Image Inpainting using Convolutions and Transposed Convolutions

Nishchal MN
PES1201701523
PES Univeristy
Bangalore, India
nishchalmn619@gmail.com

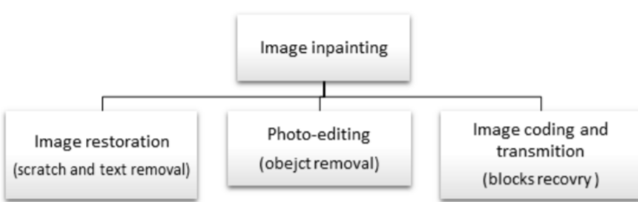
Hitesh Kumar
PES1201701523
PES Univeristy
Bangalore, India
hiteshkumarhk.info@gmail.com

Keerthan G
PES1201700963
PES Univeristy
Bangalore, India
keerthan.g2011@gmail.com

Nahusha Acharya
PES1201700044
PES Univeristy
Bangalore, India
nahusha.acharya123@gmail.com

***Abstract*—Image inpainting is a challenging task where large missing regions have to be filled based on the available visual data. Applications of this technique include the restoration of old photographs and damaged film, removal of superimposed text like dates, subtitles, or publicity and the removal of entire objects from the image like microphones or wires in special effects.**

Existing methods which extract information from only a single image generally produce unsatisfactory results due to the lack of high level context. In this paper we have proposed a model which can fill the missing details of an image. The implementation is done using Tensorflow. For our model we have used one kind of image i.e a landscape. This can be extended onto various kinds of objects irrespective of the type if the dataset is available. We have proposed a tensorflow model which successfully predicts information in the missing pieces. Our experiment on one kind of a dataset using tensorflow has successfully predicted the missing pieces.
The following flow chart represents the areas of image inpainting.



Our model is based on the middle approach i.e the object removal but due to the unavailability of the dataset, our model fills the patches generated on the image.

I. Introduction

Image inpainting is the process of reconstructing missing parts of an image so that observers are unable to tell that these regions have undergone restoration. This technique is often used to remove unwanted objects from an image or to restore damaged portions of old photos.
Image inpainting is an ancient art that originally required human artists to do the work by hand. But today, researchers have proposed numerous automatic inpainting methods. In addition to the image, most of these methods also require as input a mask showing the regions that require inpainting. The figures show example image-inpainting results.

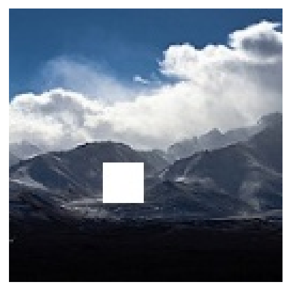


Fig. 1. Input Image

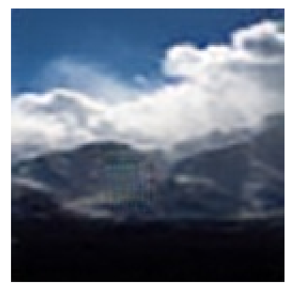


Fig. 2. Restored Image

Fig. 3. Input Image

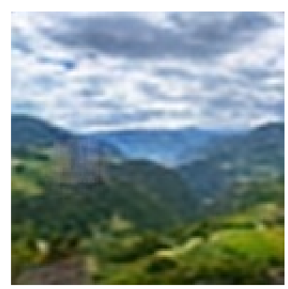


Fig. 4. Restored Image

II. Dataset

The dataset required to develop this model isn't available. We had to create the dataset. To create a set of test images, we use 128×128 images from a private collection. We then removed a 20×20 area white square at a random place on the image. To build a highly accurate model the images can be filled with different shapes and multiple patches. In this implementation we have used mountain and forest images from a private collection. Around 3K images have been generated with the patch and trained. The algorithm to generate the image dataset was developed using numpy and OpenCV python libraries.

III. Related Work

We should first note that classical image denoising algorithms do not apply to image inpainting. In common image enhancement applications, the pixels contain both information about the real data and the noise (e.g., image plus noise for additive noise), while in image inpainting, there is no significant information in the region to be inpainted. The information is mainly in the regions surrounding the areas to be inpainted. There is then a need to develop specific techniques to address these problems.In the group of disocclusion algorithms, a pioneering work is described. The authors presented a technique for removing occlusions with the goal of image segmentation. The basic idea is to connect T-junctions at the same gray-level with elastic minimizing curves. The technique was mainly developed for simple images, with only a few objects with constant gray-levels, and will not be applicable for the examples with natural images with the dataset we have built our model upon.

A close to perfect model has been developed by Nvidia. But this model has been developed using GANs. Our implementation using CNNs in tensorflow is an unique model. They have proposed the use of partial convolutions, where the convolution is masked and re-normalized to be conditioned on only valid pixels. They have further included a mechanism to automatically generate an updated mask for the next layer as part of the forward pass. Their model outperforms other methods for irregular masks. They have shown qualitative and quantitative comparisons with other methods to validate their approach.

IV. Architecture and Implementation

Our model is based on a series of convolutions followed by transposed convolution layers.
The architecture image below represents the architecture of the model we have implemented:

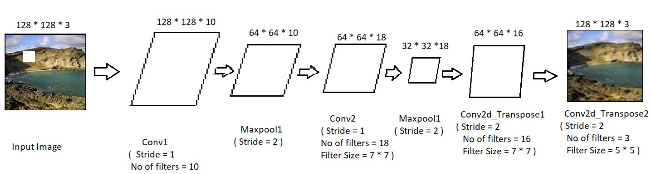


Fig. 5. Architecture of the model implemented

Input image is of shape 128*128*3 which is introduced into the first convolutional layer with 10 filters each of size 5*5 with ReLU activation function which is fed into Maxpooling layer with stride 2.

The resulting image will be of shape 64*64*10 and this will be passed to another convolutional layer with 18 filters of size 7*7 with ReLU and Maxpooling layer same with stride 2. The objective is to down-sample an input representation. The resulting image will be of shape 32*32*18.
This small sized image is then fed into the Transposed Convolution layer with stride 2 and 16 filters of size 7*7 each where the weights are trained and the resulting image will be of size 64*64*16.

The resulting image is passed to the final Transposed Convolution layer with stride 2 and 3 filters of size 5*5 which also has the corresponding weights which are trained and then we get the final output image of shape 128*128*3 along with the restored part which almost looks like the original image but without the missing part.
After the model is built with the described architecture, it is compiled with the mean absolute error loss function which is suitable when dealing with images along with the popular Adam optimiser with learning rate 0.001.

Our model is trained over 200 epochs with batch size of 32 images which is fed with the training images input and the loss training loss is calculated to be around 3.42 after the 200th epoch which is a great result as the model gave out the images in which it was almost difficult to figure out where the missing part in the image is.
Unlike other solutions like image processing methods, where they fill in the parts based on the help of surrounding pixels alone and without prior understanding of the scenario, they did not achieve great results.

GANs on the other hand need a lot of training time and also testing time because they generate new images which is also a method used by NVidia.
But our model understands the scenario in which the filling needs to take place as it has learnt to do so with the help of the trained filters which has seen the scenarios like that before and accordingly give out the output image which gave better results than other existing methods.

V. Results and Limitations

Due to unavailability of dataset on the internet, we have trained only on the images that we have generated, so if we get any large dataset for our purpose, then we will train on that and can achieve better results.

We have trained and tested on a particular type of scenario which is mountain landscapes here. In future, we will try to generalize the model to different types of scenarios.

Also, the images are a bit blurry compared to the original image due to less training time and data, which can be improved if we train with a larger dataset for more time.

The following images represent the input and the restored images which has been generated by our model:

Fig. 6. Input Image

Fig. 7. Restored Image

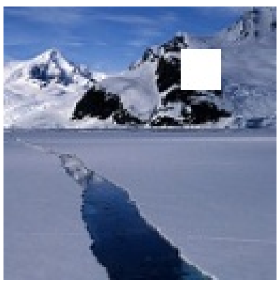


Fig. 8. Input Image

Fig. 9. Restored Image

Fig. 10. Input Image

Fig. 11. Restored Image

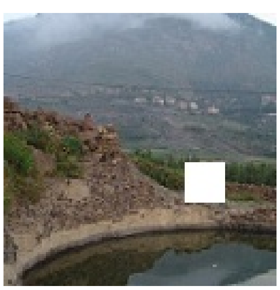


Fig. 12. Input Image

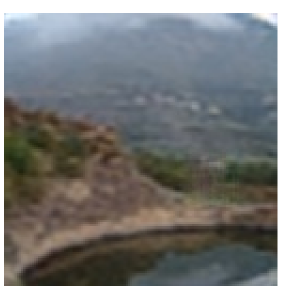


Fig. 13. Restored Image

As we can see that the images are restored almost to look like the original image and the part which is restored is almost difficult to figure out.

VI. Conclusion

In this report we have proposed a method for image-inpainting. The results obtained are great. Compared to the existing models this model is unique as it is implemented using convolutions and transposed-convolutions The results have demonstrated the superior performance of the model on challenging image inpainting examples.

VII. Refernces

http://ieeexplore.ieee.org/document/7034008/
https://towardsdatascience.com/image-inpainting-humans-vs-ai-48fc4bca7ecc
https://www.nvidia.com/research/inpainting/
https://www.cse.unr.edu/ bebis/CS474/StudentPaperPresentations/imageinpainting.pdf
https://arxiv.org/pdf/1909.06399
https://www.kaggle.com/puneet6060/intel-image-classification1000
https://www5.cs.fau.de/research/data/image-manipulation/