**HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH QUALITY TRAINING**

**INFORMATION TECHNOLOGY**

□□🕮□□

# DIGITAL IMAGE PROCESSING

# DEVELOPING TOOL FOR IMAGE ENHANCEMENT AND IMAGE FILTER IN FREQUENCY DOMAIN

**Lecturer: Hoang Van Dung**

**Class: DIPR430685E_22_2_01CLC**

**Group: 02**

**Members:**

| | |
|---|---|
| Vo Minh Hung | 20110426 |
| Nguyen Thanh Sang | 20110393 |
| Tran Anh Tai | 20110397 |

*Ho Chi Minh City, May 2023*

| Student ID | Student name | Contribution (%) |
|---|---|---|
| 20110426 | Vo Minh Hung | 100% |
| 20110393 | Nguyen Thanh Sang | 100% |
| 20110397 | Tran Anh Tai | 100% |

## Project contribution and work table

| Student Name | Work |
|---|---|
| Vo Minh Hung | • Establish project goals and objectives<br>• Design code<br>• Evaluation and Bug Fixing |
| Nguyen Thanh Sang | • Establish project goals and objectives<br>• Design code<br>• Survey applications |
| Tran Anh Tai | • Establish project goals and objectives<br>• Evaluation and Bug Fixing<br>• Write report |

## SCORE

| Criteria | Content | Presentation | Total |
|---|---|---|---|
| Point | | | |

## REMARKS OF TEACHERS

……

……

……

……

……

……

……

……

……

……

……

……

……

Ho Chi Minh City, May 2023

Teacher's score

Signature and full name

**Hoang Van Dung**

## Table of Contents

# CHAPTER 1: PROJECT DESCRIPTIONS

## 1.1. Introduction

In this project we will use Development tool for image enhancement and image filter in frequency domain.by utilizing the frequency domain, this tool provides an efficient and effective way to process images with complex patterns and structures, resulting in high-quality images suitable for a wide range of applications such as medical imaging, satellite imagery, and digital photography. With its intuitive user interface and robust features, this tool is an essential resource for anyone who needs to process and enhance digital images. From the initial conceptualization to the final testing and installation of the program, this report will provide a detailed overview of the project's lifecycle and key components.

## 1.2. Reason choosing the topic

The project of developing a tool for image enhancement and image filtering in the frequency domain was chosen due to its significance in digital image processing. With the increasing use of digital images in various fields, the need for efficient and effective image processing tools has become essential.

This project aims to provide a tool that can enhance the quality of digital images by applying frequency domain filters and techniques. By utilizing frequency domain filters, this tool can achieve better results compared to traditional spatial domain filtering methods.

## 1.3. Scope and objects

Scope: Ho Chi Minh city, knowledge from course and online research.

Objects: Some digital pictures of landscapes, people, animals,...

## 1.4. Survey some photo editing applications

### 1.4.1. Adobe Photoshop

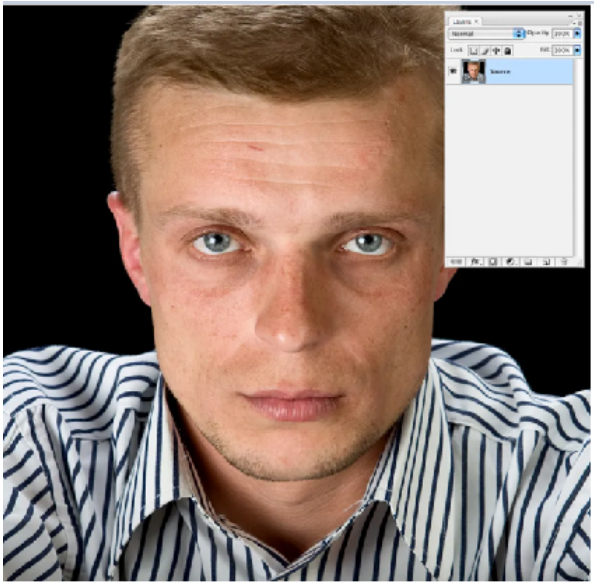
Adobe Photoshop or PS for short, is a professional image and graphics editing software used by many designers and photographers to enhance colors or design media publications for the company.

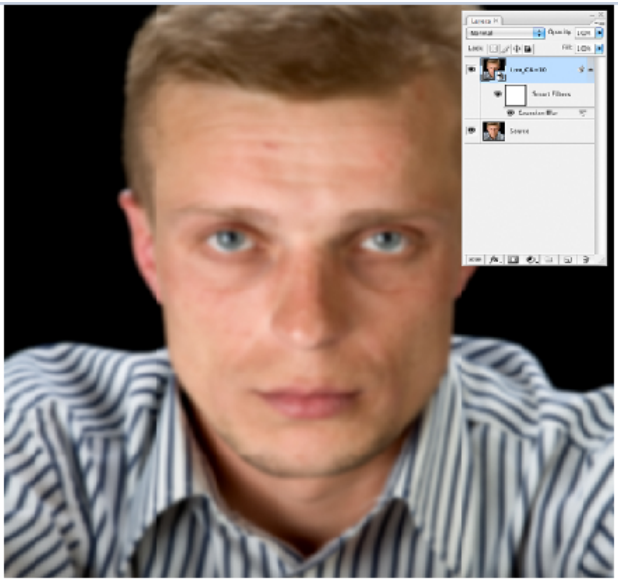
Photoshop can be used to edit digital images, create new ones, modify existing images, and create other types of artwork and graphics. Its features include crop, paste, flip, rotate, adjust brightness, contrast, resolution, color, filter, etc.

Some Photoshop filters:

+ ***Gaussian Blur:***

- The low-pass filter in Photoshop is Gaussian Blur. The larger the radius, the lower the frequency it leaves. An additional high-pass filter is High Pass.





+ ***High pass:***

- The High Pass filter in Photoshop is used to sharpen images and retouch to bring back the lost details.

+ **Motion blur filter:**

- Motion blur filter is used to add motion to photos. You can add movement to cars, bikes, and objects.


### 1.4.2. GIMP

GIMP (/gɪmp/ GHIMP), short for GNU Image Manipulation Program, is a free and open source software used for image editing, freehand drawing, converting between different image formats, and more specialized tasks.

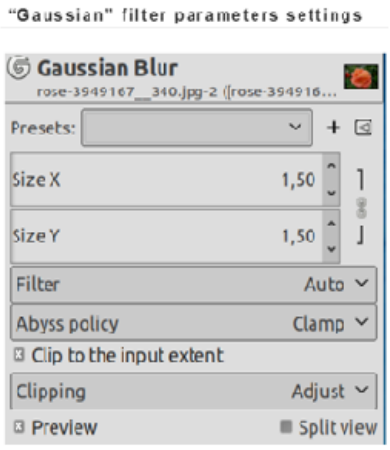
GIMP provides users with a range of tools and features for editing and processing digital images. These include color adjustment, color balance, smoothing and sharpening images, cropping and resizing images, among other functions.

+ ***Gaussian Blur***

- This is the most popular filter in Gimp because it has so many uses, like creating a common blur for your layers or being used to blend certain effects


Oprindelig


Blur applied



- The Gaussian Blur plug-in acts on each pixel of the active layer or selection, setting its Value to the average of all pixel Values present in a radius defined in the dialog. A higher Value will produce a higher amount of blur

***+ High Pass***

- In digital images, frequency refers to sudden changes in brightness or color in neighboring pixels. The High Pass filter filters high essential details, and larger scale gradients are removed. This result, combined with the original image and "Soft Light" or "Hard Light" merge mode, is used to sharpen images. It enhances fine details.

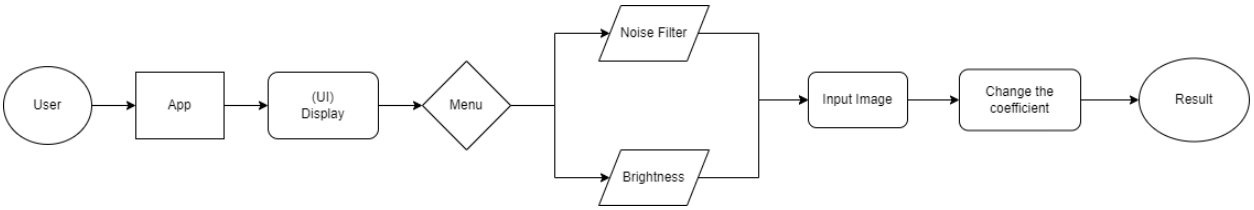
# CHAPTER 2: BACKGROUND KNOWLEDGES

We use Spider (anaconda) which use Python language to code the program. The following image is the list libraries used in the project

```
import cv2
import numpy as np
import tkinter as tk
from tkinter import filedialog
from PIL import Image, ImageTk
```

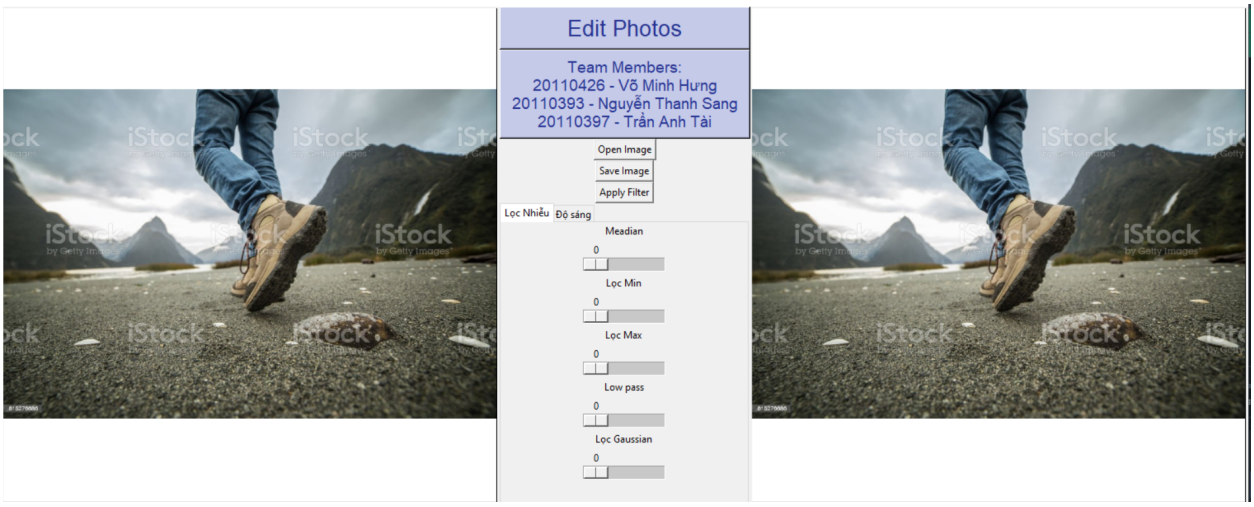
*List of used libraries*

- **cv2 (OpenCV):** image and video processing library, providing functions for reading, writing, converting and processing images.
- **numpy**: Library of arithmetic computation on multidimensional arrays (ARRAYs), widely used in image processing and machine learning.
- **tkinter**: library that allows creating graphical interfaces (GUIs) in Python. It provides objects such as windows, buttons, text, and images to build GUI applications .filedialog from TKINTER: allows to open a file search dialog on the system.
- **PIL (Python Imaging Library):** library that allows image processing, such as cropping, rotating, changing image resolution and format. This library is used to convert image objects of OpenCV to the format displayed in the GUI of tkinter.




*General block diagram*

# CHAPTER 3: INTERFACES



# CHAPTER 4: FUNCTION

## 4.1. Photo Management

### 4.1.1. Open image

- Choose a photo to edit.

```
def open_image(self):
    self.img_path = filedialog.askopenfilename()
    if self.img_path:
        self.img = cv2.imread(self.img_path)
        self.processed_img = np.copy(self.img)
        self.show_image_original()
        self.show_image_processed()
        self.reset_sliders()
```

### 4.1.2. Apply filter

- Apply filter to the image.

```
def apply_filter(self):
    # Lưu trữ ảnh đã được chỉnh sửa của filter cũ vào biến previous_processed_img
    self.previous_processed_img = np.copy(self.processed_img)
```

### 4.1.3. Save image

- After editing is complete, we can save the edited image.

```
def save_image(self):
    if self.processed_img is not None:
        save_path = filedialog.asksaveasfilename(defaultextension=".jpg")
        if save_path:
            cv2.imwrite(save_path, self.processed_img)
```

## 4.2. Noise Filter

### 4.2.1. Median filter

```
def process_image_median(self, value):
    if self.img is not None:
        kernel_size = int(value) // 2 * 2 + 1
        if self.previous_processed_img is not None:
            img_median = cv2.medianBlur(self.previous_processed_img, kernel_size)
        else:
            img_median = cv2.medianBlur(self.img, kernel_size)
        self.processed_img = img_median
        self.show_image_processed()
```

### 4.2.2. Min filter

```
def process_image_min(self, value):
    if self.img is not None:
        kernel_size = int(value) // 2 * 2 + 1
        if self.previous_processed_img is not None:
            img_min = cv2.erode(self.previous_processed_img, np.ones((kernel_size, kernel_size), np.uint8))
        else:
            img_min = cv2.erode(self.img, np.ones((kernel_size, kernel_size), np.uint8))
        self.processed_img = img_min
        self.show_image_processed()
```

### 4.2.3. Max filter

```
def process_image_max(self, value):
    if self.img is not None:
        kernel_size = int(value) // 2 * 2 + 1
        if self.previous_processed_img is not None:
            img_max = cv2.dilate(self.previous_processed_img, np.ones((kernel_size, kernel_size), np.uint8))
        else:
            img_max = cv2.dilate(self.img, np.ones((kernel_size, kernel_size), np.uint8))
        self.processed_img = img_max
        self.show_image_processed()
```

### 4.2.4. Low pass filter

```
def process_image_lowpass(self, value):
    if self.img is not None:
        kernel_size = int(value) // 2 * 2 + 1
        if self.previous_processed_img is not None:
            img_lowpass = cv2.GaussianBlur(self.previous_processed_img, (kernel_size, kernel_size), 0)
        else:
            img_lowpass = cv2.GaussianBlur(self.img, (kernel_size, kernel_size), 0)
        self.processed_img = img_lowpass
        self.show_image_processed()
```

### 4.2.5. Gaussian filter

```
def process_image_gaussian(self, value):
    if self.img is not None:
        kernel_size = int(value) // 2 * 2 + 1
        img_gaussian = cv2.GaussianBlur(self.img, (kernel_size, kernel_size), 0)
        self.processed_img = img_gaussian
        self.show_image_processed()
```

### 4.3. Brightness

#### 4.3.1. Power law.

```
def adjust_power_law_transform(self, value):
    if self.img is not None:
        power_param = 1.0 + float(value) / 100.0
        if self.previous_processed_img is not None:
            img_power_law = np.power(self.previous_processed_img, power_param)
        else:
            img_power_law = np.power(self.img, power_param)
        img_power_law = (img_power_law / np.max(img_power_law)) * 255
        img_power_law = np.array(img_power_law, dtype=np.uint8)
        self.processed_img = img_power_law
        self.show_image_processed()
```

#### 4.3.2. Histogram

```
def adjust_histogram_equalization(self, value):
    if self.img is not None:
        hist_eq_param = int(value) * 255 // 100
        if self.previous_processed_img is not None:
            img_gray = cv2.cvtColor(self.previous_processed_img, cv2.COLOR_BGR2GRAY)
        else:
            img_gray = cv2.cvtColor(self.img, cv2.COLOR_BGR2GRAY)
        img_hist_eq = cv2.equalizeHist(img_gray)
        img_hist_eq = cv2.cvtColor(img_hist_eq, cv2.COLOR_GRAY2RGB)
        img_hist_eq = np.array(img_hist_eq, dtype=np.uint8)
        img_hist_eq = cv2.addWeighted(self.img, 0.7, img_hist_eq, 0.3, 0)
        self.processed_img = img_hist_eq
        self.show_image_processed()
```

#### 4.3.3. Log transform

```
def adjust_log_transform(self, value):
    if self.img is not None:
        log_param = float(value) / 100.0
        if self.previous_processed_img is not None:
            img_log = np.log1p(self.previous_processed_img * log_param)
        else:
            img_log = np.log1p(self.img * log_param)
        img_log = (img_log / np.max(img_log)) * 255
        img_log = np.array(img_log, dtype=np.uint8)
        self.processed_img = img_log
        self.show_image_processed()
```

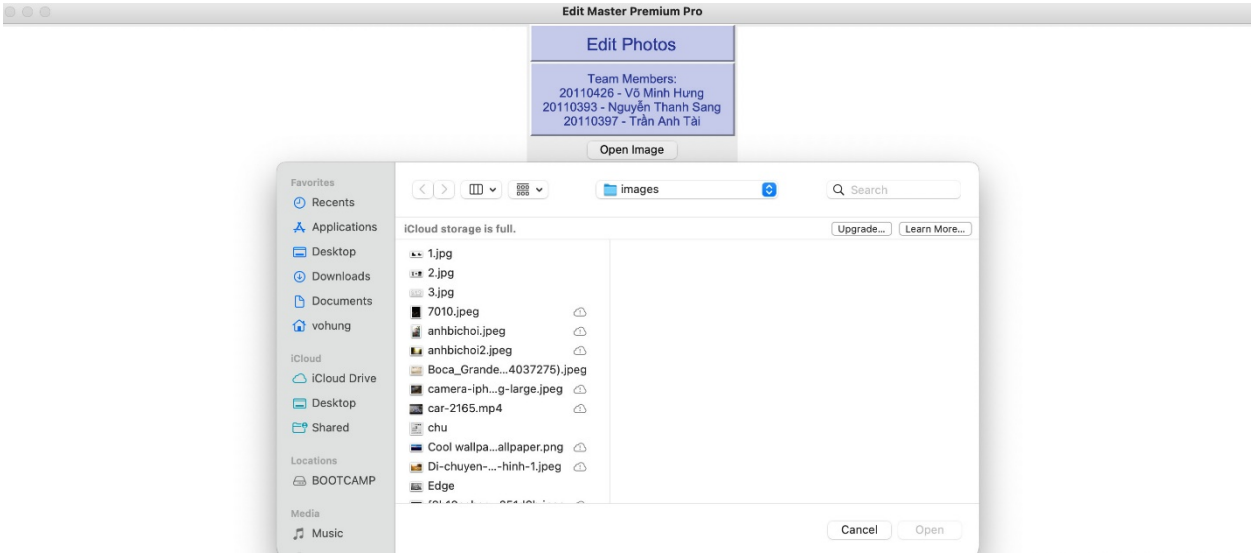
### 4.3.4. Brightness Adjustment

```
def process_image_brightness(self, value):
    if self.img is not None:
        img_brightness = cv2.convertScaleAbs(self.img, alpha=float(value))
        img_blend = cv2.addWeighted(self.img, 0.5, img_brightness, 0.5, 0)
        self.processed_img = img_blend
        self.show_image_processed()
```

# CHAPTER 5: DEMO

- Main interface


- Select photos to edit



- Show original image

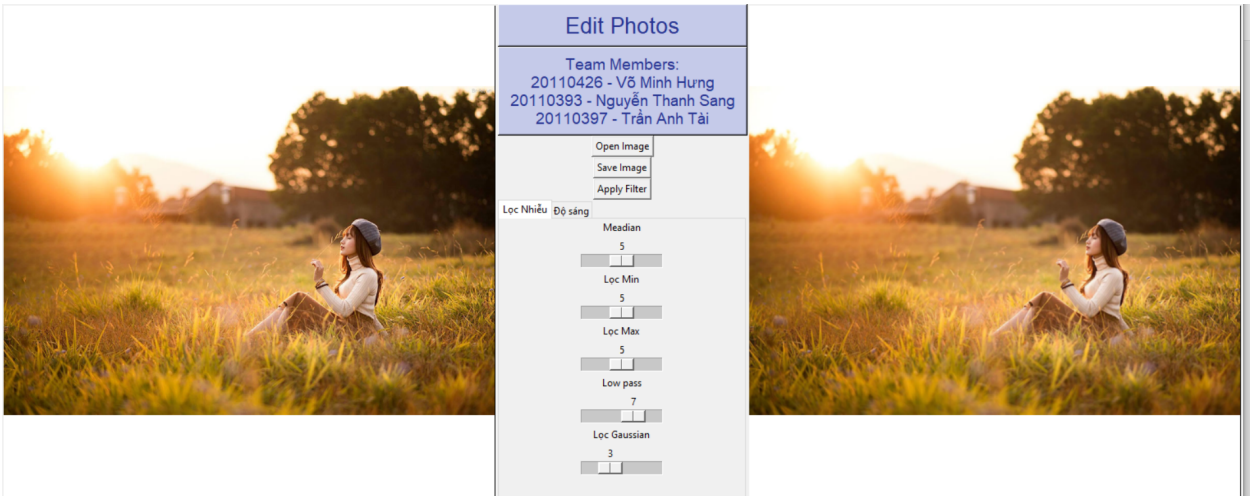

- Adjust the Median level to 5, the software shows the result


- Apply Filter and adjust the Min Filter level to 5, the software shows the result

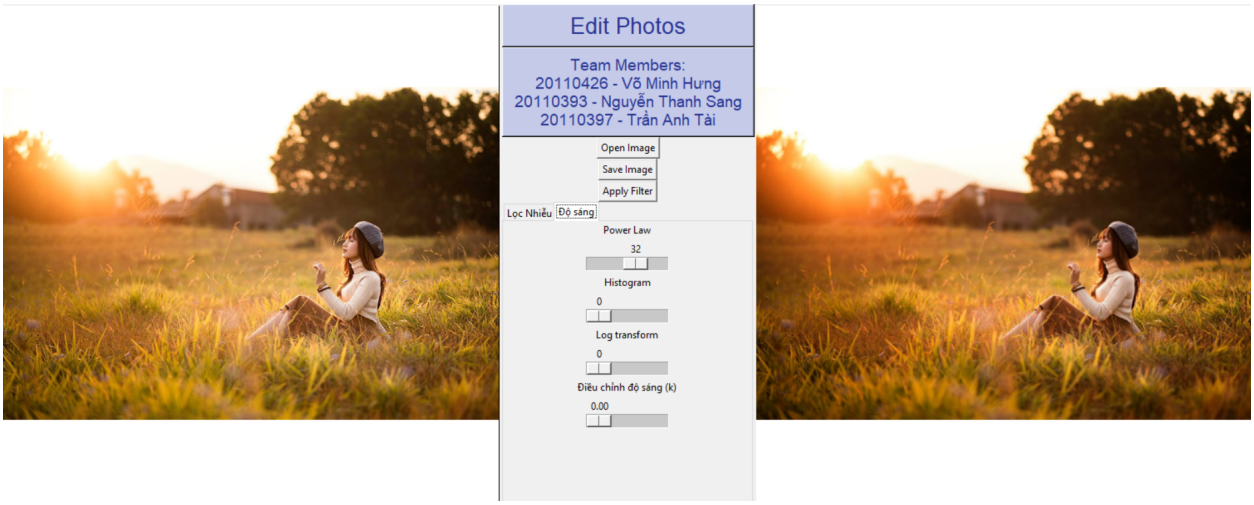
- Apply Filter and adjust the Max Filter level to 5, the software shows the result


- Apply Filter and adjust the Low pass level to 7, the software shows the result

- Apply Filter and adjust the Gaussian filter level to 3, the software shows the result

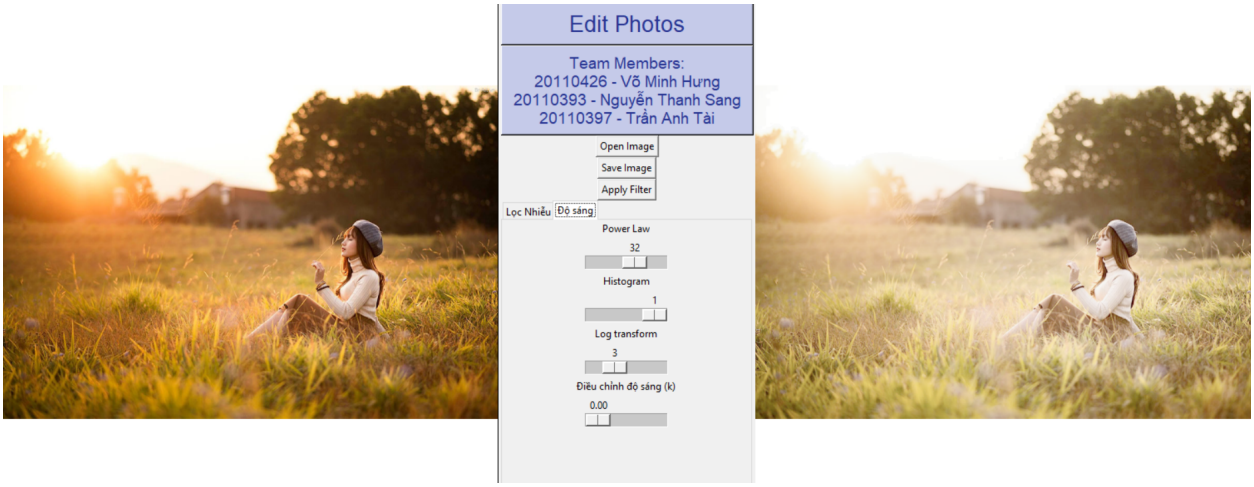



- Apply Filter and adjust the Power Law level to 32, the software shows the result

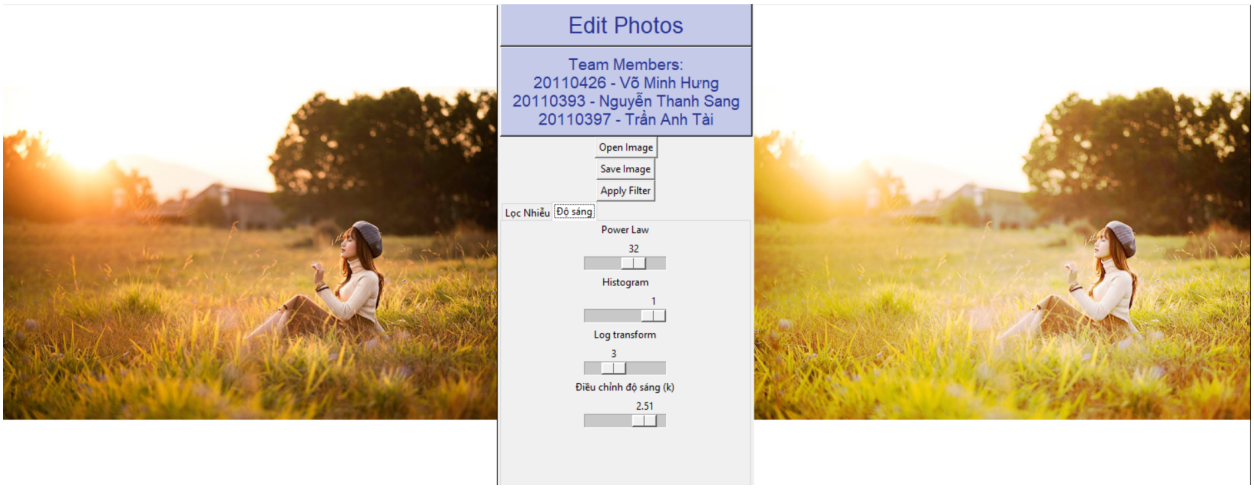


- Apply Filter and adjust the Histogram level to 1, the software shows the result

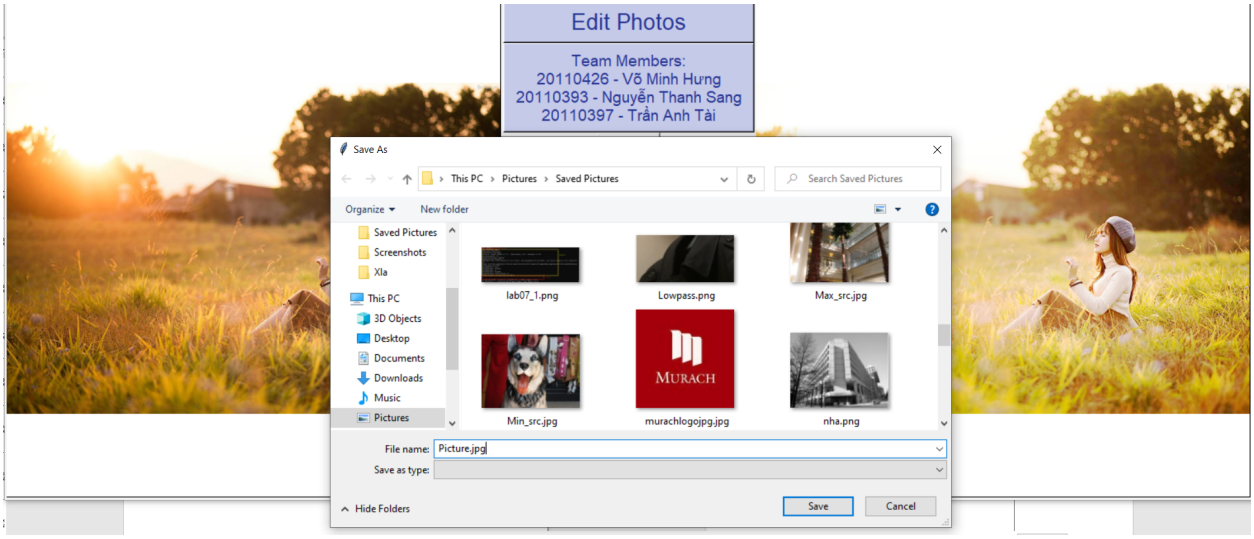

- Apply Filter and adjust the Log transform level to 3, the software shows the result




- Apply Filter and adjust the brightness level to 2.51, the software shows the result



- Save images as .jpg



# CHAPTER 6: CONCLUSION

## 6.1. Student evaluation

- Almost requirements are met.
- The code is quite clean and reuse table.

## 6.2. Difficulties

- Not enough time
- Because this is a big program so fixing bugs is very difficult.
- Difficulty in choosing a problem-solving method.

## 6.3. Development ideas

- Make a better interface.
- Upgrade the system to become more technical and professional.
- Improve the likelihood image processing and frequency filter accuracy.
- Add more features

## REFERENCES

Arena Multimedia**.** (2021). *"Hướng dẫn học Thiết kế đồ họa bằng Photoshop từ đầu chi tiết"*. https://www.arena-multimedia.vn/tin-multimedia/hoc-thiet-ke-do-hoa-bang-photoshop/

Daviesmediadesign. (2020). *"25 Hướng dẫn chỉnh sửa ảnh GIMP cho nhiếp ảnh gia"*. https://daviesmediadesign.com/vi/25-h%C6%B0%E1%BB%9Bng-d%E1%BA%ABn-ch%E1%BB%89nh-s%E1%BB%ADa-%E1%BA%A3nh-gimp-cho-c%C3%A1c-nhi%E1%BA%BFp-%E1%BA%A3nh-gia/

Fotor Team. (2022). *"19 Popular Online Photo Filters to Make Your Shots Stunning"*. https://www.fotor.com/blog/19-popular-online-photo-filters/

Hoang Van Dung. (2023). *"Digital Image Processing Course"*. https://fhqx.hcmute.edu.vn/course/view.php?id=39262