Pix2Pix GAN for Saliency Detection: An Investigation into Fixation Prediction.

Chaldaiopoulos Konstantinos
Teacher: Giannakopoulos Theodoros
Course: Deep Learning
University of Piraeus
Digital Systems - NCSR Democritus

***Abstract*-** We investigate the ability of pix2pix Generative Adversarial Networks (GAN) in image fixation prediction and saliency map generation. We adapted a pix2pix Generative Adversarial Network for predicting areas of visual interest in images and generating saliency maps. The CAT2000 dataset, having a diverse range of images along with corresponding eye-tracking data, serves as our primary training and testing. The trained pix2pix model accepts raw images and highlights the predicted areas of visual fixation.

Index Terms- pix2pix, generative adversarial networks, GAN, salient maps, fixation prediction, CAT2000

I. Introduction

In computer vision, an area of interest is to understand what parts of an image draw human attention as known as visual saliency. This concept, used both in computer vision and cognitive psychology, is based in the idea that humans focus their attention in different areas and do not process all part of a scene equally.

The goal is to intertwine these two disciplines to create algorithms that can mimic or predict this human attention mechanism and identify the most "salient" part of the image.

Pix2pix [2] GAN [1] was chosen for this task because it has demonstrated an exceptional performance in domain translation, meaning the conversion of images from one style to another style. However, it is important to note that saliency maps are not just another "style". As mentioned, salient maps highlight the fixation parts of human vision. The interest lies in observing the ability of the model to go beyond mere aesthetic alterations and capture the underlying cognitive patterns of human visual perception, demonstrating ability in predicting where a human would fixate on an image.

Finally, using GANs to generate salient maps is not without precedent [6]. This fact provides us with a strong basis for choosing a pix2pix GAN for our research.

In summary, this research process serves a dual purpose: enhancing our understanding of the complex mechanisms of human visual attention and contributing to the development of efficient models for visual saliency prediction. The aim is to create models that can accurately mimic and predict human attention mechanisms, by researching new possibilities in the field of computer vision.

II. Architecture

2.1 U- Net architecture [3]: A typical U – Net architecture used in pix2pix GANs [2] as a generator for inputs of size 256x256 is:
encoder:
C64-C128-C256-C512-C512-C512-C512-C512
decoder:
CD512-CD1024-CD1024-C1024-C1024-C512 -C256-C128
All the convolutional layers utilize batch normalization except from the first layer (64) and a leaky ReLU activation function with a slope of 0.2. "D" in "CD" denotes the use of a 0.5-dropout layer between the batch normalization and the ReLU activation function. The convolutions are usually 4x4 spatial filters applied with stride 2. The decoder size in U-net is double the size expected in a typical encoder-decoder network. This is since skip connections are established between the encoder and the decoder (Figure 1) to enable information flow from the low levels of the model. To be more specific n-1 layers of the decoder are concatenated with the n layers of the encoder. In the last layer, a Tanh activation function is applied.

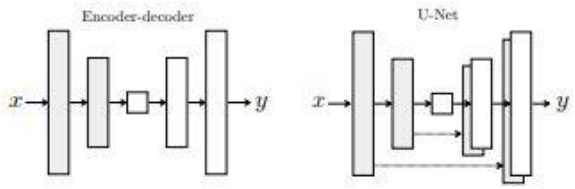

Figure 1: Difference between a typical encoder-decoder architecture and U-net architecture.

2.2 Patch-Gan architecture [3]: A typical patch-Gan architecture for inputs of size 256x256 is presented below. It is a common practice to upscale the images to 286x286 and then randomly crop them back to size to apply jitter before feeding them to the network. The architecture:
Architecture: C64-C128-C256-C512
Again, All the convolutional layers utilize batch normalization except from the first layer (64) and a leaky ReLU activation function with a slope of 0.2. After the last layer, a convolution is applied to map to a 1-dimensional output, followed by a Sigmoid function. [3]. The last layer produces a 30 x 30 feature map which corresponds to a 70x70 patch in the input image.

2.3 Our architecture: The architecture we used is similar with some slight modifications. Our model consists of U-net generator and a PatchGan discriminator. Due to a difference in the input size (250x140) there were also some differences to the spatial filters and strides. Before passing the images to the generator network, they were first interpolated to 256x144 and in the output they were converted again back to 250x140.The interpolation is only used to create easier model architectures without the need for many modifications in kernel sizes and strides:

2.3.1 U- Net architecture:

encoder:

C64-C128-C256-C512-C1024

decoder:

CD1024-CD2042-CD1024-C512-C256- C128

After each layer, instead of batch normalization we use instance normalization due to a batch size of 1, except as mentioned from the first layer (64). Like in the original paper, batch normalization is followed by a ReLU activation function with a slope of 0.2 and in the decoder dropout layers are added. The model uses a kernel size of 3 and a padding of 1 in the bottleneck layer of the decoder because of the different input dimensions from the original paper [3].

2.3.2 **Patch-GAN architecture**

The Patch-GAN architecture used was the same as in the paper but with double the layers.

Architecture: C128-C256-C512-C1024

Since we use a binary cross-entropy loss with logits function, we chose to not use the sigmoid activation function in the last layer to map into 1-dimension since the loss function already uses sigmoid. Again, instance normalization was used instead of batch normalization. The above architecture proved more efficient for our task.

III. TRAINING DETAILS

Utilizing the described Generator and Discriminator we trained our pix2pix GAN. The models were initialized using a custom weights initialization strategy. This involved setting Convolution layers to a normal distribution around 0.0 with a standard deviation of 0.02.

In the training phase, our model optimizes two key loss functions - the Binary Cross-Entropy with Logits (BCE) and the L1 loss, each serving a specific purpose in the Generative Adversarial Network (GAN) framework.

The BCE loss is employed in a different context for the Generator and Discriminator within the GAN. For the Generator, it serves as an adversarial loss, incentivizing the generation of images that the Discriminator mistakenly classifies as real. For the Discriminator, it evaluates its ability to distinguish between real and synthetic images. The Discriminator's goal is to correctly classify real images as real (by outputting ones) and fake images as fake (by outputting zeros).

The L1 loss ensures that the generated images not only fool the Discriminator but also closely resemble the original images in terms of structural features. To balance the importance of the GAN loss and the L1 loss, the latter is typically scaled by a factor of 100, referred to as lambda_L1. This practice is quite prevalent in the training of pix2pix GANs.

Over multiple epochs, our model processed pairs of original images and their corresponding saliency maps. The training leveraged the Adam optimizer with a learning rate and betas set to 0.00001 and (0.5, 0.999), respectively. During each iteration, the Generator crafted a 'fake' image from its corresponding real image. The generator loss, computed as the sum of the GAN loss and the L1 loss. Simultaneously, the Discriminator calculated its loss as the average of losses associated with the real and fake images. These losses were then utilized to optimize the Discriminator's and Generator's weights.

To assess the effectiveness and progress of the training process, the losses for both the Generator and Discriminator were tracked and displayed every 500 iterations. Beyond serving as a monitoring tool, this also helps identify any potential issues during training, such as model convergence problems.

Additionally, at the end of each epoch, the state of the Generator was preserved by saving its model parameters. This allows for an in-depth analysis of the Generator's performance over time. By examining the output images generated at different stages of training, we can gain insights into how the Generator's ability to create realistic images evolves throughout the training process. It is also important to note, that even when the model reaches a state of convergence, it could still produce higher quality images over time. This is because the interplay between the generator and discriminator in a GAN may continue to refine the details of generated images. However, numerical convergence doesn't always equate to optimal visual results. Hence, involving human judgement in the process is paramount. By saving the generator's state at various training stages, we ensure a spectrum of outputs, each representing a unique point in the learning process. This allows a human evaluator to choose the model that delivers the most visually satisfying results, bridging the gap between mathematical convergence and perceptual image quality.

The training time for 100 epoch was ~ 1 hour in an NVIDIA A100 Tensor Core GPU.

IV. RESULTS

After completing the training process, the performance of our pix2pix saliency GAN model, was evaluated using two primary assessment methods: the Normalized Scanpath Saliency (NSS) [7] metric and visual inspection.

1. Normalized Scanpath Saliency (NSS): the NSS score, a standard measure in the field of visual attention, provided a quantifiable indication of the model's performance. It reflects the correlation between the predicted saliency map and the human fixation data. Remarkably, the NSS score achieved by our model was 1.43. Contrasted to the minimum human baseline of 1.54 the results are significant. However, it is important to note that there are model architectures that achieve much higher accuracy like CEDNS [4] which achieve an accuracy of 2.39 which is the highest recorded in the CAT2000 [5] dataset. It is worth noting that when we used a PatchGan of C64-C128-C256-C512 architecture, the NSS score was lower (1.37).

2.Visual Inspection: In addition to the NSS metric, we evaluated our model using visual inspection of the generated saliency maps. This method allows us to assess the performance of our model by examining ourselves the visual saliency, but also, examining the quality of the produced images. Upon reviewing the generated images, we found them to be of decent quality,

exhibiting strong likeness to the corresponding real salient maps. The saliency maps successfully highlighted the areas of interest that would likely attract human attention in many cases, providing further evidence of the model's efficacy.
Below are presented some of our results (Figure 2):

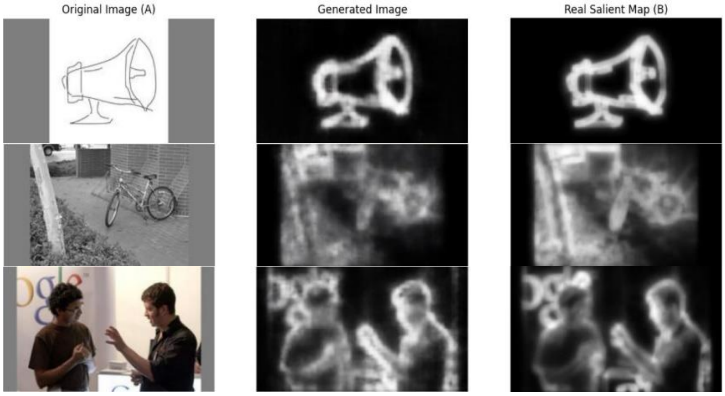

Figure 2: Examples of our results

V. Conclusions

In this study, we demonstrated the potential of pix2pix GAN in image fixation prediction and saliency map generation Our findings indicate that the pix2pix model can successfully adapt to the task of predicting areas of visual interest in images, marking a noteworthy progress in the development of predictive models for visual saliency.

The NSS score achieved by our model, 1.43, although lower than the current state-of-the-art models, such as CEDNS (with a score of 2.39), is significantly higher than random models, showing that our approach is promising. The visual inspection of the generated images confirmed the ability of our pix2pix GAN to produce qualitatively pleasing saliency maps that adequately captured areas of human visual interest.

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. In arXiv:1406.2661.

[2] Isola, P., Zhu, J., Zhou, T., & Efros, A. A. (2016). Image-to-Image Translation with Conditional Adversarial Networks. In arXiv:1611.07004.

[3] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In arXiv:1505.04597

[4] Qi, F., Lin, C., Shi, G., & Li, H. (2019). A Convolutional Encoder-Decoder Network With Skip Connections for Saliency Prediction. In IEEE Access (Vol. 7, pp. 60428-60438). IEEE. doi:10.1109/ACCESS.2019.2915630.

[5] Borji, A., & Itti, L. (2015). CAT2000: A Large Scale Fixation Dataset for Boosting Saliency Research. In arXiv:1505.03581.

[6] Pan, J., Ferrer, C. C., McGuinness, K., O'Connor, N. E., Torres, J., Sayrol, E., & Giro-i-Nieto, X. (2017). SalGAN: Visual Saliency Prediction with Generative Adversarial Networks. In arXiv:1701.01081.

[7] Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., & Durand, F. (2016). What do different evaluation metrics tell us about saliency models? In arXiv:1604.03605.

Author: Chaldaiopoulos Konstantinos
E-mail: kostachaldaio@gmail.com