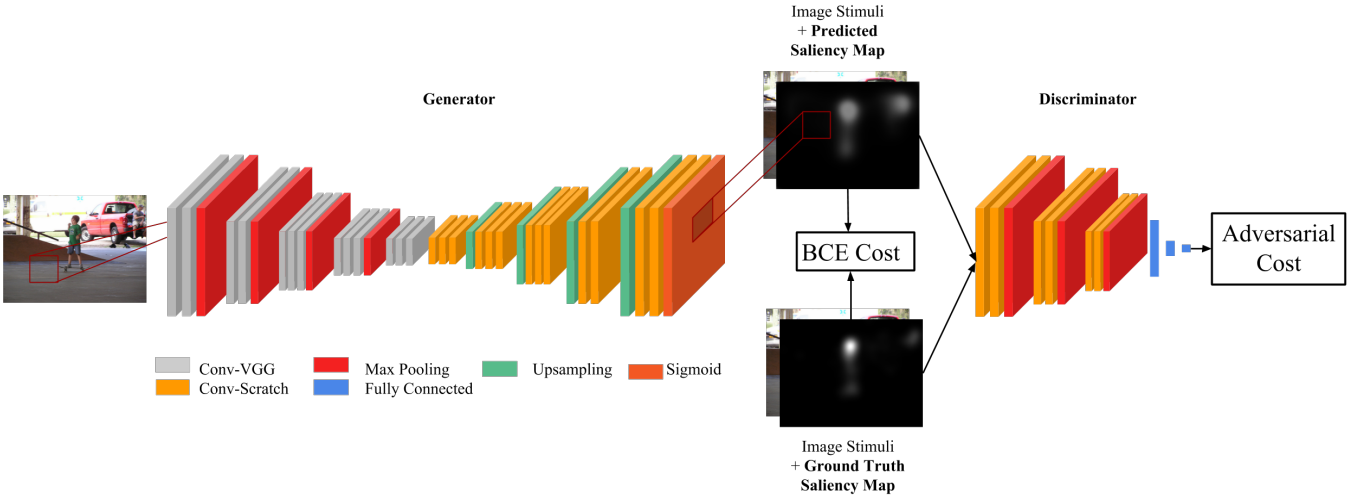

FIG. 2. Original architecture of SalGAN [1]

$$\mathcal{L}_{BCE} = \frac{-1}{N}\sum_{j=1}^{N}(S_j \log(\hat{S}_j) + (1 - S_j)\log(1 - \hat{S}_j)) \quad (1)$$

In this method, the training procedure is alternated between the generator and discriminator after every batch of data and backpropagating the error from the discriminator to the generator. For the generator they have used the combination of the BCE loss and the error from the discriminator. It will be (2).

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{BCE} + L(D(I, \hat{S}), 1) \quad (2)$$

Here, L is the binary cross entropy, D(I,$\hat{S}$) is the chance of fooling the discriminator and 1 is the label for category of real labels. If the chances of fooling the discriminator are lower, the error will be higher. As the BCE loss is only available during training the generator, we need a loss function for the discriminator. This comes out to be (3)

$$\mathcal{L}_{\mathcal{D}} = L(D(I, S), 1) + L(D(I, \hat{S}), 0) \quad (3)$$

The first term refers to the error in classifying ground truth saliency map as real. The second term refers to the error in classifying a fake saliency map as fake. The discriminator is aiming to reduce this combined error.

An example of how the generator trains from predicting noise to accurate saliency maps can be seen in Figure 3. As we can see the results are ideal in epoch 50 and a bit distorted in epoch 100. An additional animation of the generator at every epoch can be seen here.

IV. EVALUATION

After training the model, test set images were fed through the generator to get saliency maps. This was then sent to get saliency metrics from the SALICON evaluation server. The metrics discussed here AUC-B, sAUC, CC and NSS. AUC-B is a location based metric where we calculate the area under curve of True Positives to False Positive. These TP and FP values are the fixations in the generated map compared to the ground truth. sAUC is another version of AUC and stands for shuffled AUC. This metric is just another variant of the previous one and samples the image differently. It gives better score when the map is not predominantly salient in the center. CC stands for Pearson's Correlation Coefficient. This calculates a correlation between the maps and deals with false positives and false negatives symmetrically. CC gives a high value when at any location the generated map and the ground truth have same magnitudes. NSS stands for Normalized Scanpath Saliency. This metric computes the average normalized saliency at fixed location. It is also sensitive to False Positives. The best NSS score will be recieved to the density of fixations distribution.

Model	AUC-B	sAUC	CC	NSS
SalGAN	0.884	0.772	0.781	2.459
Our Model	0.839	0.698	0.756	1.581

TABLE I. Metrics compared with SalGAN on SALICON test set

A comparison between the performance of this model and the original model for these four metrics can be seen in section IV TABLE 1.

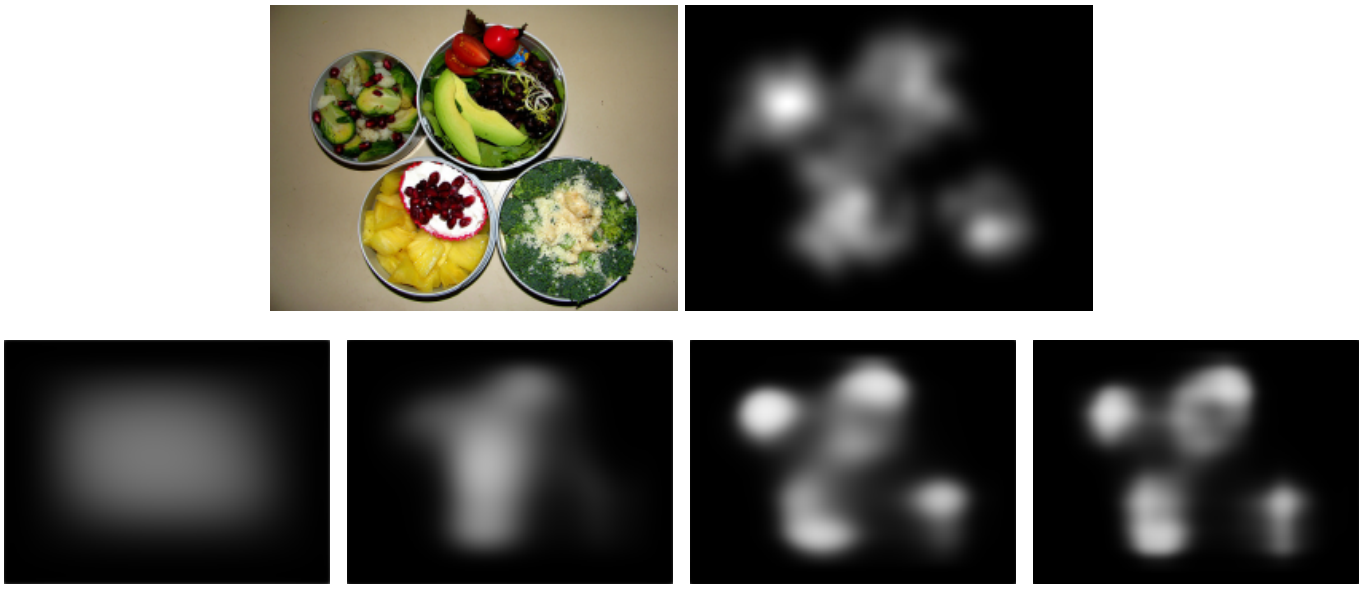

FIG. 3. (Left-Top) An image fed through the generator and the (Right-Top) ground truth for that image. The saliency map at (Left to Right-Bottom) Epochs 5, 20, 50 and 100

V. RESULTS AND DISCUSSIONS

As we can see, the results in Table 1 are comparable to those of the original model but are poorer. This proves the importance of the pre-trained features in the encoder part of the generator network. We can see that the extra information of the pre-trained weights helps the model to immediately find better features and continue to increase attention for those features. The model has knowledge of what features are important and what features are not.

Our model, on the other hand, has to start the feature extraction process from scratch with limited data and limited knowledge. A future work and extension to this work is to try a wide variety of pre trained feature extraction models in place of VGG16 to compare how they perform in the same setting for the same task. Another future work is to experiment with the loss functions to penalize the networks more heavily or lightly.

It can also be discussed that comparing final results with the ground truth is not really a good measure as attention is really subjective. This can be seen and discussed further with the examples in Appendix A.

VI. CONCLUSIONS

In this work we replicate the method of predicting visual saliency maps using GAN as proposed in [1]. We also omit the use of a pretrained VGG16 network as the encoder for the generator and compare the results with the original. We come to the conclusion that it is an important part and this ablation study provides metrics and calculations for how important it is and what it adds to the task at hand. This also supports the power of transfer learning when used correctly and for the right tasks.

DATA AVAILABILITY

The dataset is available at SALICON.

CODE AVAILABILITY

The code and instructions to reproduce can be found at my Github Repository.

ACKNOWLEDGMENTS

I want to thank my professor Dr. Marcin Abram for motivating me throughout the project. I also want to thank him for providing crucial feedback and suggestions to improve and refine this work. I also want to thank my friends who helped me get through the hard work and meet the deadlines. I also want to thank both Dhruvil and Jiacheng for providing great suggestions and improvements during the peer review.

[1] J. Pan, C. Canton-Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giró-i-Nieto, Salgan: Visual saliency prediction with generative adversarial networks, CoRR **abs/1701.01081** (2017), arXiv:1701.01081.

[2] 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**, 1254 (1998).

[3] J. Harel, C. Koch, and P. Perona, Graph-based visual saliency, in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06 (MIT Press, Cambridge, MA, USA, 2006) p. 545–552.

[4] T. Judd, K. Ehinger, F. Durand, and A. Torralba, Learning to predict where humans look, in *IEEE International Conference on Computer Vision (ICCV)* (2009).

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

[6] E. Vig, M. Dorr, and D. Cox, Large-scale optimization of hierarchical features for saliency prediction in natural images, 2014 IEEE Conference on Computer Vision and Pattern Recognition , 2798 (2014).

[7] M. Kümmerer, L. Theis, and M. Bethge, Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet, in *ICLR Workshop* (2015).

[8] 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 (ICCV)* (2015).

[9] M. Jiang, S. Huang, J. Duan, and Q. Zhao, Salicon: Saliency in context, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015) pp. 1072–1080.

[10] N. Liu and J. Han, A deep spatial contextual long-term recurrent convolutional network for saliency detection, CoRR **abs/1610.01708** (2016), arXiv:1610.01708.

[11] M. Kümmerer, T. S. A. Wallis, and M. Bethge, Deepgaze II: reading fixations from deep features trained on object recognition, CoRR **abs/1610.01563** (2016), arXiv:1610.01563.

[12] 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 **41**, 740 (2019).

Appendix A: Additional Results

Some extra results can be seen from the validation set in Figure 4. We show the original image, the ground truth and the saliency map predicted by our model. In some cases, like image 3 in these examples, our model predicts the dog collar as important but the same is not reflected in the ground truth. This result is not necessarily a bad result as it is subjective as to what a person looks at in an image. Similarily, in the last image, the animal in the background and the animal in the foreground has an equal amount of magnitude in the ground truth, but that is not necessarily right. The saliency map predicted by the model gives some attention to that animal (you can see a very slight blob of brightness in that area) but not as much as the one in the foreground.

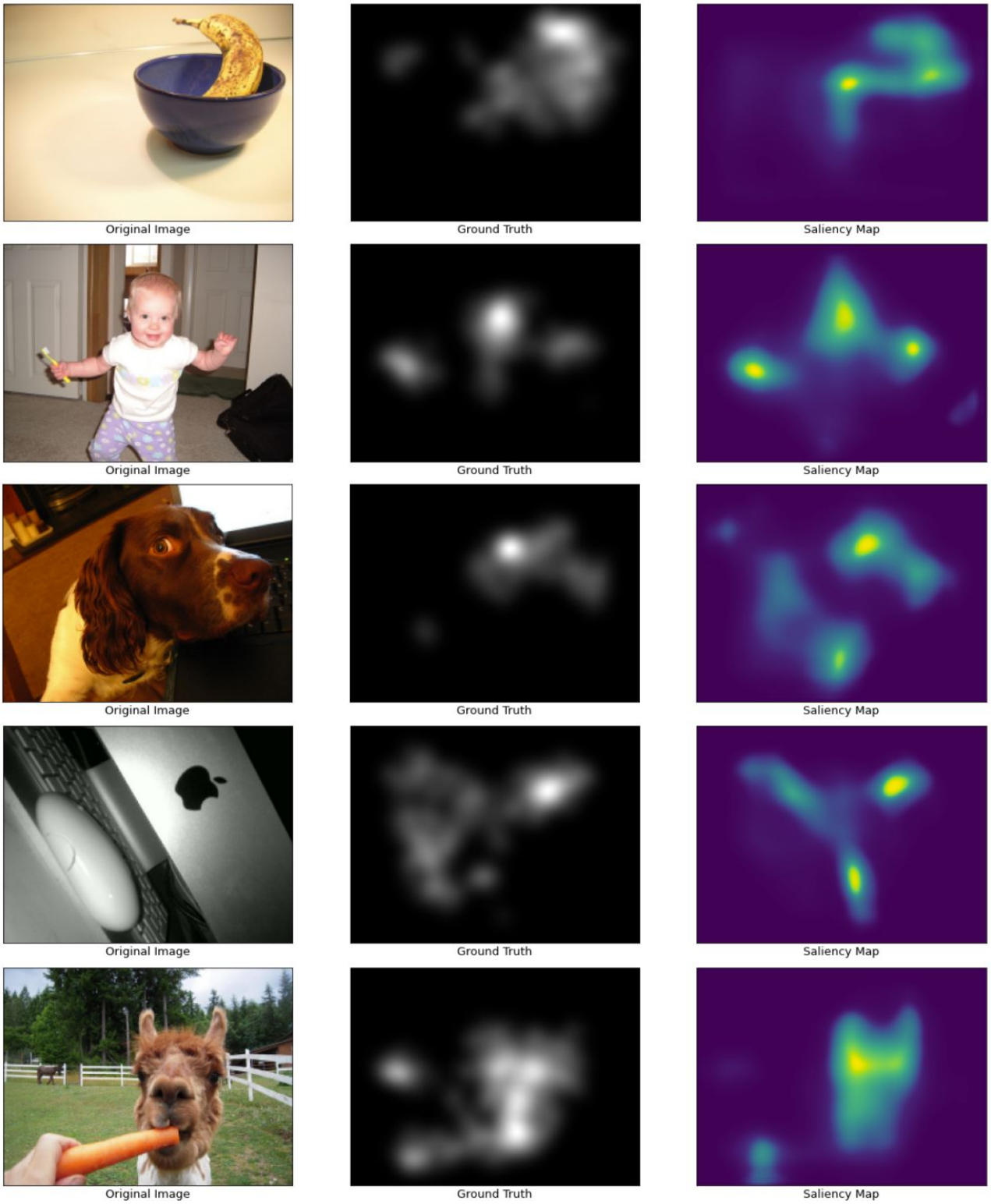

FIG. 4. **(Left)** Original Images **(Middle)** Ground Truth Saliency maps **(Right)** Saliency Maps predicted by my model