

# Report

## Introduction

This report outlines the functionality and execution process of a Python script designed for image enhancement using a pre-trained deep learning model. The script leverages TensorFlow and Keras libraries to load the model, perform recursive image enhancement, and calculate the Peak Signal-to-Noise Ratio (PSNR) to evaluate the quality of enhanced images.

## Dependencies

The script utilizes the following libraries:

- ``tensorflow`` for loading the model and performing PSNR calculations.
- ``glob`` for file path manipulations.
- ``imageio`` for reading and writing image files.
- ``numpy`` for handling arrays.
- ``os`` for directory operations.

## Model Loading

The pre-trained model is loaded from a file. This model is assumed to be trained for image enhancement tasks. A custom PSNR metric function is provided to the ``load_model`` function to ensure compatibility.

## PSNR Calculation

The PSNR is a widely used metric to measure the quality of reconstructed images. It is calculated using TensorFlow's ``tf.image.psnr`` function.

## Image Enhancement Function

The core of the script is the recursive function ``enhance_image``, which enhances the input image for a specified number of iterations. Depending on the ``flag`` parameter, the function normalizes the image differently and calculates the enhanced image using the model's predictions.

## **Loading and Enhancing Images**

The script loads images from a specified directory and stores them in a list. The first image in this list is then enhanced using the `enhance_image`` function with 8 iterations and a flag set to 1.

## **Saving Enhanced Image**

The enhanced image is saved to a specified output directory. The script ensures the output directory exists and writes the enhanced image as a PNG file.

## **Conclusion**

The provided script demonstrates a systematic approach to enhancing images using a deep learning model. It integrates model loading, image normalization, recursive enhancement, and PSNR calculation to ensure high-quality image output. The final enhanced image is saved for further use or analysis, providing a complete workflow for image enhancement tasks.