Image Reconstruction through Generative Adversarial Networks

Michael Huang, Kevin Frans
Henry M. Gunn High School

Abstract

For many years, inpainting has been a focus in the field of computer vision. To be able to reconstruct damaged portions of an image has numerous applications, from realistically repairing photographs to smoothly removing unwanted areas in digital image editing.

Recently, the Generative Adversarial Network (GAN) has shown to perform well when tasked with recreating natural images.

In this paper, we aim to evaluate the GAN in comparison to a direct convolutional network, for the task of recreating missing areas of various images.

Introduction

Convolutional networks

In fully-connected neural networks, each neuron in a layer is connected to each neuron in the next layer. This works well when input data are independent. In an image, however, pixel colors are correlated with the pixels nearby. In addition, similar image features such as lines or circles often have the same meaning when present in different areas of the image. The convolutional network takes advantage of these assumptions. Rather than having a distinct weight and bias for every pixel in an image, a convolutional layer instead consists of a small filter. This filter is slid across the image, and at each location a matrix multiplication is calculated, with the results making up the next layer's input. In doing so, a convolutional layer requires much fewer parameters than a fully-connected layer, with the assumption that features look the same in different areas of an image.

GANs

Traditionally, convolutional neural networks have employed L1 or L2 loss when tasked with image generation. These loss functions attempt to measure the mathematical distance between an original image and a generated one. Simply reducing the distance, however, can lead to blurry results, as colors converge to a mean of two plausible colorations rather than choosing one or the other. The GAN attempts to remedy this issue, by learning the loss function itself. It formulates image generation as a game between two networks: the discriminator, and the generator. The discriminator is trained to differentiate between real and generated images. At the same time, the generator is trained to trick the discriminator into believing its generated images as real. The discriminator acts as a loss function for the generator, resulting in images following a more natural distribution.

Related work

In recent years, Convolutional Neural Networks (CNNs) have greatly enhanced the AI field, performing exceptionally well in classification[s] and segmentation[s], improving the computer vision field as well. The success of CNNs has led to their use in harder problems such as understanding features in the image, and image generation.

Image Generation

Dosovitskiy *et al* [1] demonstrates the how CNNs can be applied to the task of image generation of chairs, through the use of very large labeled datasets of 3D chair models.
GANs have gained prominence as another method for image generation, as Radford *et al* proposed a new architecture for DCGANs, producing conving results. Others, such as Goodfellow *et al* [2], use a convolutional network for the generator model.

Inpainting

The main departure from the models described above is the presence of a conditional layer. Denton *et al*'s work [3] showed that a Laplacian pyramid of adversarial generator and discriminator could generate images at multiple resolutions, and could also be conditioned for more controlled output.
One thing to note is that traditional methods such as the Navier-Stokes model, commonly applied to small inpainted regions, cannot be applied to our task, since the missing region is too large for non-semantic methods.

Our Approach

To compare the direct convolutional network and the GAN, we formulate an inpainting task. We make use of celebA, a dataset of 200,000 celebrity images. Each image is cropped to a square, and resized down to 64 x 64 pixels. To form the inputs, we remove the the center 32x32 square of the image, filling it with black. Both networks are tasked with accurately recreating this inner square to a natural standard.

Direct Network

The direct network consists of stacked convolutional layers, each with a stride of two and a filter size of 5x5. The input starts as a 64x64x3 tensor, and with each additional layer, height and width shrink by a factor of two, while the feature dimensionality increases. Accordingly, the first four layers have a size of [64x64x3 -> 32x32x64 -> 16x16x128 -> 8x8x256 -> 4x4x256]. Then, a fully-connected layer is used, mapping from 4x4x256 to 4x4x256. Finally, four transpose convolution layers are utilized to bring the feature matrix back to the image's original size.

Additionally, we employ a few tricks to improve training. Each convolutional layer is followed by a rectified linear unit, introducing non-linearity. Batch normalization is also applied, allowing greater gradient flow.

GAN network

In the GAN method, the generator network is structured the same as in the direct network. However, we must introduce a second network: the discriminator. The discriminator's structure is similar to the first half of the generator: convolutional layers are stacked until a dense 4x4x256 matrix is reached. Then, a fully-connected layer connects this feature matrix to a single output, which represents the discriminator's belief as to whether the image is real or generated.

Experiment

The input to our networks is an image with the central portion of the image cropped out, meaning that they held zero information. These portions were set to black. The location of the "dropped" portion could have been anywhere, but the center was chosen because the most features were a part of the center portion most often.

Region Block

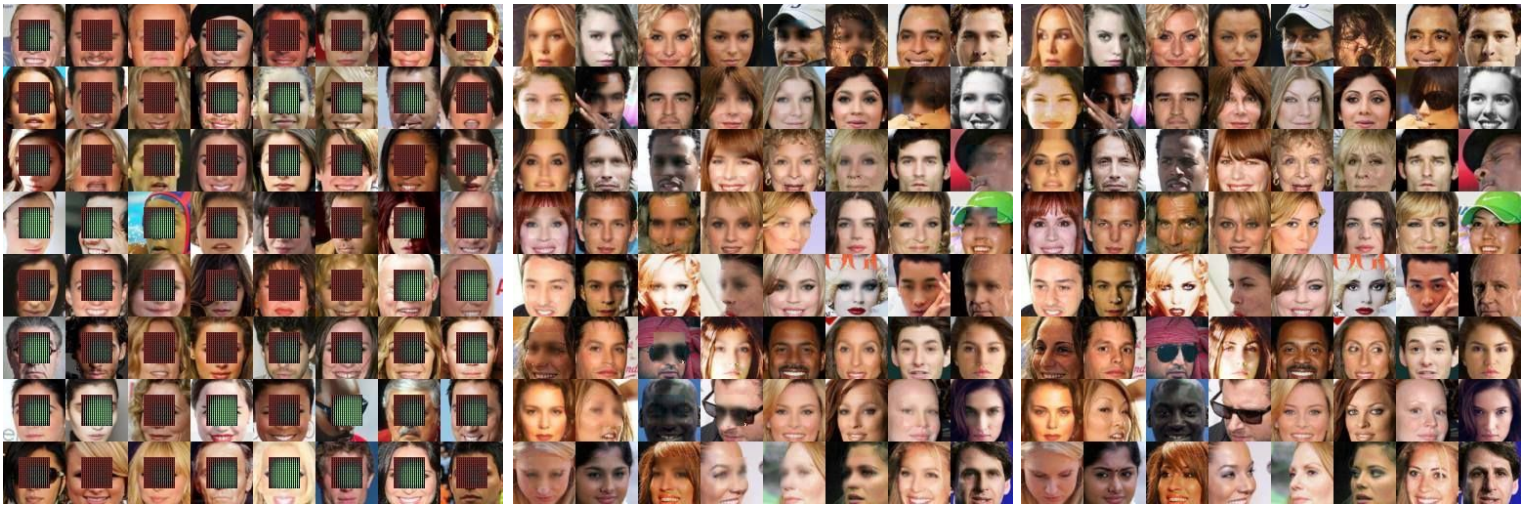
In our experiment, dataset was taken from a random subset of CelebA, downsized to 64x64, and a central square block, 32x32 size, was cropped out of each image in the dataset, filled with black.

Implementation

The networks were implemented in Tensorflow, the Direct network was implemented and run first, then modified to the specifications of the GAN network, both with a batch size of 64.

Results

The results for the direct method are shown below.



Dataset Recreation Original

These results show that the direct method recreates the missing portion to an realistic degree. The recovered images with this method look greatly similar to the original untouched dataset, there are sharp edges around the cropped portion, and features are recreated as well.

Conclusion

Our data shows that networks successfully converge when tasked with recreating missing content in an image. The examined final photos look natural to a realistic degree, confirming three major claims: the network can backpropagate and learn patterns, faces contain similar features across all humans, and the faces dataset is comprehensive enough to generalize onto all faces of a similar angle.

Our research tested the effectiveness of Convolutional Neural Networks and Generative Adversarial Networks on repairing images. These two networks are widely used in machine learning for image generation, and during our research project we found out that work has been done applying these networks specifically to what we focused on, inpainting.

Our research agrees with past findings, improving on past methods as well as contributing to the current research on image generation and inpainting. Past findings agrees with past models that require the user to specify the damaged region, as well as those that only infers what's missing based off of the the colors and gradients around the missing portion. Our research agrees and shows that these methods are effective at solving the problem of inpainting, and improves it.

There were some weaknesses in our investigation. We only tested on a dataset of faces - the same method could lead to varying results when used on, say, a dataset of landscape photographs. In addition, we arbitrarily chose hyperparameters such as hidden layer size, filter size, and the number of layers. These parameters could potentially be adjusted to change the after-training results.

Our work paves the way for future research in this topic. There are many directions that could be taken to improve results. As mentioned earlier, hyperparameters could be adjusted. The training method (stochastic gradient descent) could be modified to have an adjustable learning rate. Finally, we could take an iterative approach, passing the image back into the network multiple times, each time increasing in detail.

References

[1] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. CVPR, 2015.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014

[3] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In Advances in Neural Information Processing Systems, pages 1486–1494, 2015.

Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson: “Semantic Image Inpainting with Perceptual and Contextual Losses”, 2016; arXiv:1607.07539.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell: “Context Encoders: Feature Learning by Inpainting”, 2016, CVPR 2016; arXiv:1604.07379.