BLIND FACE RESTORATION

Sk Ramiz Hossain[1], Ruhul Qudse[1], Nilkantha Paramanik[1], and Somnath Malakar[1]

[1]NSHM Institute of Engineering and Technology

Abstract: During the data collecting process, images are frequently degraded. Blurring, information loss due to sampling, quantization effects, and different forms of noise are all examples of degradation. The goal of picture restoration is to correct for the flaws that cause an image to degrade.

Motion blur, noise, and camera unfocused are all examples of degradation. In some circumstances, such as motion blur, it's feasible to approximate the real blurring function and then "undo" the blur to recover the original image. When an image is contaminated by noise, the best we can hope for is to compensate for the loss of quality it has caused. We will introduce and implement numerous picture restoration methods utilized in the image processing sector in this paper. Medical imaging, astronomical imaging, forensic research, and other applications are all possible. The advantages of enhancing image quality to the greatest extent possible frequently exceed the cost and complexity of the restoration techniques required.

Keywords: GAN, Face Restoration, Visual distortion

1. Introduction

Images in digital form are essentially electronic representations of scenes, composed of pixels, typically arranged in a grid pattern, with each pixel assigned a specific value corresponding to its tone at a specific location. These images find applications in various fields, such as everyday photography, astronomy, remote sensing, microscopy, and medical imaging. Image restoration is a method used to recover damaged images by addressing issues like noise, blur, or other forms of degradation. Unfortunately, most photographs tend to have some level of haziness due to the presence of interference both from the camera and the surrounding environment.

Image restoration is the process of improving the quality of images that have been corrupted in some way. Visual distortions, such as blur and noise, can arise from factors like atmospheric turbulence, object movement, or camera misfocusing. The degradation model employed in the envision system is as follows:

$$Gk = Ik * h + n$$

Here, Gk represents the degraded image, Ik is the original image, h stands for the point spread function, and 'n' denotes the additive noise. The asterisk (*) denotes a two-dimensional linear

convolution operation. Noise, which manifests as unwanted variations in the image, can significantly impact its visual clarity. Several types of noise can affect digital images related to digital signals, with Noise types include Gaussian noise, Salt and Pepper noise (also known as Impulse noise), Uniform noise, and Brownian noise. being the most common varieties.

Image blurring can result from various factors, including motion during the image capture process, the use of wide-angle lenses, and extended exposure times. Here are some specific examples of blur:

Motion blur: Occurs when the recording equipment or the scene itself moves during the image capture process, leading to a blurred image.

Rectangular blur: Involves blurring a specific region of the image, which can be either round or rectangular and positioned in any part of the image.

Defocus blur: Arises when the camera fails to properly focus on the image, impacting the image's resolution. Greater defocus results in lower image resolution.

Gaussian blur: A type of image blurring filter that employs a Gaussian function to calculate the transformation applied to each pixel.

2. Literature Survey:

The GAN inversion method was commonly utilized in traditional restoration models. They 'invert' the damaged picture Return to a latent state of the pre-trained. GAN before reconstructing the feed images using image-specific optimization approaches. GFP-GAN, on the other hand, employs sensitive designs to achieve a decent combination of realism and fidelity in a single image processing forward pass. GFPGAN consists of multiple Layers of Channel-Split Spatial Feature Transform (CS-SFT) arranged in a progression from coarse to fine. Additionally, it incorporates a module dedicated to removing degradation and a pretrained face GAN serving as a facial feature capturer. These components are all linked together through a direct mapping of latent codes.

The CS-SFT layers conduct spatial modulation on a subset of features while allowing the remaining characteristics to flow through directly for improved information preservation. This allows it to effectively incorporate generative features while retraining the original image's high fidelity. It also includes facial component loss with local discriminators to increase picture fidelity by enhancing perceptual facial details in the image while stressing identity preservation.

Many traditional facial landmark detection methods have been proposed in the literature during the

last few decades. Active Appearance Models (AAMs), Constrained Local Models (CLMs), and Cascaded Regression are examples of parameterized appearance models. They finish the job by increasing the image's confidence in the part position. AAMs and their descendants try to model global holistic appearance and shape together, but CLMs and variants learn a set of local experts by applying various shape constraints. The main operation in Cascaded Regression is vector addition, which is efficient and has a low computational complexity.

In traditional GAN, G employs Xr as an input and produces the composite picture Xs in classic GAN, whereas D accepts them as input regardless of whether they are actual or created images. D and G compete during the training phase, with D maximizing classification accuracy and G.

Attempting to synthesis high-quality images to diminish D's classification accuracy. When D is unable to tell whether the image generated by G is from the real sample, it will converge, indicating that the true and false domain image quality is very similar.

When picture information is absent, i.e., pixel information is insufficient, the traditional method is incapable of creating something from nothing and restoring the lost image in a reasonable manner.

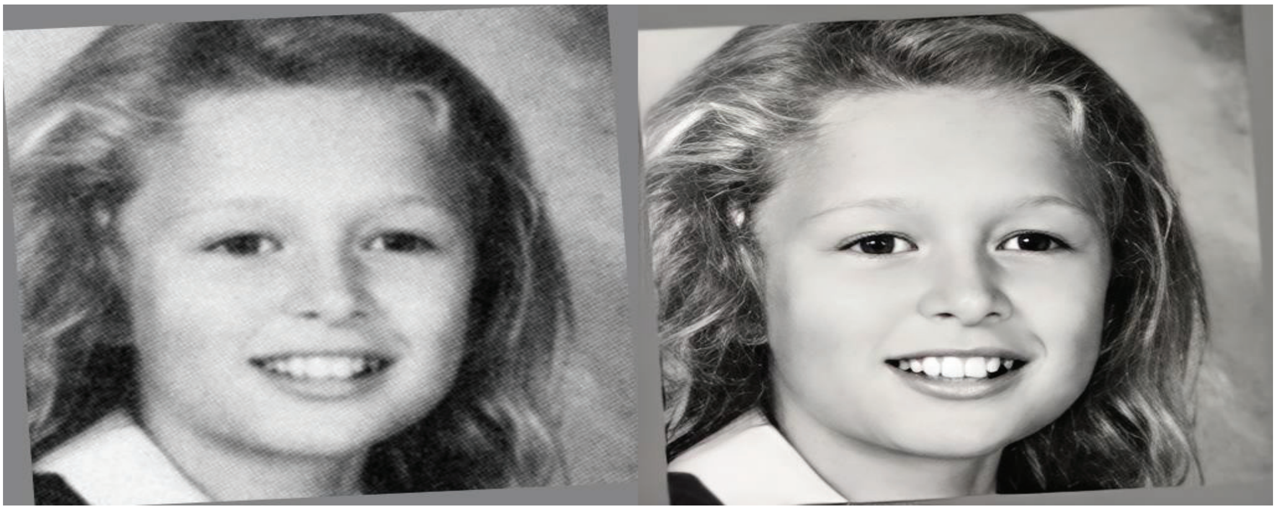
3. The challenge of restoring facial images and photos with artificial intelligence

Image restoration should not be conflated with digital retouching alone. While these two processes share similarities, their techniques can differ significantly. The former is generally more objective, while the latter is subjective in nature. Digital retouching aims to enhance the visual appeal of an image, whereas restoration involves reversing known degrading operations applied to images to make these degradations no longer visible.

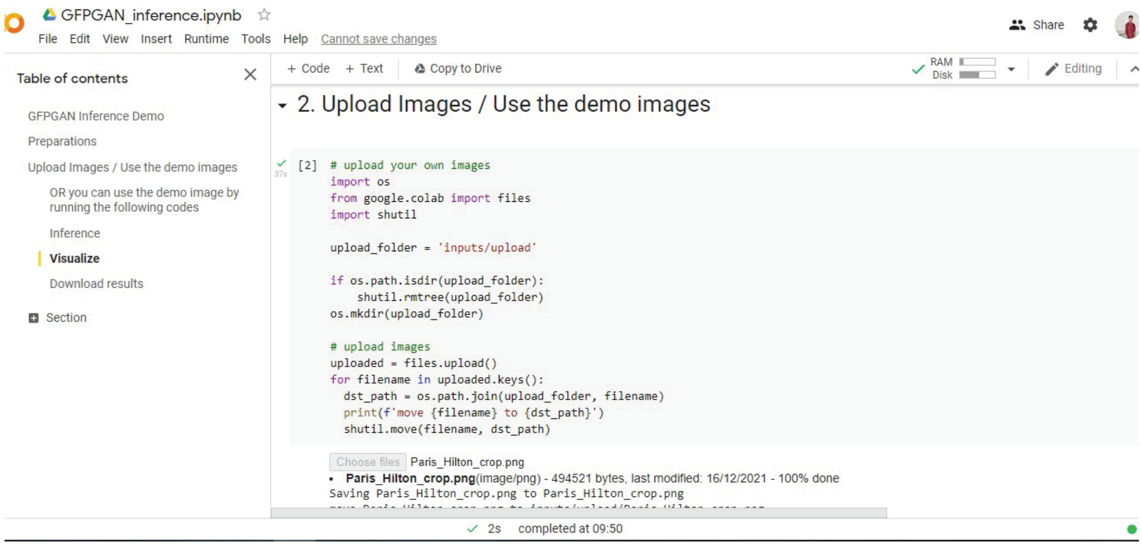
In the context of facial restoration, several factors, such as facial geometry, typically play a crucial role. However, the frequently poor quality of input images often poses challenges when applying techniques related to facial geometry. This limitation restricts the scope of restoration applications. To address this issue, a research team has introduced GFP-GAN, which leverages alternative features and techniques, powered by artificial intelligence, to effectively restore photographs featuring faces.

4. Analysis of Test Results and Screenshots

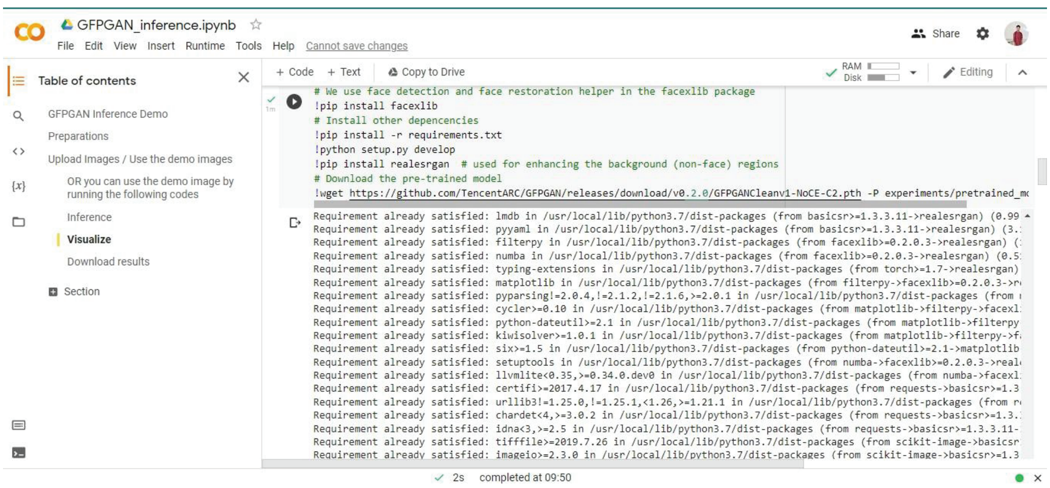
The difference between input and output is clearly visible:



Here we can see that the output image is recovered from its degraded input image. The noise, blur is much less in the output image.


GFPGAN_inference.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
Table of contents
GFPGAN Inference Demo
Preparations
Upload Images / Use the demo images
OR you can use the demo image by running the following codes
Inference
Visualize
Download results
Section
+ Code + Text Copy to Drive
RAM
Disk
Editing
2. Upload Images / Use the demo images
[2] # upload your own images
import os
from google.colab import files
import shutil
upload_folder = 'inputs/upload'
if os.path.isdir(upload_folder):
shutil.rmtree(upload_folder)
os.mkdir(upload_folder)
# upload images
uploaded = files.upload()
for filename in uploaded.keys():
dst_path = os.path.join(upload_folder, filename)
print(f'move {filename} to {dst_path}')
shutil.move(filename, dst_path)
Choose files Paris_Hilton_crop.png
Paris_Hilton_crop.png(image/png) - 494521 bytes, last modified: 16/12/2021 - 100% done
Saving Paris_Hilton_crop.png to Paris_Hilton_crop.png
2s completed at 09:50


GFPGAN_inference.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
Table of contents
GFPGAN Inference Demo
Preparations
Upload Images / Use the demo images
OR you can use the demo image by running the following codes
Inference
Visualize
Download results
Section
+ Code + Text Copy to Drive
RAM
Disk
Editing
OR you can use the demo image by running the following codes
import shutil
import os
upload_folder = 'inputs/upload'
if os.path.isdir(upload_folder):
shutil.rmtree(upload_folder)
os.makedirs(upload_folder, exist_ok=True)
shutil.move('inputs/whole_imgs/Blake_Lively.jpg', 'inputs/upload/Blake_Lively.jpg')
'inputs/upload/Blake_Lively.jpg'
3. Inference
[3] # Now we use the GFPGAN to restore the above low-quality images
We use [Real-ESRGAN](https://github.com/xinntao/Real-ESRGAN) for enhancing the background (non-face) regions
!rm -rf results
!python inference_gfpgan.py --upscale 2 --test_path inputs/upload --save_root results --model_path experiments/pretrained
2s completed at 09:50


GFPGAN_inference.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
Table of contents
GFPGAN Inference Demo
Preparations
Upload Images / Use the demo images
OR you can use the demo image by running the following codes
Inference
Visualize
Download results
Section
+ Code + Text Copy to Drive
RAM
Disk
Editing
We use face detection and face restoration helper in the facexlib package
!pip install facexlib
Install other depencencies
!pip install -r requirements.txt
!python setup.py develop
!pip install realesrgan # used for enhancing the background (non-face) regions
Download the pre-trained model
!wget https://github.com/TencentARC/GFPGAN/releases/download/v0.2.0/GFPGANCleanv1-NoCE-C2.pth -P experiments/pretrained_mc
Requirement already satisfied: lmdb in /usr/local/lib/python3.7/dist-packages (from basicsr>=1.3.3.11->realesrgan) (0.99
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from basicsr>=1.3.3.11->realesrgan)
Requirement already satisfied: filterpy in /usr/local/lib/python3.7/dist-packages (from facexlib>=0.2.0.3->realesrgan)
Requirement already satisfied: numba in /usr/local/lib/python3.7/dist-packages (from facexlib>=0.2.0.3->realesrgan)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from torch>=1.7->realesrgan)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from filterpy->facexlib>=0.2.0.3->
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->filterpy->facexl
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->filterpy
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->filterpy->f
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.1->matplotlib
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from numba->facexlib>=0.2.0.3->real
Requirement already satisfied: llvmlite<0.35,>=0.34.0.dev0 in /usr/local/lib/python3.7/dist-packages (from numba->facexl
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->basicsr>=1.3
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->basicsr>=1.3.
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->basicsr>=1.3.3.11-
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-packages (from scikit-image->basicsr
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->basicsr>=1.3
2s completed at 09:50

5. Conclusion

Restoration procedures are employed to bring a corrupted image back to its pristine condition, thereby enhancing image quality. Blind image restoration involves two primary approaches: direct measurement and indirect estimation. In the first approach, the unknown characteristics of the system, such as the blur impulse response and noise level, are directly measured from the image slated for recovery. These measured parameters are subsequently used in the restoration process.

On the other hand, indirect estimation methods rely on approaches that either derive the restoration itself or determine crucial aspects of the restoration procedure. This discussion aims to introduce you to restoration strategies rooted in noise and blur models, offering insights into effective image restoration techniques.

References

1. Boracchi, G., & Foi, A. (2012). Modeling the performance of image restoration from motion blur. IEEE Transactions on Image Processing, 21(8), 3502-3517.

2. M. Tico and M. Vehvilainen, "Estimation of motion blur point spread function from differently exposed image frames," 2006 14th European Signal Processing Conference, Florence, Italy, 2006, pp. 1-4..

3. Yuan, L., Sun, J., Quan, L., & Shum, H. Y. (2007). Image deblurring with blurred/noisy image pairs. In ACM SIGGRAPH 2007 papers (pp. 1-es).

4. T. F. Chan and Chiu-Kwong Wong, "Total variation blind deconvolution," in IEEE Transactions on Image Processing, vol. 7, no. 3, pp. 370-375, March 1998, doi: 10.1109/83.661187.

5. Li, X., Li, W., Ren, D., Zhang, H., Wang, M., & Zuo, W. (2020). Enhanced blind face restoration with multi-exemplar images and adaptive spatial feature fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2706-2715).

6. Xinqiao Liu and A. El Gamal, "Synthesis of high dynamic range motion blur free image from multiple captures," in IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 50, no. 4, pp. 530-539, April 2003, doi: 10.1109/TCSI.2003.809815.