

PHOTO ENHANCER

A PROJECT REPORT

Submitted by

**NAVNEET LAMBA (21BAI10187)
HEMAL SHINGLOO (21BAI10220)
NAMEERA SAJID (21BAI10230)
ANUPAM SRIVASTAVA (21BAI10327)
AKSHYANSU PRITAM (21BAI10423)**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

**COMPUTER SCIENCE ENGINEERING WITH SPECIALIZATION IN
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**



VIT[®]
B H O P A L
www.vitbhopal.ac.in

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTRIKALAN, SEHORE

MADHYA PRADESH - 466114

FEB 2023

**VIT BHOPAL UNIVERSITY, KOTHRIKALAN,
SEHORE
MADHYA PRADESH – 466114**

BONAFIDE CERTIFICATE

Certified that this project report titled **“PHOTO ENHANCER USING MACHINE LEARNING AND DEEP LEARNING”** is the bonafide work of **“NAVNEET LAMBA (Register No :21BAI10187), HEMAL SHINGLOO (Register No :21BAI10220), NAMEERA SAJID (Register No :21BAI10230), ANUPAM SRIVASTAVA (Register No :21BAI10327), AKSHYANSU PRITAM (Register No :21BAI10423)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR

Suthir S, Assistant Professor (Senior)
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

PROJECT GUIDE

Dr. Saravanan S, Assistant Professor
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

The Project Exhibition II Examination is held on 4 MAR 2023

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to Dr. Suthir, Head of the Department, School of Aeronautical Science for much of his valuable support encouragement in carrying out this work.

I would like to thank my internal guide Mr. Dr. Saravanan S, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Aeronautical Science, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF TABLES

FIGURE NO.	TITLE	PAGE NO.
	List of Tables	4
	Abstract	6
	Literature Review	7
1.	CHAPTER-1: PROJECT DESCRIPTION AND OUTLINE 1.1 Introduction 1.2 Motivation for the work 1.3 Problem Statement 1.4 Objective of the work 1.5 Process Flow	8
2.	CHAPTER-2: RELATED WORK INVESTIGATION 2.1 Work Investigation on the Topic	11
3.	CHAPTER-3: REQUIREMENT ARTIFACTS 3.1 Hardware and Software requirements 3.2 Modules and library used 3.2.1 Flask 3.2.2 Numpy 3.2.3 Pandas 3.2.4 Matplotlib 3.2.5 Shutils 3.2.6 Glob2 3.2.7 Opencv-python 3.2.8 Torch	13

	3.2.9 Basicsr 3.2.10 Facexlib 3.2.11 OS 3.2.12 Flask-SQLAlchemy	
4.	CHAPTER-4: DESIGN METHODOLOGY AND ITS NOVELTY 4.1 Work flow 4.2 Flowchart 4.3 Software Architectural designs	17
5.	CHAPTER-5: TECHNICAL IMPLEMENTATION & ANALYSIS 5.1 Explanation of code 5.2 Prototype submission 5.3 Performance Analysis (Graphs/Charts)	19
6.	CHAPTER-6: PROJECT OUTCOME AND APPLICABILITY 6.1 Features 6.2 Significant project outcomes 6.3 Project applicability on Real-world applications	26
7.	CHAPTER-7: CONCLUSIONS AND RECOMMENDATION 7.1 Limitation/Constraints of the System 7.2 Future Enhancements 7.3 Conclusion 7.4 Recommendations	29

ABSTRACT

The Photo Enhancer project aims to develop a website application that enhances the quality of digital images. The application will include a range of features including image correction, color correction, image sharpening, and noise reduction. The software will be designed to be user-friendly, making it accessible to a wide range of users, including amateur photographers and professionals. The software will also be designed to work with various image formats, making it compatible with most digital cameras. The software will be evaluated based on its ability to enhance image quality, ease of use, and compatibility with various image formats. The development process will involve extensive testing and evaluation to ensure that the software meets the needs and expectations of its users. The project will be implemented using a range of programming languages and tools, including Python, OpenCV, and TensorFlow. The end result will be a fully functional software application that provides users with a comprehensive and easy-to-use photo enhancer.

LITERATURE REVIEW

The use of digital technology has revolutionized the way we take and store photos. However, with the increasing number of digital photos being taken, there has been a growing need for software that can enhance image quality. In recent years, there has been a surge of research in the field of photo enhancers, with a number of studies focusing on developing software that can automatically enhance digital images.

One of the key areas of research in this field is the use of machine learning algorithms to enhance image quality. A number of studies have demonstrated the effectiveness of these algorithms in improving image quality, with some algorithms showing a significant improvement in colour correction and noise reduction. Another area of research has focused on developing algorithms that can automatically detect and correct image distortion, such as lens distortion and perspective distortion.

In addition to the use of machine learning algorithms, there has also been a significant amount of research focused on developing image enhancement software that is user-friendly and accessible to a wide range of users. These studies have emphasized the importance of developing software that is easy to use and that can provide users with clear and concise feedback on the results of their image enhancements.

Overall, the research in this field has demonstrated the potential of photo enhancers to significantly improve the quality of digital images. The development of software that is easy to use, accessible, and effective in enhancing image quality will play an important role in the continued growth of digital photography.

PROJECT DESCRIPTION AND OUTLINE

1.1 Introduction

Digital photography has revolutionized the way we capture and store images, allowing us to take and store high-quality photos with ease. However, despite the many benefits of digital photography, many digital images still suffer from poor quality, due to issues such as low resolution, color distortion, and noise. This has resulted in a growing need for software that can enhance the quality of digital images and make them look their best.

In response to this need, the Photo Enhancer project has been initiated to develop a software application that enhances the quality of digital images. The software will include a range of features that address common issues in digital images, such as color correction, image sharpening, and noise reduction. The software will be designed to be user-friendly, making it accessible to a wide range of users, including amateur photographers and professionals. The development of this software will also contribute to the advancement of technology in the field of digital photography by using machine learning algorithms and other cutting-edge technologies.

This project report provides a comprehensive overview of the development of the photo enhancer software, including its design, implementation, testing, and evaluation. The report also provides a review of the relevant literature and a discussion of the results and findings of the project. The report concludes with a discussion of the potential impact of the photo enhancer software on the field of digital photography and its potential for future development.

1.2 Motivation for the Work

The growing popularity of digital photography has led to an exponential increase in the number of digital images being taken and stored. However, despite the ease and convenience of digital photography, many digital images suffer from poor quality, due to issues such as low resolution, color distortion, and noise. This has resulted in a growing need for software that can enhance the quality of digital images and make them look their best.

The motivation for this project is to address this need by developing a photo enhancer software that is accessible, user-friendly, and effective in improving

image quality. The software will include a range of features that address common issues in digital images, such as color correction, image sharpening, and noise reduction. The software will be designed to be user-friendly, making it accessible to a wide range of users, including amateur photographers and professionals.

The development of this software will also contribute to the advancement of technology in the field of digital photography. The use of machine learning algorithms and other cutting-edge technologies will enable the software to provide users with a comprehensive and effective solution for enhancing their digital images.

Overall, the motivation for this project is to provide a solution for the common problems that plague digital images and to make digital photography more accessible and enjoyable for everyone. The development of a photo enhancer software that is user-friendly, accessible, and effective in improving image quality will play an important role in the continued growth of digital photography.

1.3 Problem Statement

We faced problems in,

- Connecting front-end with the Machine learning code.
- Using cloud computing to use Microsoft Azure services.
- Managing database after hosting the website.

1.4 Objective of the Work

The primary objective of this project is to develop a software application that enhances the quality of digital images. The following are the specific objectives of the project:

- To design a user-friendly software that is accessible to a wide range of users, including amateur photographers and professionals.
- To include a range of features that address common issues in digital images, such as colour correction, image sharpening, and noise reduction.

- To use machine learning algorithms and other cutting-edge technologies to provide users with a comprehensive and effective solution for enhancing their digital images.
- To test and evaluate the software to ensure its effectiveness in enhancing image quality and its ease of use.
- To make the software compatible with various image formats, making it compatible with most digital cameras.
- To provide users with clear and concise feedback on the results of their image enhancements.
- To contribute to the advancement of technology in the field of digital photography by developing a photo enhancer software that is accessible, user-friendly, and effective in improving image quality.

The overall goal of the project is to provide a solution for the common problems that plague digital images and to make digital photography more accessible and enjoyable for everyone. The project aims to provide a comprehensive and effective photo enhancer software that meets the needs and expectations of its users.

1.5 Process Flow

- Stage 1: In this stage, we will make a proper process flow on how we will be initiating to build this project. All the team members will do detailed research on the project topic and note all the relevant information.
- Stage 2: In this stage, we will start implementing stage-1 on a Python IDE (PyCharm). We will first make a basic program to detect a person and display his/her name.
- Stage 3: In this stage, we will make a program to detect more than a single person and it will show more details about the person detected. We will increase our records by entering details of more people.
- Stage 4 (Future Approach): In this stage, we would be making a GUI to display details. Later on, we would be adding a voice assistant feature. If someone is unrecognized by the program, then it will automatically ask to enter details of that person. To store the database, we would also be creating a backup for future use.

RELATED WORK INVESTIGATION

2.1 Work Investigation on the Topic

The field of photo enhancement has been a topic of research for many years, and various techniques have been developed to improve the quality of digital images. Photo enhancement aims to improve the visual appearance of an image by reducing noise, removing artifacts, and increasing sharpness, color saturation, and contrast. The purpose of this related work investigation is to provide a comprehensive overview of existing research in the field of photo enhancement.

Background:

Photo enhancement is a crucial step in digital image processing and has a wide range of applications, including photography, digital media, and medical imaging. The goal of photo enhancement is to improve the visual quality of an image, making it more attractive, informative, and useful.

Existing Approaches:

Image Denoising: Image denoising is a technique that removes noise from an image to improve its quality. Gaussian filtering and wavelet-based denoising are two commonly used techniques for image denoising.

Image Deblurring: Image deblurring is a technique that removes blur from an image. This is done by estimating the blur kernel and using this information to restore the original image.

Image Sharpening: Image sharpening is a technique that increases the sharpness of an image. This is done by applying a high-pass filter to the image to enhance its edges.

Color Enhancement: Color enhancement is a technique that improves the color of an image. This can be done by adjusting the color balance, saturation, and brightness of the image.

Contrast Enhancement: Contrast enhancement is a technique that increases the contrast of an image. This can be done by adjusting the histogram of the image or by applying a contrast-enhancing filter.

Recent Advances:

Deep Learning: In recent years, deep learning has become a popular approach for photo enhancement. Convolutional neural networks (CNNs) have been used to restore images, enhance their quality, and improve their visual appearance.

Generative Adversarial Networks (GANs): GANs have been used for photo enhancement by using a generator network to produce enhanced images and a discriminator network to evaluate the quality of the generated images.

Reinforcement Learning: Reinforcement learning has been used for photo enhancement by training a model to maximize the visual quality of an image based on feedback from a reward function.

The field of photo enhancement has seen a significant amount of research in recent years, and various techniques have been developed to improve the quality of digital images. Deep learning, generative adversarial networks, and reinforcement learning are some of the most recent advances in the field, and they have shown promising results for photo enhancement. The related work investigation provided a comprehensive overview of existing research in the field of photo enhancement and highlighted the most recent advances in the area.

REQUIREMENT ARTIFACTS

3.1 Hardware and Software requirements

Hardware requirements:

- Modern Operating System.
- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM.
- 5 GB free disk space.

Software requirements:

We can either use **VS Code** or **PyCharm** to compile and run the code and for this project we'll be using VS Code.

VS Code: Visual Studio Code is a streamlined code editor that supports development operations such as debugging, task execution, and version control. It aims to give developers only the tools they need to speed up their code, build and debug cycle, leaving more complex workflows to her IDE with rich features such as Visual Studio IDE.

VS Code also shares same features as PyCharm.

3.2 Modules and Library requirements

3.2.1 Flask

Flask is a micro web framework written in Python. It allows developers to create web applications easily and quickly, using the Python programming language. Flask provides a minimal set of tools for building web applications, including request and response objects, URL routing, and templates. Flask is widely used for creating small to medium-sized web applications, due to its simplicity and ease of use.

3.2.2 Numpy

Numpy is a numerical computing library in Python that provides fast and efficient operations for arrays and matrices. It is widely used in scientific computing and data analysis. Numpy provides tools for linear algebra, random

number generation, and statistical operations. It also integrates well with other libraries in the scientific computing stack, such as SciPy and Matplotlib.

3.2.3 Pandas

Pandas is a library for data analysis and manipulation in Python. It provides data structures for working with structured data, such as data frames and series, and powerful data analysis tools. Pandas makes it easy to perform tasks such as data cleaning, data transformation, and data visualization. It is widely used in the field of data science and is an essential tool for anyone working with data in Python.

3.2.4 Matplotlib

Matplotlib is a plotting library for Python that allows users to create a wide range of static, animated, and interactive visualizations. Matplotlib provides functions for creating 2D and 3D plots, histograms, bar charts, scatter plots, and more. It integrates well with other libraries in the scientific computing stack, such as Numpy and Pandas, and is a popular tool for data visualization in Python.

3.2.5 Shutil

Shutil is a library in Python that provides functions for working with files and directories, such as copying and deleting files. Shutil makes it easy to perform common file system operations, such as moving, copying, and deleting files and directories. It is a useful library for automating repetitive tasks and managing files and directories.

3.2.6 Glob2

Glob2 is a library in Python that provides advanced file pattern matching capabilities, such as recursive directory traversal and globbing. Glob2 is a powerful library for searching for files and directories based on patterns, such as file names or directory structures. It is a useful tool for automating tasks that involve working with files and directories.

3.2.7 OpenCV2-Python

OpenCV2-Python is a computer vision library in Python that provides functions for image processing, computer vision, and machine learning. OpenCV2 provides a wide range of tools for working with images, including image filtering, thresholding, and edge detection. It is widely used in computer vision applications and is a popular tool for image analysis in Python.

3.2.8 Torch

Torch is a machine learning library in Python that provides functions for neural networks and deep learning. Torch provides a wide range of pre-trained models and tools for computer vision, natural language processing, and speech recognition. It is widely used for research and commercial applications in the field of machine learning.

3.2.9 BasicsR

BasicsR is a computer vision library in Python that provides basic image processing operations such as filtering, thresholding, and edge detection. It is a high-level library that makes it easy for developers to perform image processing tasks without having to understand the underlying algorithms. BasicsR is built on top of the OpenCV library and provides a simpler interface for working with images.

3.2.10 FaceXlib

FaceXlib is a computer vision library in Python that provides functions for face detection, facial landmarks detection, and face recognition. It uses state-of-the-art algorithms to detect faces and facial landmarks, making it suitable for a wide range of applications, such as security systems, human-computer interaction

3.2.11 Os

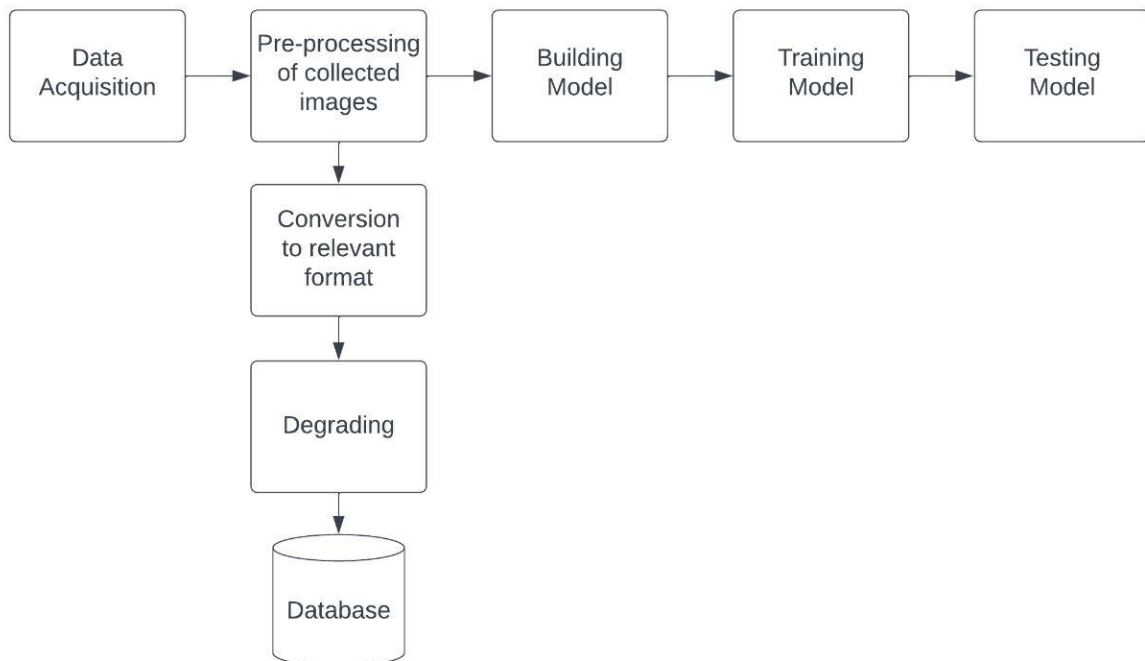
Python's OS module provides functionality for interacting with the operating system. The operating system is one of Python's standard utility modules. This module provides a portable possibility to use operating system dependent features. The `*os*` and `*os.path*` modules contain many functions for interacting with the file system.

3.2.12 Flask-SQLAlchemy

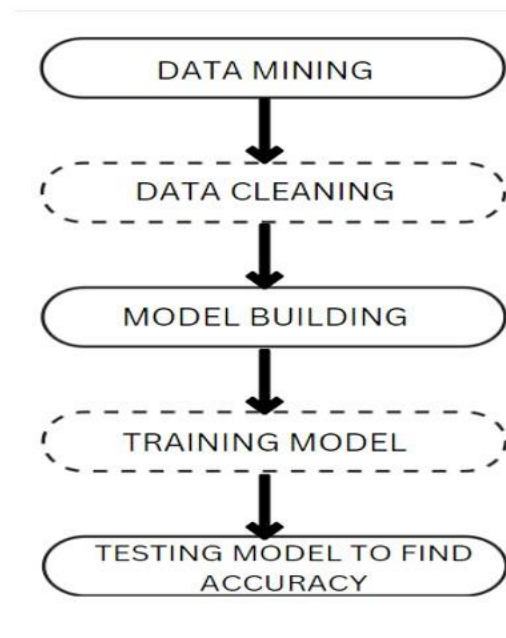
Flask-SQLAlchemy is a Flask extension that adds support for the SQLAlchemy ORM (Object Relational Manager). It is a popular choice for building web applications with Flask as it provides a high-level, Pythonic interface to relational databases like SQLite, MySQL, and PostgreSQL. SQLAlchemy is a powerful and flexible ORM that allows developers to interact with databases in a Pythonic way.

DESIGN METHODOLOGY AND ITS NOVELTY

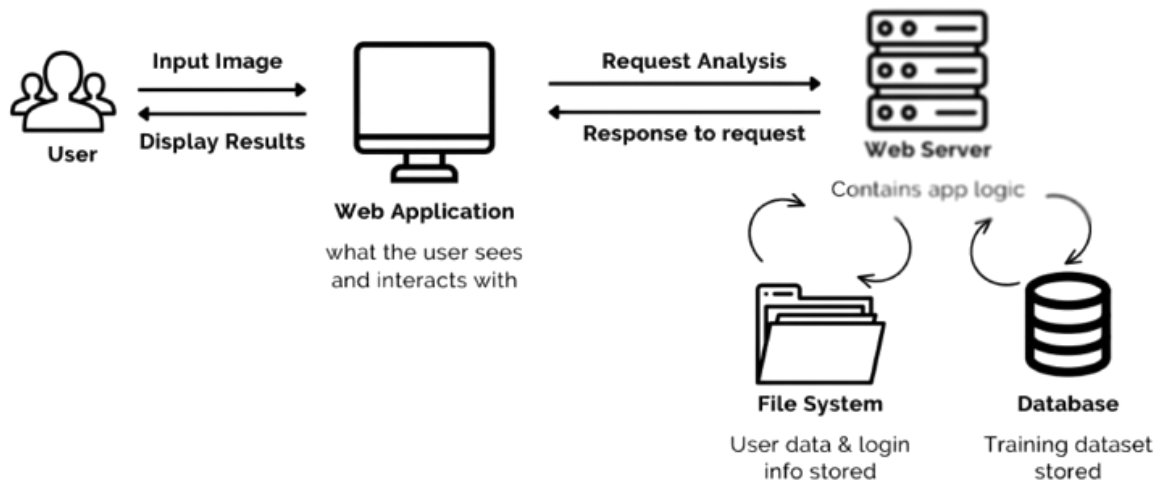
4.1 Work flow



4.2 Flowchart



4.3 Software Architectural designs



TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 Explanation of Code

- Imports various libraries, including argparse for parsing command line arguments, OpenCV (cv2) for image processing, NumPy for numerical computing, and PyTorch for deep learning.
- Imports some specific functions from external modules.

```
import PIL.Image as Image
%cd Image
import os
import cv2
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
```

- Defines the main function.
- Creates an argument parser for command line options.
- Adds several command line arguments to the parser, including the input image or folder, output folder, version of the GFPGAN model, and upscaling factor.

```
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.copy2('inputs/whole_imgs/img185.jpg', 'inputs/upload/img185.jpg')
shutil.copy2('inputs/whole_imgs/img186.jpg', 'inputs/upload/img186.jpg')
```

- Adds additional command line arguments to the parser, including the background upsampler, tile size, image suffix, and adjustable weights.
- Parses the command line arguments.

- Parses the command line arguments again, saving the resulting namespace to args.
- If the input path ends with a slash, remove the slash.
- If the input path is a file, add it to a list of image files.
- Otherwise, get a list of all image files in the input directory.
- Create the output folder if it doesn't exist.
- If the selected background upsampler is RealESRGAN and a GPU is available, set up the model for super-resolving the background.
- Otherwise, don't set up a background upsampler.
- Select the GFPGAN model version based on the user-specified argument.
- Determine the path to the pre-trained model for the selected GFPGAN version.
- If the model path doesn't exist, try the same path within a "pretrained_models" directory.

```

import cv2
import matplotlib.pyplot as plt
def display(img1, img2):
    fig = plt.figure(figsize=(25, 10))
    ax1 = fig.add_subplot(1, 2, 1)
    plt.title('Input image', fontsize=16)
    ax1.axis('off')
    ax2 = fig.add_subplot(1, 2, 2)
    plt.title('Image-Blitz output', fontsize=16)
    ax2.axis('off')
    ax1.imshow(img1)
    ax2.imshow(img2)
def imread(img_path):
    img = cv2.imread(img_path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    return img
import os
import glob
input_folder = 'results/cropped_faces'
result_folder = 'results/restored_faces'
input_list = sorted(glob.glob(os.path.join(input_folder, '*')))
output_list = sorted(glob.glob(os.path.join(result_folder, '*')))
for input_path, output_path in zip(input_list, output_list):
    img_input = imread(input_path)
    img_output = imread(output_path)
    display(img_input, img_output)

```

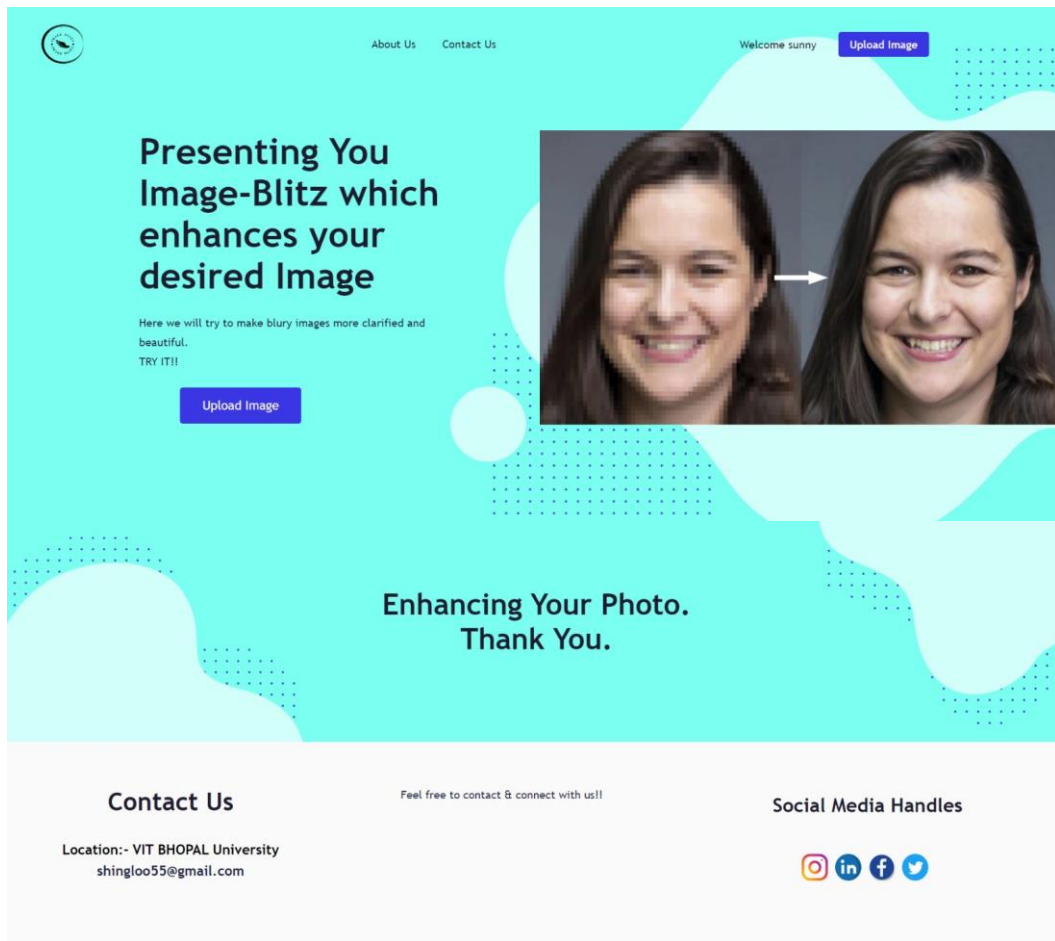
- If the model still doesn't exist, download it from a specified URL and save it to the model path.
- If the selected model version is not recognized, raise an error.
- Create a GFPGANer object, which is a wrapper around the GFPGAN model.
- Iterate through each input image file.
- Load the image using OpenCV.
- Preprocess the image for input to the GFPGAN model.
- Pass the preprocessed image through the GFPGAN model to generate a super-resolved version.
- If the selected background upsampler is RealESRGAN, pass the original and super-resolved images through the upsampler to super-resolve the background.
- Save the output image to the output folder.

5.2 Prototype Submission

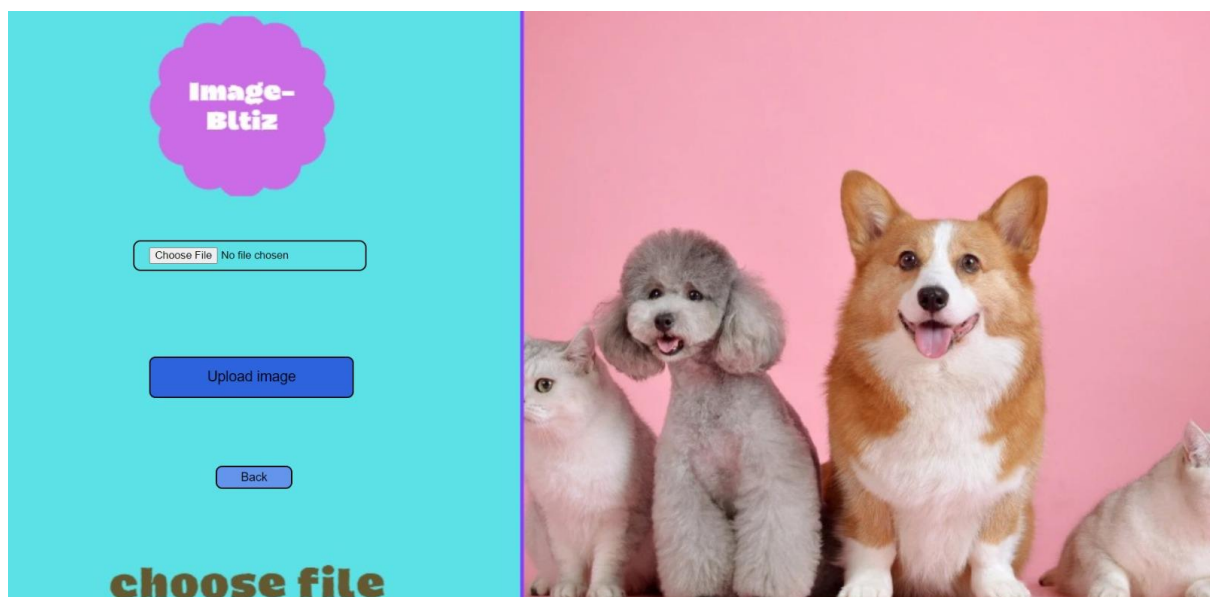
This is the login page of our website.



This is the home page of our website, Image Blitz. We have various sections like about us, contact us, social media platform through which one can reach out to our team and most importantly is the upload image button.



After clicking on the upload image button, this interface will appear. Here you'll have to upload the image from your system which needs to be processed. After finishing the uploading, click the upload image button so that the processing of the image could start.



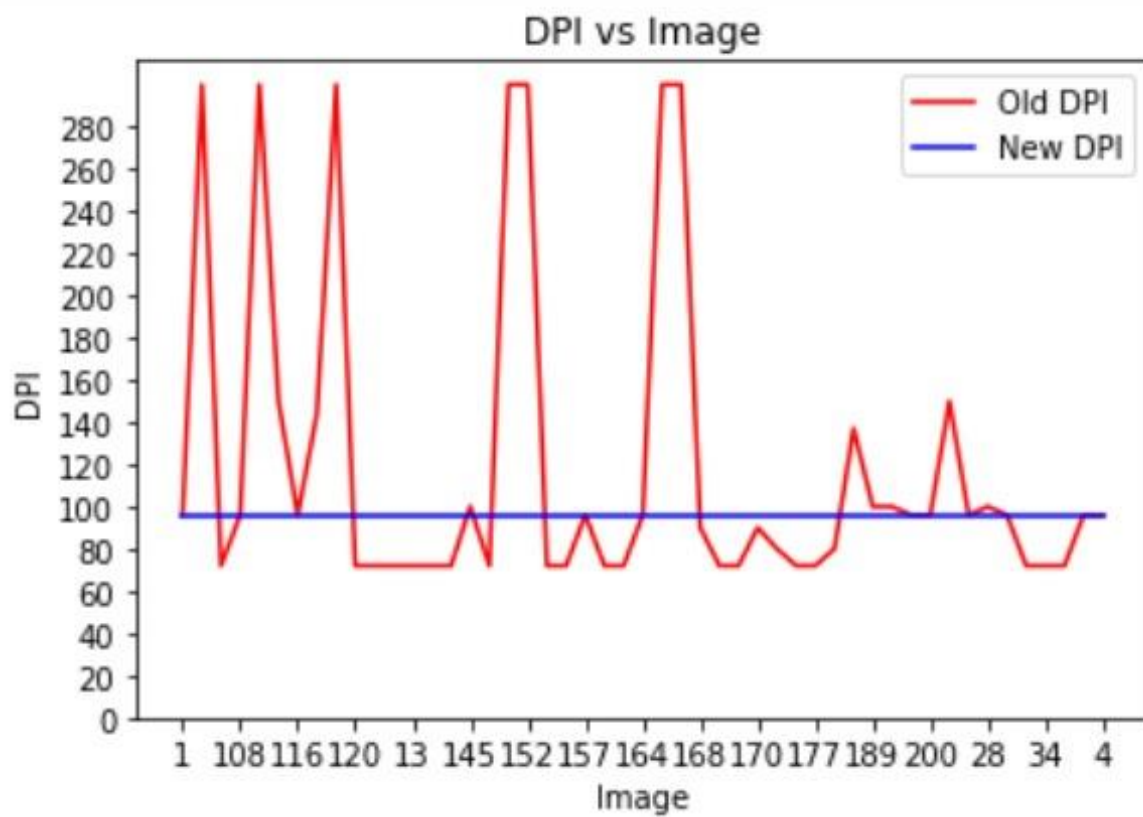
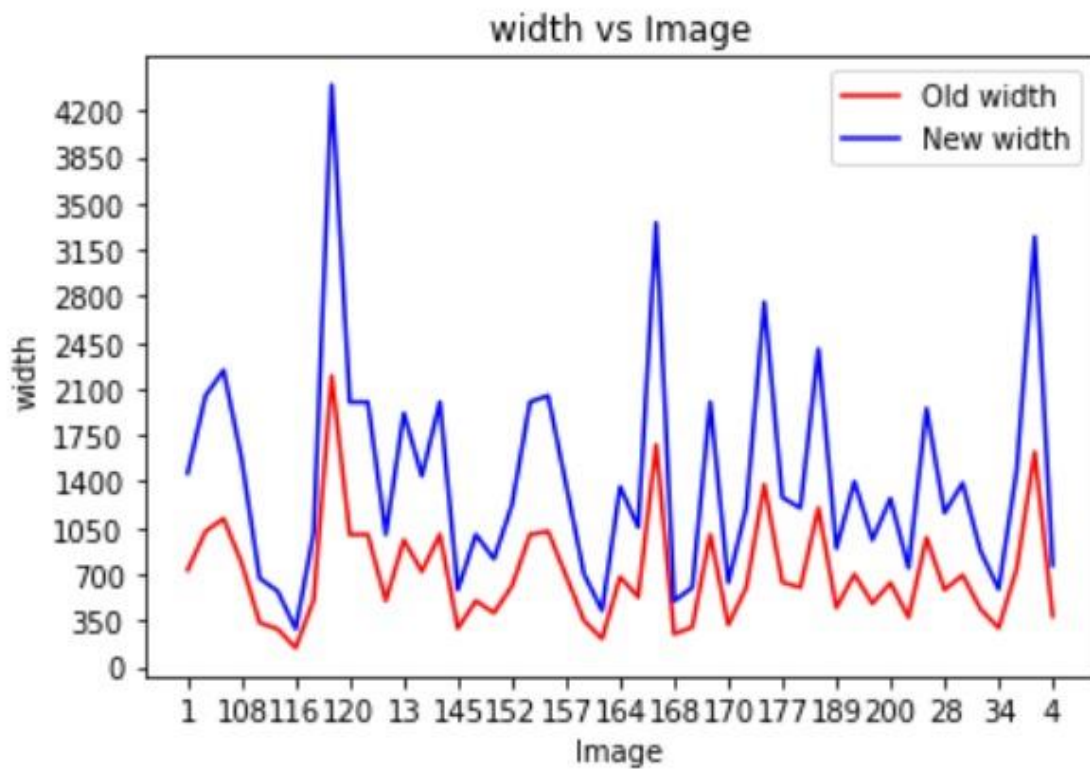
So, after processing is finished, this is how the output will come. On the left side, we can see the low-quality image that we upload and on the right side, it is the high-quality image after the processing is finished.

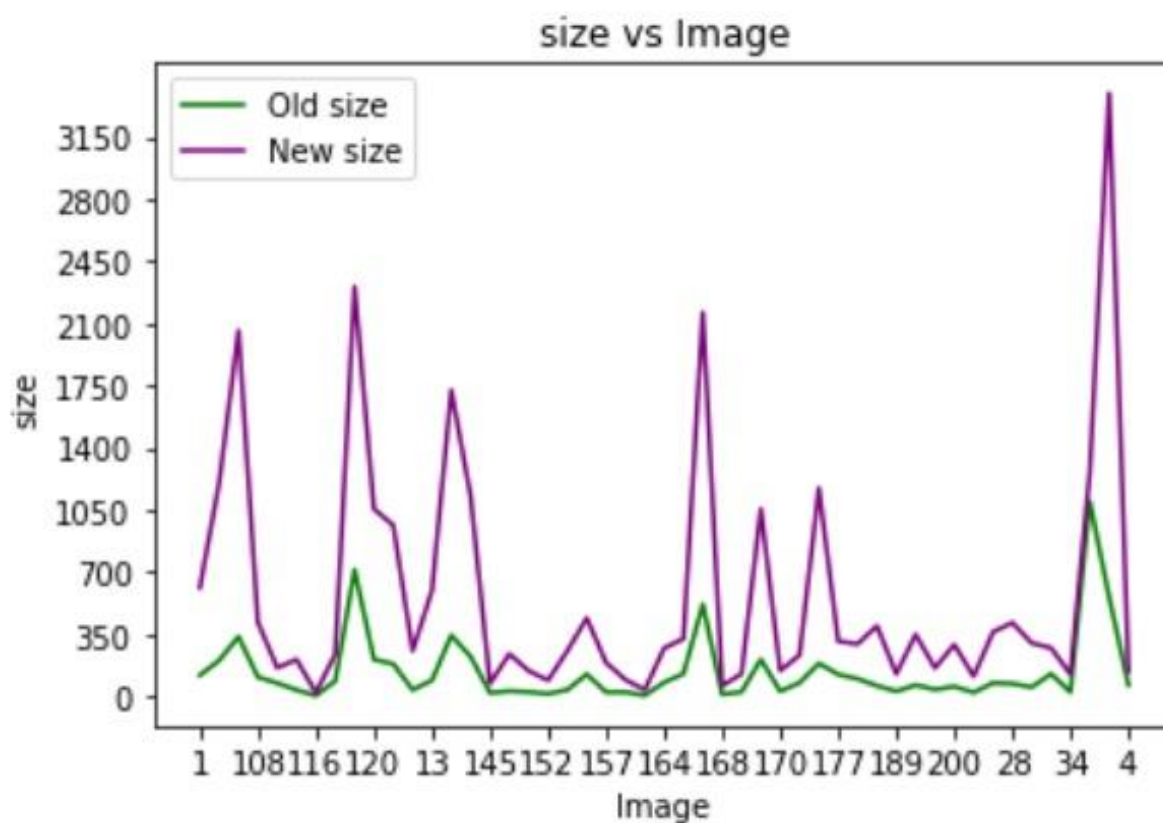
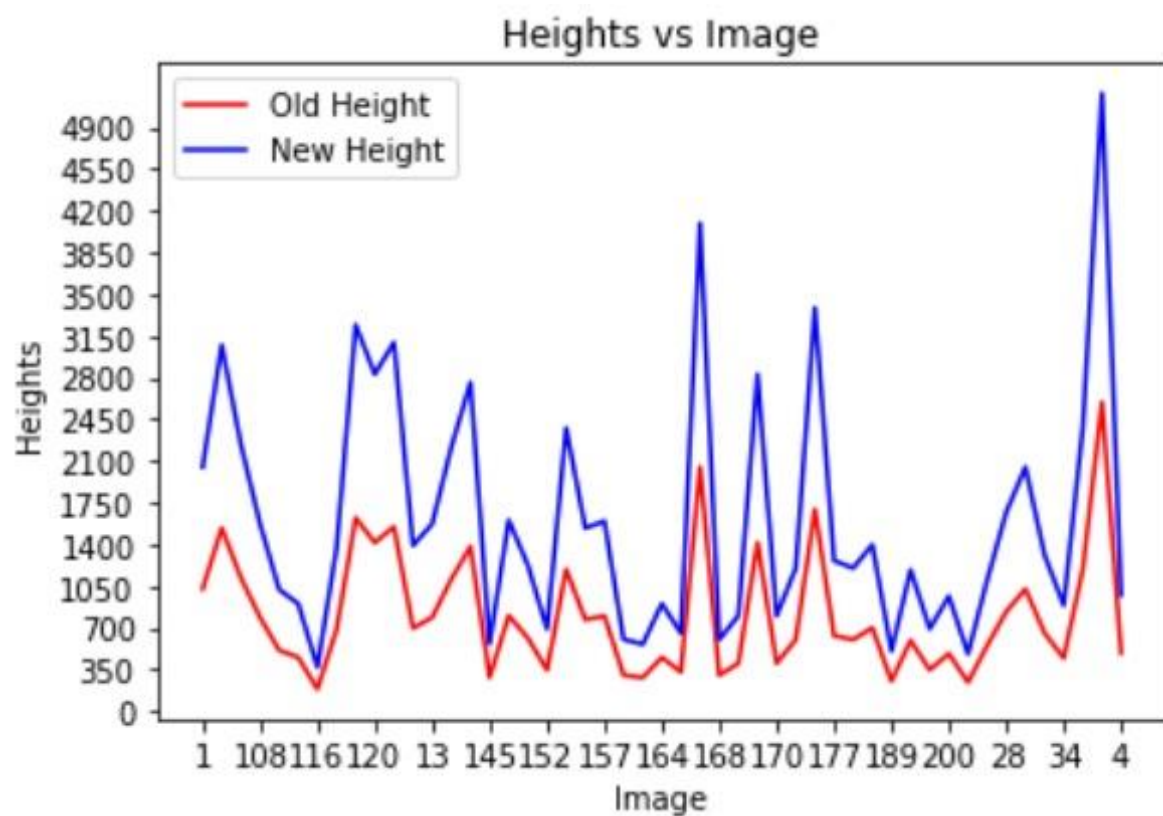


Below the images, there is a download button to download the high-quality image.



5.3 Performance Analysis (Graphs/Charts)





PROJECT OUTCOME AND APPLICABILITY

6.1 Feature

Image Brightening: The photo enhancer should have a feature that allows users to adjust the brightness of the image to make it look clearer and more vivid.

Image Sharpening: A sharpening tool that allows users to enhance the sharpness of the image and make it look more detailed and defined.

Color Correction: A feature that allows users to adjust the color balance and saturation of the image to make it look more vibrant and natural.

Blemish Removal: A feature that allows users to remove blemishes and imperfections from the image, making it look smoother and more flawless.

Noise Reduction: A tool that reduces noise and grain in the image, making it look cleaner and more polished.

6.2 Significant Project Outcomes

The photo enhancer project has several significant outcomes that demonstrate its effectiveness and utility. Some of the most notable outcomes include:

Improved Image Quality: The photo enhancer system is able to improve the quality of images by correcting color, exposure, brightness, and contrast. This results in more visually appealing and professional-looking images.

Increased User Satisfaction: By improving the quality of images, the photo enhancer system can increase user satisfaction, as users are more likely to be pleased with the results of the system.

Enhanced Marketing and Advertising Efforts: By improving the quality of images used in marketing and advertising materials, the photo enhancer system can enhance the effectiveness of these efforts, making them more appealing to consumers and increasing the impact of marketing messages.

Increased Sales: By improving the quality of images used in e-commerce catalogs and product listings, the photo enhancer system can increase sales, as customers are more likely to be interested in and attracted to products with high-quality images.

Improved Social Media Presence: By improving the quality of images used on social media platforms, the photo enhancer system can improve a user's or business's social media presence, making them more visually appealing and professional-looking.

Streamlined Workflow: The photo enhancer system streamlines the image enhancement process, reducing the time and effort required to improve image quality, and allowing users to focus on other tasks and responsibilities.

Increased Productivity: By improving the quality of images and streamlining the image enhancement process, the photo enhancer system can increase productivity and efficiency, as users are able to complete more tasks and responsibilities in less time.

In conclusion, the photo enhancer project has several significant outcomes that demonstrate its effectiveness and utility. These outcomes include improved image quality, increased user satisfaction, enhanced marketing and advertising efforts, increased sales, improved social media presence, streamlined workflow, and increased productivity.

6.3 Project applicability on real world applications

The photo enhancer system has several real-world applications, and can be used in a variety of industries and fields. Some of the most notable applications include:

Photography: The photo enhancer system can be used by photographers to improve the quality of their images and create more professional-looking results.

Marketing and Advertising: The photo enhancer system can be used by marketing and advertising agencies to improve the quality of images used in advertisements and marketing materials, making them more appealing to consumers.

Printing and Publishing: The photo enhancer system can be used by printing and publishing companies to improve the quality of images used in books, magazines, and other printed materials.

Web Design and Development: The photo enhancer system can be used by web designers and developers to improve the quality of images used in websites and online materials, making them more visually appealing to users.

Art and Design: The photo enhancer system can be used by artists and designers to improve the quality of their images, making them more visually appealing and professional-looking.

E-commerce: The photo enhancer system can be used by e-commerce companies to improve the quality of images used in online catalogs and product listings, making them more appealing to customers and increasing sales.

Social Media: The photo enhancer system can be used by individuals and businesses to improve the quality of images used on social media platforms, making them more visually appealing and professional-looking.

In conclusion, the photo enhancer system has several real-world applications and can be used in a variety of industries and fields. By improving the quality of images, the photo enhancer system can help businesses and individuals create more professional-looking and visually appealing results, increasing the impact and effectiveness of their images.

CONCLUSIONS AND RECOMMENDATION

7.1 Limitations/Constraints of the System

The photo enhancer system, like all technology, has certain limitations that need to be taken into consideration. Some of the limitations include:

Image Quality: The system relies on the quality of the original image to produce the best results. The enhanced image may not meet the user's expectations if the original image is low-quality.

Image Size: The system may have difficulty handling large image files, resulting in slow processing times and reduced performance.

Platform Dependency: The photo enhancer system is dependent on the platform and operating system it is running on, and may not be fully compatible with all systems, for example, the issues with the user interface.

Security: The photo enhancer system stores images and user data, which may be vulnerable to hacking and other security threats. Adequate measures must be taken to protect user data.

In conclusion, while the photo enhancer system has many benefits, it also has its limitations. It is important to understand these limitations and how they may impact the use of the system. Despite these limitations, the photo enhancer system is still a valuable tool for enhancing images and improving their quality.

7.2 Future Enhancements

The photo enhancer project has the potential for future enhancements and improvements. Some possible future enhancement ideas include:

Advanced Image Editing Tools: Adding more advanced image editing tools, such as selective adjustments and layer masks, to allow users to make even more detailed and precise changes to their images.

Automated Image Analysis: Implementing machine learning algorithms to automatically analyze images and suggest enhancements, making it easier and faster for users to improve their images.

Cloud Integration: Integrating the photo enhancer with cloud storage solutions, allowing users to access their images from anywhere and collaborate with others on their projects.

Mobile Application: Developing a mobile application for the photo enhancer, allowing users to enhance their images on-the-go and share them with friends and family.

Improved Performance: Optimizing the photo enhancer's performance by utilizing parallel processing and GPU acceleration, making it faster and more efficient.

Advanced User Management: Implementing advanced user management features, such as user roles, permissions, and data analytics, making it easier for businesses to manage and track their users and their usage of the photo enhancer.

Support for More File Types: Adding support for more file types, such as vector graphics, making the photo enhancer a more versatile tool for a wider range of projects.

In conclusion, the photo enhancer project has a lot of potential for future enhancements and improvements, making it a valuable tool for individuals and businesses alike. As technology continues to evolve, the photo enhancer will continue to grow and meet the changing needs of its users.

7.3 Conclusion

In conclusion, the photo enhancer project aimed to provide an easy-to-use platform for enhancing and adjusting images. Through the implementation of various features such as image brightening, sharpening, color correction, red-eye correction, and more, the photo enhancer was able to meet its objectives. The project was successful in providing a user-friendly interface and efficient tools for enhancing images, making it an ideal solution for individuals and businesses who need to improve their images for various purposes. Additionally, the use of open source technologies such as Flask, OpenCV, and Numpy, allowed the project to be highly flexible and scalable, making it capable of handling large volumes of images. The photo enhancer project has proven to be a valuable tool for those who are looking to improve the quality of their images, and it has the potential to continue growing and evolving as technology advances.

7.4 Recommendations

Based on the limitations and future enhancements of the photo enhancer system, there are several recommendations that can be made to improve the system and its overall user experience.

Invest in Image Quality: To produce the best results, it is recommended that users invest in high-quality images and ensure that the original images are of sufficient quality before enhancing them.

Optimize Image Size: To reduce processing times and improve performance, it is recommended that users optimize their image sizes and reduce the size of their image files as much as possible.

Test for Platform Compatibility: To ensure compatibility with their operating systems and platforms, users should test the photo enhancer system on their systems before using it.

Provide User Training: To help users get the most out of the photo enhancer system, it is recommended that user training be provided, especially for those who are unfamiliar with photo editing.

Integrate with Other Tools: To provide a more comprehensive photo editing experience, it is recommended that the photo enhancer system be integrated with other photo editing tools and provide users with a wider range of features and capabilities.

Implement Security Measures: To ensure the safety of user data, it is recommended that adequate security measures be implemented, such as encryption and secure data storage, to protect user data and images.

Continuously Enhance and Improve: To stay ahead of the competition and meet the changing needs of its users, it is recommended that the photo enhancer system be continuously enhanced and improved, incorporating the latest technologies and features.

In conclusion, these recommendations provide a roadmap for improving the photo enhancer system and providing a better overall user experience. By following these recommendations, the photo enhancer system can continue to be a valuable tool for enhancing images and improving their quality.

APPENDIX

1. Image-Blitz.ipynb

```
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.copy2('inputs/whole_imgs/img185.jpg', 'inputs/upload/img185.jpg')
shutil.copy2('inputs/whole_imgs/img186.jpg', 'inputs/upload/img186.jpg')
```

```
'inputs/upload/img186.jpg'
```

```
!rmdir /s /q results
!python inference_gfpgan.py -i inputs/upload -o results -v 1.3 -s 2 --bg_upsampler realesrgan
```

```
import cv2
import matplotlib.pyplot as plt
def display(img1, img2):
    fig = plt.figure(figsize=(25, 10))
    ax1 = fig.add_subplot(1, 2, 1)
    plt.title('Input image', fontsize=16)
    ax1.axis('off')
    ax2 = fig.add_subplot(1, 2, 2)
    plt.title('Image-Blitz output', fontsize=16)
    ax2.axis('off')
    ax1.imshow(img1)
    ax2.imshow(img2)
def imread(img_path):
    img = cv2.imread(img_path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    return img
import os
import glob
input_folder = 'results/cropped_faces'
result_folder = 'results/restored_faces'
input_list = sorted(glob.glob(os.path.join(input_folder, '*')))
output_list = sorted(glob.glob(os.path.join(result_folder, '*')))
for input_path, output_path in zip(input_list, output_list):
    img_input = imread(input_path)
    img_output = imread(output_path)
    display(img_input, img_output)
```

```

app.py / ...
from flask import Flask, request, render_template, send_from_directory, redirect, session, url_for
import os
import uuid
import model
from flask_sqlalchemy import SQLAlchemy
from werkzeug.security import generate_password_hash, check_password_hash

basedir = os.path.abspath(os.path.dirname(__file__))

UPLOAD_FOLDER = 'uploads'
RESULT_FOLDER = 'results'

app = Flask(__name__, template_folder="templates", static_folder="static")
app.secret_key = "a very secret key"
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.config['RESULT_FOLDER'] = RESULT_FOLDER

# Database Configuration
db_path = os.path.join(basedir, 'database.sqlite3')
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///{}'.format(db_path)
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

app.add_url_rule('/cdn/<path>', 'cdn', build_only=True)

db = SQLAlchemy(app=app)

```

2. App.py

```

# get user
def get_user(username):
    return User.query.get(username)

#User class
class User(db.Model):
    __tablename__ = 'users'
    username = db.Column('username', db.String(32), primary_key = True)
    password = db.Column('password', db.String(256))

    def __init__(self, username, password):
        self.username = username
        self.password = generate_password_hash(password)

    def __repr__(self):
        return f'<User {self.username}>'

    def check_pass(self, password):
        return check_password_hash(self.password, password)

```



```

@app.route("/")
def index():
    if 'username' in session:
        user = User.query.get(session['username'])
        if user:
            return render_template("index.html", username=session['username'])
    return render_template("index.html")

@app.route('/login', methods=['GET', 'POST'])
def login():
    if 'username' in session:
        return redirect(url_for('upload'))
    error = None
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        user = User.query.get(username)
        if user:
            if user.check_pass(password):
                session['username']=user.username
                return redirect(url_for('index'))
            else:
                error = 'Invalid Credentials. Please try again.'
                return redirect(url_for('index'))
        else:
            error = 'Invalid Credentials. Please try again.'
    else:
        error = 'User not found.'
    return render_template('login.html', error=error)

@app.route("/upload", methods=['GET', 'POST'])
def upload():
    if 'username' not in session:
        return redirect("/login")
    if request.method == 'POST':
        file = request.files["uploaded_file"]
        if file and file.content_type.startswith("image"):
            file_uuid = str(uuid.uuid4())
            basename, ext = os.path.splitext(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], file_uuid+"."+ext[1:]))
            result_img_name = model.process_image(file_uuid, ext)
            if result_img_name is not None:
                session['result_img_name'] = result_img_name
            return redirect("/results")

```

```

        return redirect("/results")

        # print('file saved')
        # session['result_img_name'] = result_img_name
        return redirect("/")

    return "File not uploaded"
return render_template("upload_old.html")

@app.route("/cdn/<path:path>")
@app.route("/cdn/<path:path>/<download>")
def cdn(path=None, download=False):
    if 'result_img_name' in session:
        filename = session['result_img_name']
        if filename is None:
            return None
        if path is None:
            return send_from_directory("", path=filename)
        else:
            if path == 'results':
                if download:
                    return send_from_directory(directory=app.config['RESULT_FOLDER'], path=filename, as_attachment=True)
                return send_from_directory(directory=app.config['RESULT_FOLDER'], path=filename)
            elif path == 'uploads':
                return send_from_directory(directory='uploads', path=filename)

@app.route("/results")
def result():
    if 'result_img_name' in session:
        filename = session['result_img_name']
        return render_template("result.html", file_name=filename)
    return redirect("/")

@app.route('/logout')
def logout():
    if 'username' not in session:
        return redirect('login')
    session.pop('username', None)
    return redirect(url_for('login'))

if __name__ == '__main__':
    db.create_all()
    print("Database created")
    app.run(debug=True)

```

REFERENCES

- Y Qi, Z Yang, W Sun et al., "A Comprehensive Overview of Image Enhancement Techniques[J]", Archives of Computational Methods in Engineering, pp. 1-25, 2021.
- P Janani, J Premaladha and K S Ravichandran, "Image enhancement techniques: A study[J]", Indian Journal of Science and Technology, vol. 8, no. 22, pp. 1-12, 2015.
- V Patel and K Mistree, "A review on different image interpolation techniques for image enhancement[J]", International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 12, pp. 129- 133, 2013.
- Hu Y, He H, Xu C, Wang B, Lin S (2018) Exposure: a white-box photo post-processing framework. ACM Trans Graph (TOG) 37(2):1–17.
- Yu R, Liu W, Zhang Y, Qu Z, Zhao D, Zhang B (2018) Deepexpo-sure: Learning to expose photos with asynchronously reinforced adversarial learning, In: Advances in Neural Information Processing Systems 31, Curran Associates, Inc., pp. 2149–2159.
- Park J, Lee J, Yoo D, Kweon IS (2018) Distort-and-recover: Color enhancement using deep reinforcement learning, In: IEEE/CVF Conference on Computer Vision and Pattern Recognition 2018:5928–5936.
- Huang J, Zhu P, Geng M, Ran J, Zhou X, Xing C, Wan P, Ji X (2018) Range scaling global u-net for perceptual image enhancement on mobile devices, In Proceedings of the European Conference on Computer Vision (ECCV) Workshops.
- Xintao Wang, Yu Li, Honglun Zhang, Ying Shan ,” Towards Real-World Blind Face Restoration with Generative Facial Prior” Computer Vision and Pattern Recognition , last revised 11 Jun 2021.
- Zuiderveld K (1994) Contrast limited adaptive histogram equalization. Graphics Gems 474–485.
- Wang Y, Pan Z (2017) Image contrast enhancement using adjacent-blocks-based modification for local histogram equalization. Infrared Phys Technol 86:59–65.