Assignment 4: Image Inpainting

Yan Zhao

Due date: June 30, 2018

1 Introduction

Most of you will have some old degraded photos at your home with some black spots, some strokes etc on it. Have you ever thought of restoring it back? We can't simply erase them in a paint tool because it is will simply replace black structures with white structures which is of no use. In these cases, a technique called image inpainting is used. The basic idea is simple: Replace those bad marks with its neighbouring pixels so that it looks like the neighbourhood. You can learn more here: Wikipedia.

In this assignment, you will implement three tasks for image inpainting and compare the performance of different inpainting methods by using two evaluation measures. The more tasks you accomplish, the higher score you will get.

2 Dataset

The dataset you use is the CelebA dataset, CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. CelebA has large diversities, large quantities, and rich annotations, and you can learn more and download here: CelebA.

The CelebA dataset is very large, it is very slow to process on the CPU. So let's make our own dataset!

The whole process is divided into the following steps:

- Download the CelebA dataset first, you can choose to download jpg format or png format. It is recommended to download jpg format. Download link: CelebA;
- Select 2064 images with clear faces (non-side faces) from the dataset you download in the previous step;
- Preprocess the selected 2064 images, select an appropriate program, extract face area and resize each image to 64*64. In order to make it easier for everyone to understand, you can see the process in the Figure 1;

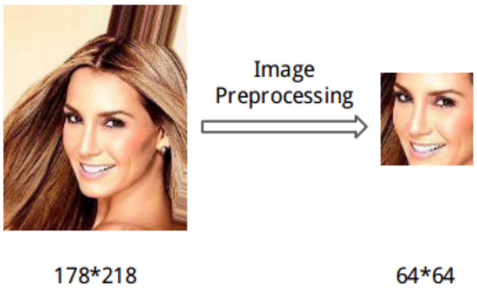

Figure 1: Image preprocessing

- Divide your images (2064 images) into two parts: 2000 images as the training set and 64 images as the test set. Named training set and test set respectively;
- Create a folder named celeba_your name. The folder contains your training set, test set, and your code for image processing.

Having done make your own dataset, let's start our task of image inpainting!

3 Task 1: OpenCV-Image Inpainting

In this task, you will implement image inpainting using OpenCV. OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. You can learn more and download OpenCV here: OpenCV.

- We will learn how to remove small noises, strokes etc in photos by a method called inpainting.
- We will see inpainting functionalities in OpenCV.

You can learn more here: OpenCV-Image Inpainting.

You will use OpenCV to implement image inpainting. You can choose any programming language to implement this task, including Python, MATLAB, C or C++, etc.

The dataset used in this task is your test set. From your test set (64 images), select 10 images randomly, use OpenCV for image inpainting, and use PSNR and SSIM for image quality evaluation.

Put your code, datasets, experiment results, and evaluation results into a folder named task1_your name.

4 Task 2: Semantic Image Inpainting - I

In this task, you will use deep learning to perform simple semantic image inpainting tasks. This method was proposed by paper **"Semantic image inpainting with deep generative models"**. You can learn more about this paper here: Semantic Image Inpainting.

Task 2 is divided into five steps:

- Read the paper to understand the process of semantic image inpainting;
- Download the source code for the semantic image inpainting here: Semantic Image Inpainting;
- Use the model provided by the author dcgan-100.pb to test your own test set. If the above link is not accessable, you could download the dataset using Baidu Drive. The retrieve password is javb;
- When testing the model, at least test 3 mask types, and use PSNR and SSIM for image quality evaluation;
- Put your code, datasets, experiment results, and evaluation results into a folder named task2_your name.

You have completed a simple image inpainting task. Let's do something more challenging!

5 Task 3: Semantic Image Inpainting - II

The third task has been upgraded! You need to personally train your own semantic image inpainting model! Instead of using other trained models to test the experimental results!

Task 3 is divided into four steps:

- Read the paper to understand the process of semantic image inpainting;
- Use your own training set (2000 images) to train semantic image inpainting based on DCGAN, you can use Tensorflow or PyTroch to implement this task;
- Choose at least 3 mask types to test your model, and use PSNR and SSIM for image quality evaluation;
- Put your code, datasets, experiment results, and evaluation results into a folder named task3_your name.

Congratulations on completing all your image inpainting tasks!

6 Evaluation

In this part, you will compare the quality of image inpainting methods using two evaluation measures.

- Peak Signal to Noise Ratio (PSNR)
- Structural Similarity index (SSIM)

Compare all inpainting results on each evaluation measure. You can choose Python or MATLAB to implement these two evalution measures. Put your code into a folder named evaluation_your name.

7 Submission and Grading

After various parts of the assignment are completed, the following files including

1. Your datasets.
2. Your codes.
3. Your results.
4. A PDF report containing your results and the analysis of your experiments.

Zip all your files and submit your assignment to ouceecv@163.com with the subject: YourName_Assignment4.zip. The name of your zip file should be the same as the email subject.

Be sure to finish and submit Assignment 4 before the due date June 30, 2018. Then we will grade your assignment based on the files your submitted. The following is a breakdown of how each part of this assignment is scored:

Part	Points
Task 1	30 points
Task 2	30 points
Task 3	30 points
PSNR	5 points
SSIM	5 points
Total points	100 points

If you have any questions, you can contact zhaoyan4899@stu.ouc.edu.cn.