Image Inpainting with Generative Models

Daniyar Nariman, Sultanov Azat, Nazyrov Amir
Innopolis University
Course: Computer Vision 2019

Abstract— **Recent advances in Deep Learning have shown an exciting promise in filling the gaps in the natural images. In computer vision this kind of tasks called image inpainting, which aims to synthesize alternative contents in missing regions based on the context. Many approaches have been proposed on this kind of task, that confirms about its relevance in the current days, but it still remains a challenging problem. In this project we will briefly explain the current approaches of Image Inpainting task and show the working prototype of one of the methods.**

I. INTRODUCTION

Image inpainting is a task of synthesizing visually realistic and semantically correct contents in missing regions.[1] There are mainly two broad approaches: classical patch-based ones and convolutional networks based deep generative models. Patch based approach can synthesize plausible stationary textures but typically does not work well in cases where semantics are needed. The CNN based approach can utilize semantics learned from large scale datasets. In this paper we will focus on the second approach as it seems more promising.

II. APPROACHES

In this section we will describe some of the approaches, which was developed for the past few years, but before going deeply on the methods of Image Inpainting, we will shortly explain abstract workflow.

- Feed the generative network with an input image with holes than need to be filled. These patches can be considered a hyperparameter required by the network since the generator has no way of discerning what actually needs to be filled in. To make network understand what part need to be filled, we need a separate layer mask that contains pixel information for the missing data.
- The generator will produce the entirely synthetic image generated from scratch.
- The layer mask allows us to discard those portions that was already presented in the incomplete image, since we did not interested in those parts of the image.
- The new generated image is then combined with the incomplete image that we sent to the network at the beginning, so that we can get our new image that generated pixels at the place where the hole was.
- This generated image is then sent to discriminator network. This network tries to ensure that the image does not look obviously fake.

A. Context Encoder [2] [3]

It is a convolutional neural network trained to generate the contents of an arbitrary image region conditioned on its surroundings.

1) Generator: The overall architecture is a simple encoder-decoder pipeline. The encoder was to be derived from the *AlexNet* architecture and composed of five convolutional layers. The output of the encoder is connected to the input of the decoder by a channel-wise fully-connected layer (6*6*256=9126 activations layers), in order to propagate the information within each feature map. The decoder follows with five up-convolutional layers that augment the dimensions of the latent representation in-between the encoder and the decoder.

2) Discriminator: Five layer convolutional network that takes as input the context either completed with the output of the encoder-decoder pipeline (which is the generated missing area) or the ground truth missing area.

3) Loss function: The overall joint loss function is shown below.

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv}$$

where the first part is *reconstruction loss* which is equal to

$$\mathcal{L}_{rec}(x) = || \hat{M} * (x - F((1-\hat{M}) * x)) ||_2^2$$

where x designate the input image and $\hat{M}$ the binary mask such that $\hat{M} * x$ (Hadamard product) selects the missing region. The reconstruction loss is the L2 distance between the content generated by the network and the original region of the image that has been masked to the network, and the second part is represented as,

$$\mathcal{L}_{adv}(F, X) = E_{X \in \mathcal{X}}[logD(X)] + log(1 - D(F((1-\hat{M}) * x))$$

F here is the function of our proposed context encoder and λ is hyperparameter.

The main advantage of this approach is that, because of encoder-decoder based architecture, it learns a representation that captures not just appearance but also the semantics of visual structures, that gives the model more power in the case of image inpainting task.

B. Another Approach

another approach to be added

C. Another Approach

another approach to be added

III. Experiments and Evaluation

A. Implementation

As the part of the research, before training on our dataset, we conducted an experiment on Context encoder GAN, by training it on 240 images. The results are represented below in Fig. 1.

The implementation of Context encoder GAN you can find in the following link: https://github.com/nariman9119/image-inpainting-research

The next checkpoint for our team, is to train this model on generating the mustaches on the cropped images.

B. Dataset

The training that we planning to use is a Large-scale CelebFaces Attributes (CelebA) Dataset with the attribute ”Mustache”. The number of images in this dataset 8417.

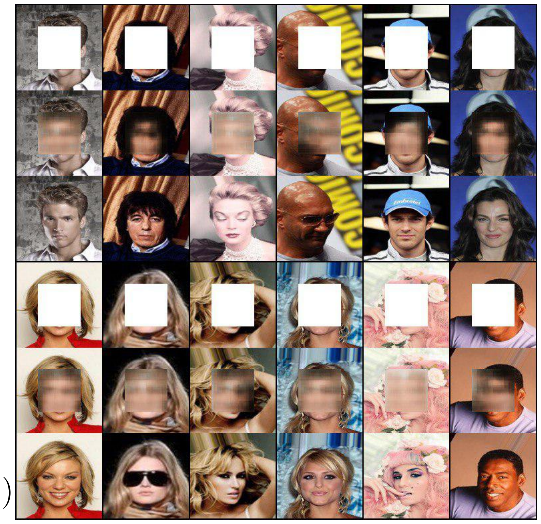

Fig. 1. Each image sample has 3 states: cropped, generated from our Context encoder, original

IV. Conclusion

TODO

References

[1] Tianyuan Zhang, 2018. Survey on Image Inpainting.

[2] Deepak Pathak, Philipp Krahenb, Jeff Donahue, Trevor Darrell, Alexei A. Efros, 2016. Context Encoders: Feature Learning by Inpainting

[3] Quentin LEROY and Bastien PONCHON 2018. Understanding Context encoders: Feature Learning by Inpainting.