DATA SCIENCE ENGINEERING METHOD AND TOOLS

GP-GAN: Towards Realistic High-Resolution Image Blending

DIVYA UNNIKRISHNAN (001001182)
SHUBHANG SHAH (002111376)

INTRODUCTION

It is common but challenging to address high-resolution image blending in the automatic photo editing application. Gaussian-Poisson Generative Adversarial Network (GP-GAN) was introduced in the research paper GP-GAN: Towards Realistic High-Resolution Image Blending by Huikai Wu, Shuai Zheng, Junge Zhang, Kaiqi Huang.

The code for our implementation is on Github.

IMPORTANT TERMINOLOGIES

- **Image Editing**
    - Image editing refers to modifying or improving digital or traditional photographic images using different techniques, tools, or software. Image editing is done to create the best possible look for the images and also to improve the overall quality of the image according to different parameters.
- **Image Blending**
    - Image Blending is mixing two images of the corresponding pixel values to create a new target image. The concept of blending images is comparatively very easy. To achieve this, we can simply make a copy of an image and transfer each source pixel's values into a pixel in the target image

- **Image Processing**
    - Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it
- **Generative Adversarial Networks**
    - Generative Adversarial Networks, or GANs for short, are an approach to generative modeling using deep learning methods, such as convolutional neural networks.
    - In GANs it is a zero-sum game of learning a generator and discriminator.
    - In a generator model with the given input data, it learns the patterns from input data in such a way that it generates new outputs which can be plausibly generated from the original dataset
    - In a discriminator model, it aims to distinguish between generated and the input data
- **Poisson Editing**
    - Poisson Blending is an image processing technique, where the image processing operator allows the user to insert one image into another, without introducing any visually unappealing seams. It also makes sure that the color of the inserted image is also shifted so that the inserted object feels as if it is part of the environment of the target image.
- **CONDITIONAL GANS**
    - Our work is also related to conditional GANs, which aim to apply GANs in a conditional setting. Conditional GANs has been applied to discrete labels, image inpainting, image prediction from a normal map, image

manipulation guided by user constraints, product photogeneration, style transfer, and Image-to-Image translation

  - Different from previous work, we use an improved adversarial loss and discriminator for training the proposed Blending GAN. The paper proposes the Gaussian-Poisson Equation to produce high-resolution images.

DATASET

We are **Transient Attributes** dataset. The dataset consists of **8571 images from 101 webcams, all annotated with 40 attribute labels**. This dataset is licensed under a Creative Commons Attribution 4.0 International License.

Gaussian-Poisson GANs

- **PURPOSE**

The purpose of GP-GANs is solely to bridge the talent gap between expert users and beginners on image editing. Given a source image x, a destination image y, and a mask image mask, using the copying-and-pasting strategy, a composite image x can be obtained by:

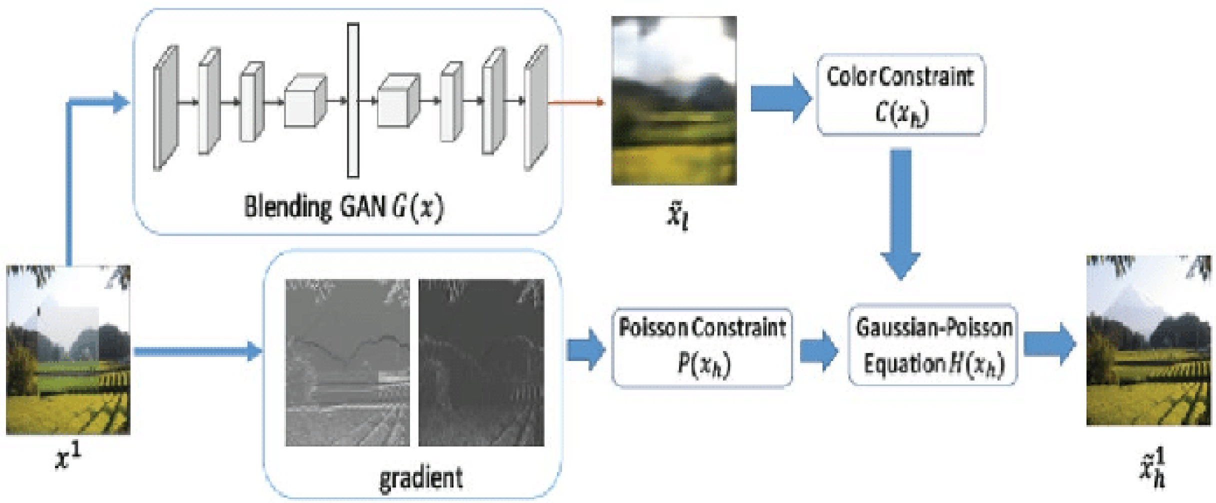
$$x=x*mask+y*(1-mask)$$

The goal of conditional image generation is to generate a well-blended image ~x that is semantically similar to the composited image x but looks more realistic and natural with resolution unchanged. x is usually a high-resolution image.

- **ARCHITECTURE**

GP-GANs generate high resolution and realistic images using GAN. It seeks a well-blended high-resolution image by optimizing a loss function composited by color constraint and gradient constraint. The color constraint tries to make the generated image more realistic and natural while the gradient constraint captures the high-resolution details like textures and edges.

With the help of Blending GAN, a low-resolution realistic image is generated that helps in generating color constraints. With the help of the Gaussian-Poisson Equation, we generate a high-resolution image including textures and edges of the composited image.



- **BLENDING GAN**

We seek a low-resolution well-blended image ~xl that is visually realistic and semantically similar to the input image. A straightforward way is to train a conditional GAN and use the generator to generate realistic images. Since we have both the input image and the corresponding ground truth xg, we aim to train a generator that can produce a realistic image approximating xg.

l1 and l2 loss can fasten the training process but tend to produce blurry images. The perceptual loss is good at generating high-quality images but is very time and memory-consuming. We explore l2 loss because it could shorten the training process and generate sharp and realistic images when combined with GANs. The combined loss function is defined as:

$$L(x,x_g)=\lambda L_{l2}(x,x_g)+(1-\lambda)L_{adv}(x,x_g)$$

where λ is 0.999 in our experiment. L_{l2} is defined as:

$$L_{l2}(x,x_g)=\| G(x)-x_g \|_2^2$$

and L_{adv} is defined as:

$$L_{adv}(x,x_g)=\max_D E_{x\in\chi}[D(x_g)-D(G(x))]$$

- **GAUSSIAN-POISSON EQUATION**

Networks like the proposed Blending GAN G(x) could only generate low-resolution images. Even for slightly larger images, the results tend to be blurry and have unpleasant artifacts, which is unsuitable for image blending as the task usually need to combine several high-resolution images and blend them into one realistic, high-resolution image. To make use of the natural images generated by Blending GAN G(x), the Gaussian-Poisson equation is proposed. It is fashioned by the well-known Laplacian pyramid for generating high-resolution and realistic images.

We observe that although our Blending GAN G(x) couldn't produce high-resolution images, the generated image $\tilde{x}_l$ is as natural and realistic as a low-resolution image. So it is possible for us to seek a high-resolution and realistic image $\tilde{x}_h$ by approximating the color in $\tilde{x}_l$ while capturing rich details like texture and edges in the original high-resolution image x. We formulate the requirement using two constraints: one is the color constraint, the other is the gradient constraint. The color constraint forces $\tilde{x}_h$ to have a similar color to $\tilde{x}_l$, which can be achieved by generating a $\tilde{x}_h$ with the same low-frequency signals as $\tilde{x}_l$. The simplest way to extract the low-frequency signals is using a Gaussian filter. The gradient constraint tries to restore the high-resolution details which are the same as forcing $\tilde{x}_h$ and $\tilde{x}_l$ to have the same high-frequency signals. This could be implemented by using gradient or divergence.

Formally, we need to optimize the objective function defined as:

$$H(x_h)=P(x_h)+\beta C(x_h)$$

$P(x_h)$ is inspired by the well-known Poisson Equation [25] and is defined as:

P(xh)=∫T // divv−Δxh // 22dt

C(xh) is defined as:

C(xh)=∫T // g(xh)−~xl // 22dt

and β represents the color-preserving parameter. We set β to 1 in our experiment. In Equation 6, T represents the whole image region, div represents the divergence operator and Δ represents the Laplacian operator. v is defined as:

vij={∇xsrcif xijmask=1∇xdstif xijmask=0

where ∇ is the gradient operator. Gaussian filter is used in Equation 7 and is denoted as g(xh). The discretized version of Equation 5 is defined as:

H(xh)= // u−Lxh // 22+λ // Gxh−~xl // 22

u is the discretized divergence of the vector field v, L is the matrix that represents the Laplacian operator while G represents the Gaussian filter matrix. xh and ~xl are the vector representation of xh and ~xl.

- **FLOW AND RESULTS**

Please find the flow of the model in the given README.md file of our Github repo.

REFERENCES

- https://arxiv.org/pdf/1703.07195.pdf
- https://github.com/VicenteAlex/Tensorflow_GP-GAN
- https://deepai.org/publication/gp-gan-towards-realistic-high-resolution-image-blending
- https://github.com/wuhuikai/GP-GAN