Image Restoration Web-App

Anish Pandita - 90688024 | Gurkanwal Singh Kang - 93887116 |
Mayank Chandak - 86041404 | Neelendu Wadhwa - 92966132

CS 295P, Lecture A, Professor Harris
Winter Quarter 2022

1. **Introduction** –

Have any of these thoughts, “I wish I had better quality photos of my grandparents as memories” or “What if there could be a way to make my blurry vacation photos clearer and less pixelated” crossed your mind. If yes, then our Image Restoration Web-App is the go-to solution. From being able to bring fresh life to old, bad quality images to applying style transfers to make the images more creative with a sense of modern style and aesthetics, our application has a plethora of use cases. There are similar products in the market but none offer functionalities like image restoration, style transfer, and storage altogether. There is no one-stop solution. Even if they do try to bundle up everything, the result is a big hole in your pocket as these applications cost lots of bucks plus who can forget those annoying advertisements.

Our open-source implementation with revolutionary architecture and cutting-edge design built on Deep convolutional networks is the backbone for the image generation and restoration process. We use them because their excellent performance is imputed to their ability to find realistic image priors from an outsized number of example images. Generally speaking, image restoration is the operation of taking a corrupted image and estimating the initial image, which is understood to be an ill-posed inverse problem.

Recently, deep neural networks (DNNs) have shown their superior performance in image processing and computer vision tasks, starting from high-level recognition, semantic segmentation to low-level denoising and superresolution. One of the first deep learning models which are used for image denoising is the Stacked Denoising Auto-encoders.

Denoising auto-encoders are often stacked to make a deep network by feeding the output of the previous layer to the present layer as input.

2. **Features** –
    1) **Image Restoration:**
    This is the core feature of our platform. The platform will provide the users with the functionality to upload a corrupted image and the image will be processed by the image processing module which consists of machine learning models[1] and the transformed image would be displayed to the user. Here corrupted image refers to an image that could be having a low resolution or an image having some missing regions.

    2) **Style Transfer:**
    The platform will also provide the users with the feature of applying the style of one image to another. The user would be able to upload two separate images, of which one would be used as a styling image and another would be the one on which the

style is applied.[2] The user would also be able to choose previously uploaded images for styling from the dashboard.

3) **Black and white to a colored image:**
The platform will also provide the feature of allowing the users to upload black and white images and transform them into colored images.[3]

4) **Storage:**
The images uploaded by the user would be stored in the database and the user would be provided with a dashboard where the user can view the uploaded images and the restored images. All the uploaded images would be associated with a user account and the user would be able to view the images present on their account.

5) **Authentication (Privacy):**
The user would be able to create a profile and credentials will be associated with the user's account through which the user can log in to the platform and view his/her data.

6) **Dashboard:**
The user would be provided with a central dashboard from where he would be able to make all the core requests to the several services and functionalities provided by the platform i.e. view uploaded images, restored images, and make requests to restore images.

7) **Download images:**
The user would also be provided with the feature to download the images that are uploaded by him and also the restored images processed by the platform and associated with his account.

3. **Interface** –

1) **Login/Registration Screen:**
The user would be provided with text inputs where the user would be prompted to enter the credentials. In case the user is not already registered on the platform, then the user would also be provided with the option to register. On an unsuccessful login attempt, the user would be redirected back to this screen with the appropriate error message.

2) **Registration Screen:**
The user would be prompted to enter the registration details on this screen.

3) **Central Dashboard:**
Upon successful registration, the user would be redirected to a central dashboard where the user would be able to use the primary features of the platform. The central dashboard would provide the user with buttons using which the user can make requests for uploading images, restoring images, applying style transfer to images, along transforming a black and white image into a colored image.

The user would also be provided with the facility to store the uploaded images in the Database which can be requested through a button provided in the central dashboard.

4) **View Previous Images Screen:**
In this screen, the user would be able to view the previously uploaded images and request for restoration of the images using a button that would be provided in the interface. Upon clicking the button the user would be able to select multiple images for restoration.

5) **View Restored/Transformed Images Screen:**
Through this screen, the user would be able to view the images that the user had restored previously.

4. **Architecture –**

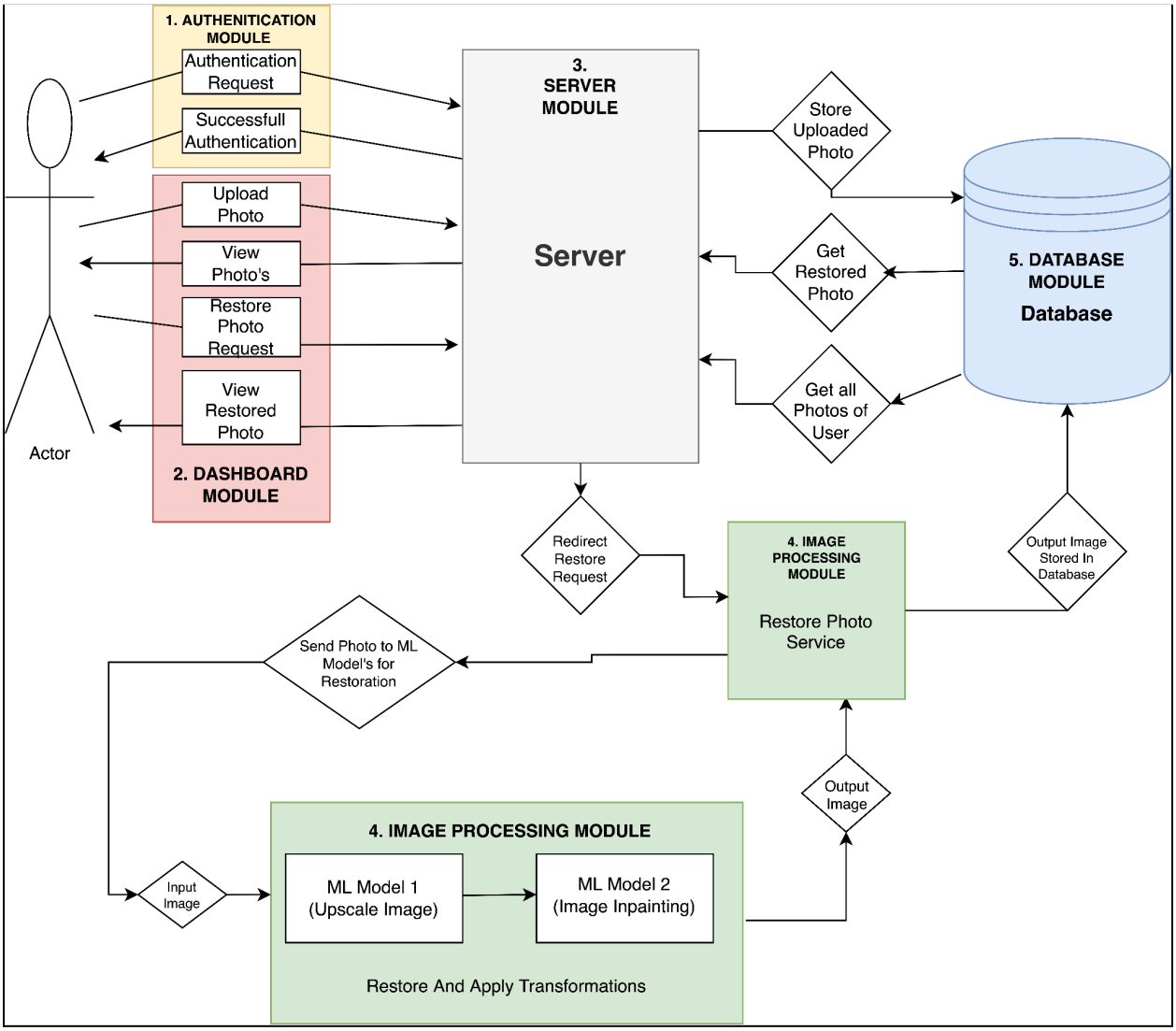


Figure 1: Architectural Diagram

The Architectural Diagram is shown in Figure 1 and here are the descriptions of each block:

1) **Authentication Module:**
   This block would be responsible to carry out the functionality of User Authentication and a separate code block would be enough to carry out this task. We would authenticate the user by parsing encoded credentials with a request header and verifying the details using a serializer and database queries.
   The user would be able to enter the credentials here and authorization would be granted to the user based on the credentials entered. The user would be redirected to the DashBoard Block from here.

2) **Dashboard Module:**
   After the verification of credentials of the user through the authentication module. The user would be redirected to the dashboard module where the user would be able to use the core functionality of the platform including -
   - Viewing the old Images
   - Viewing the restored images
   - Request for restoration of the images.
   - Upload the Images to the platform.

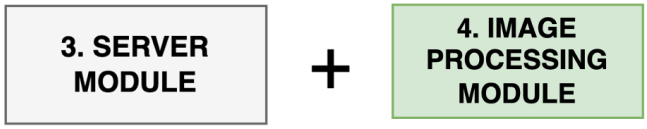
   This block represents the portions of the dashboard from where the user would be able to make requests to view their previously-stored photos, restore their uploaded photos and upload new photos. These API requests are redirected to the Server Block, which processes them and redirects them to the required modules/blocks.

3) **Server Module (Controller):**
   This module is where the core logic of request routing and functionality of components is defined. This module redirects the request for restoration of images by the user to the Image processing module and redirects the data (restored image) received from the Image Processing module to the database. It also consists of logic for fetching the restored image from the database and displaying it to the user on the dashboard.
   This block represents the portion of the web application where the logic for routing the requests of the user is defined and the logic for fetching and processing data from the database on the basis of the request of the user is defined.

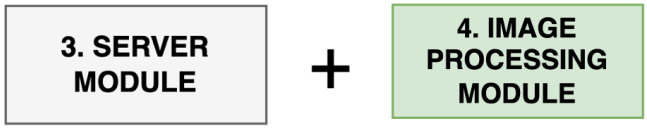
4) **Image Processing Module:**
   This module consists of the Machine Learning models which are given the input of the image uploaded by the user and apply convolution and other operations on the image and the predicted output from the models is then sent to the server which is then stored in the database.

5) **Database Module:**
   Here the models and relationships between the different attributes of the models are defined and it also consists of logic for serializing the data and storing the received images into cloud-hosted S3 buckets, which would allow the application to use commoditized computer resources and scale as per the requirements of the user. Block 8 represents the database module.

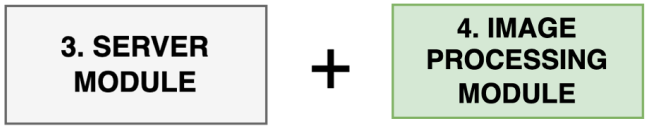
5. **Feature Map** –
    1. **Image Restoration:**
    This feature would be making use of the following modules from the Architecture diagram (Figure 1).



    2. **Style Transfer:**
    This feature would be making use of the following modules from the Architecture diagram (Figure 1).



    3. **Black and white to a colored image:**
    This feature would be making use of the following modules from the Architecture diagram (Figure 1).



    4. **Storage:**
    This feature would be making use of the following modules from the Architecture diagram (Figure 1).

    5. **Authentication (Privacy):**
    This feature would be making use of the following modules from the Architecture diagram (Figure 1).

1. AUTHENITICATION MODULE + 3. SERVER MODULE + 5. DATABASE MODULE

6. **Dashboard:**
   This feature would be making use of the following modules from the Architecture diagram (Figure 1).

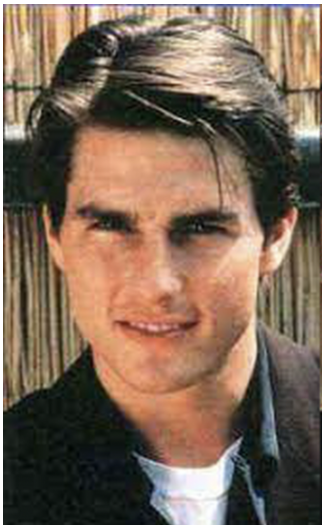
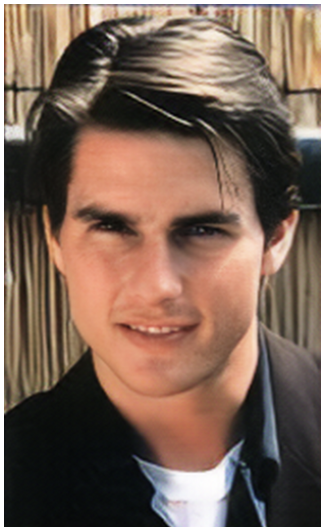
2. DASHBOARD MODULE + 3. SERVER MODULE

7. **Download images:**
   This feature would be making use of the following modules from the Architecture diagram (Figure 1).

2. DASHBOARD MODULE + 3. SERVER MODULE + 5. DATABASE MODULE

6. **Output** –

Before: **After**:

7. **References** –

[1] GAN — What is Generative Adversarial Networks GAN? | by Jonathan Hui | Medium
[2] Image Style Transfer Using Convolutional Neural Networks
[3] Colorizing black & white images with U-Net and conditional GAN — A Tutorial | Towards Data Science