

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**SOICT**

**SHORT REPORT**

**ĐỀ TÀI: LÀM TĂNG CHẤT LƯỢNG ẢNH  
THẺ BỊ MỜ**

**NHÓM 26**

**Sinh viên thực hiện:** Hoàng Công Phú 20215451  
Ngô Ngọc Sâm 20215469  
Ngô Văn Tân 20210769

**Giảng viên hướng dẫn:** PSG.TS. Phạm Văn Hải

Hà Nội, Ngày 9 tháng 1 năm 2024

## 1. Bài toán

Trong thời đại ngày nay, ảnh thẻ đóng vai trò quan trọng trong nhiều lĩnh vực, từ các công việc thường ngày, xác thực danh tính, đảm bảo an ninh. Tuy nhiên, thỉnh thoảng chúng ta có thể gặp phải tình trạng ảnh thẻ mờ mịt, làm giảm chất lượng và khả năng nhận diện. Điều này thúc đẩy nhu cầu nghiên cứu và phát triển giải pháp hiệu quả để làm nét ảnh thẻ.

Bài báo cáo này tập trung vào ứng dụng của Convolutional Neural Networks (CNN) trong quá trình làm nét ảnh thẻ bị mờ. CNN, một trong những tiến bộ quan trọng nhất trong lĩnh vực học máy, đã chứng minh khả năng xuất sắc trong việc xử lý và trích xuất đặc trưng từ dữ liệu hình ảnh. Image Super-Resolution Using Deep Convolutional Network(SR-CNN) học bằng cách ánh xạ end-to-end từ ảnh có chất lượng thấp(low resolution) đến ảnh có chất lượng cao(high resolution) Chúng ta sẽ khám phá cách SR-CNN có thể được áp dụng để cải thiện chất lượng ảnh thẻ, từ đó tăng cường khả năng nhận diện và sử dụng trong các ứng dụng thực tế.

## 2. Hướng dẫn cài đặt

### a. Các thư viện cần thiết

Version Python sử dụng là 3.11.x

```
import os # to read image from file location system
import tensorflow as tf
import numpy as np # version 1.24.2
import cv2 # version 4.8.1
import matplotlib.pyplot as plt #version 3.8.2
from IPython.display import Image
from skimage.metrics import structural_similarity as ssim #version 0.22.0
```

### b. Các gói dữ liệu Train, Test và Validation

- Tập Train, Test là các ảnh có định dạng bmp được lưu trong folder *dataset/SRCNN – dataset/Train* và *dataset/SRCNN – dataset/Test*
- Tập Validation là các ảnh thẻ có định dạng jpg với kích thước và độ mờ khác nhau được lưu trong folder Validation
- Hàm main được lưu trong file main.ipynb chứa đầy đủ các bước từ tiền xử lý dữ liệu và xây dựng tập Train, Test và chạy kết quả đầu ra.
- Link driver: Driver

## 3. Metric Function: Phương pháp đánh giá

Khi nhìn vào 1 bức ảnh, chúng ta không thể biết được liệu rằng đó là ảnh chất lượng cao hay không. Có 3 phương pháp để ước lượng chất lượng ảnh mà chúng ta sẽ sử dụng:

- Peak Signal to Noise Ratio (PSNR)
- Mean Squared Error (MSE)
- Structural Similarity (SSIM)

*a. Peak Signal to Noise Ratio và Mean Squared Error (MSE)*

Được định nghĩa trên Wikipedia, PSNR là một thuật ngữ kỹ thuật dùng để mô tả tỷ lệ giữa năng lượng tối đa có thể của một tín hiệu và công suất của nhiễu gây ảnh hưởng đến tính chất chân thực của ảnh. Như bạn có thể thấy từ thuật ngữ "Tỷ lệ" trong từ ngữ, thường nó được biểu diễn dưới dạng thang đo logarithmic decibel (dB), và có mối liên hệ như sau.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

Here,  $I$  is monochrome image and  $K$  is its noisy approximation. Expressed in dB scale,

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left( \frac{\text{MAX}_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10}(\text{MAX}_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

### Hình 3.1 MSE và PSNR

Từ công thức, chất lượng hình ảnh (Image quality) sẽ tốt hơn nếu giá trị PSNR cao và MSE sẽ thấp.

*b. Structural Similarity (SSIM)*

SSIM là 1 phương pháp dự đoán nhận thức chất lượng hình ảnh, và được sử dụng để đo đường mức độ tương đồng giữa 2 bức ảnh, cụ thể là ảnh mờ và ảnh gốc (low and high resolution). Nếu giá trị càng gần 1 thì 2 bức ảnh càng giống nhau.

The SSIM index is calculated on various windows of an image. The measure between two windows  $x$  and  $y$  of common size  $N \times N$  is:<sup>[4]</sup>

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

with:

- $\mu_x$  the pixel sample mean of  $x$ ;
- $\mu_y$  the pixel sample mean of  $y$ ;
- $\sigma_x^2$  the variance of  $x$ ;
- $\sigma_y^2$  the variance of  $y$ ;
- $\sigma_{xy}$  the covariance of  $x$  and  $y$ ;
- $c_1 = (k_1 L)^2$ ,  $c_2 = (k_2 L)^2$  two variables to stabilize the division with weak denominator;
- $L$  the dynamic range of the pixel-values (typically this is  $2^{\# \text{bits per pixel}} - 1$ );
- $k_1 = 0.01$  and  $k_2 = 0.03$  by default.

## Hình 3.2 MSE và PSNR

### 4. Tiền xử lý dữ liệu - Data Processing

#### a. Chuẩn bị ảnh mờ làm dữ liệu cho tập Train

Chúng ta sẽ chuẩn bị dữ liệu là các ảnh mờ bằng cách resized image. Tức là chúng ta sẽ down scaling (giảm kích thước ảnh) đi  $k$  lần và sau đó up scaling(tăng kích thước) về kích thước của bức ảnh cũ. Sau quá trình này, ta sẽ thu được 1 bức ảnh mờ làm dữ liệu cho tập Train



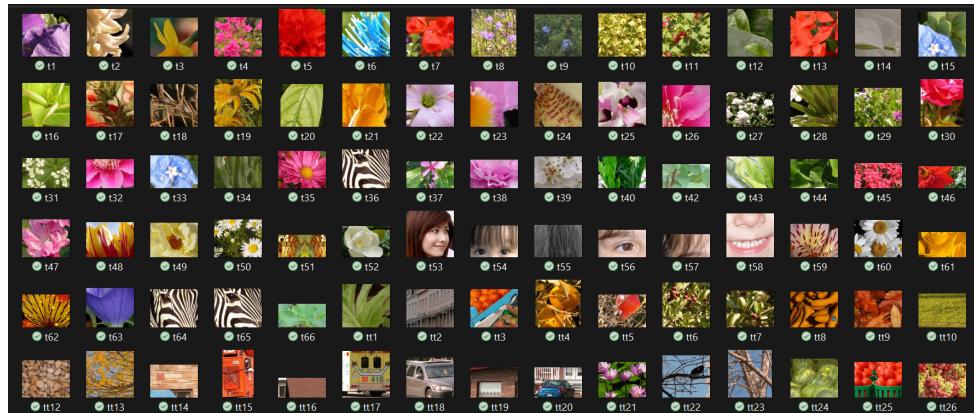
Hình 4.3 Ảnh gốc và ảnh làm mờ

Chúng ta có thể thấy bức ảnh bên phải(ảnh đã resized) mờ hơn ảnh gốc bên trái. Chúng ta có thể thay đổi tham số  $k$  để điều chỉnh độ mờ trong các bức ảnh của tập Train, trong báo cáo này chúng tôi đề xuất  $k = 2$ .

Model CNN sẽ học bằng cách điều chỉnh các tham số để làm nét ảnh từ resized image(ảnh mờ) về origin image(ảnh gốc)

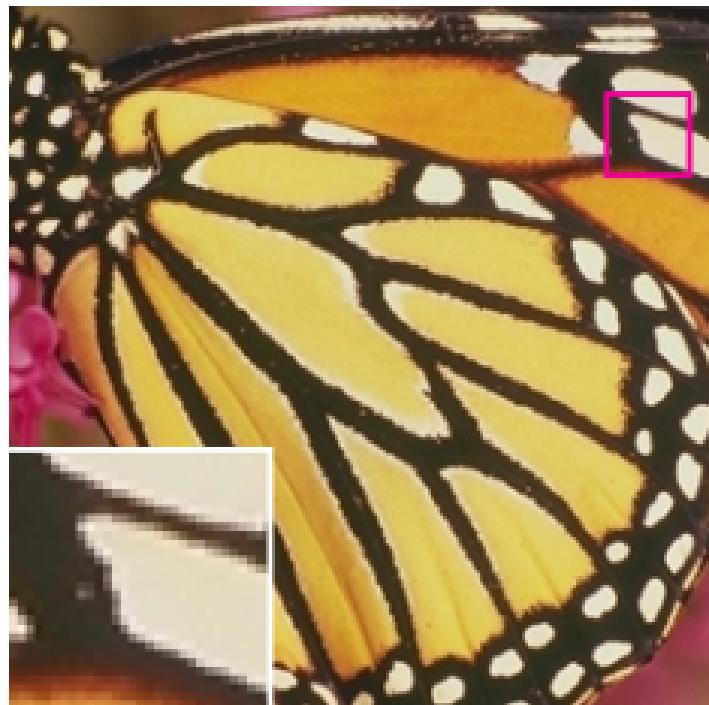
#### b. Tăng số lượng samples cho tập Train

Chúng ta sẽ làm tăng số lượng samples bằng cách chia những bức ảnh gốc thành các kernel nhỏ hơn với kích thước 32x32.



**Hình 4.4 Folder ảnh gốc**

Từ 90 bức ảnh ban đầu, sau khi phân tách chúng ta thu được hơn 14900 samples kích thước 32x32:



**Hình 4.5 Chia ảnh gốc thành nhiều ảnh nhỏ**

## 5. Mô hình bài toán

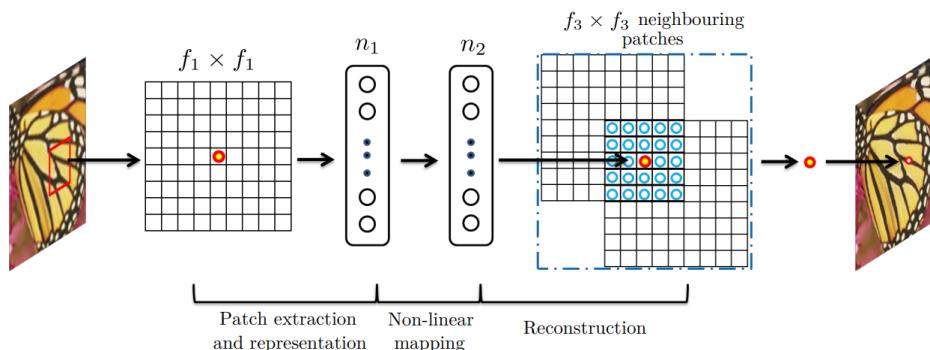
### a. Xây dựng SR-CNN Model

Chúng ta sẽ chuyển dữ định dạng ảnh sang h5 format(HDF5). H5 format được sử dụng để lưu trữ các mô hình máy học (cụ thể là các image), bao gồm trọng số của

mạng nơ-ron, kiến trúc mạng và làm tăng hiệu suất mô hình học máy.

Chúng ta sẽ xây dựng SR-CNN Model sử dụng Tensorflow-keras với model tuần tự bằng tf.keras.Sequential

Mô hình chung của bài toán có thể được mô tả như hình vẽ dưới đây:



**Hình 5.6 Mô hình chung**

### b. Thông số mô hình

- Input là low-resolution image (ảnh mờ)
- 2 convolutional hidden layer với Activation Function: ReLU
- Optimizer: sử dụng Adam với learning rate = 0.0003
- Loss Function: sử dụng Mean Square Error
- Output là high-resolution image(ảnh chất lượng cao)

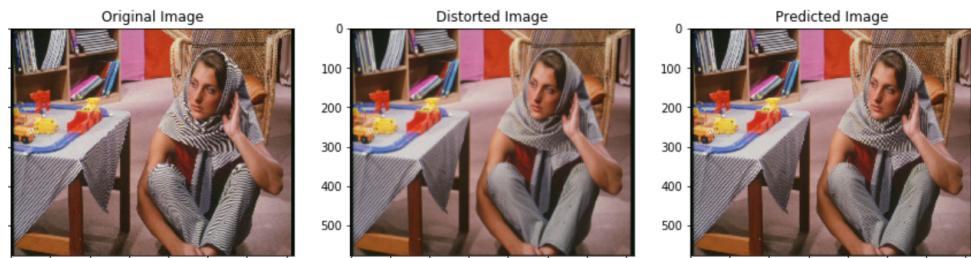
Model: "SRCNN"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, None, None, 128)	10496
conv2d_1 (Conv2D)	(None, None, None, 64)	73792
conv2d_2 (Conv2D)	(None, None, None, 1)	1601
Total params: 85,889		
Trainable params: 85,889		
Non-trainable params: 0		

**Hình 5.7 Tham số mô hình**

## 6. Training và đánh giá trên tập Test

Nhóm đã thực hiện training model qua 100 epochs với learning rate = 0.0003 trực tiếp trên CPU. Thiết bị được sử dụng là laptop chip i7 xung nhịp trung bình 2.8GHz, RAM 7GB trong thời gian 1h30p. Và sau đây là kết quả demo trên tập

Test:



**Hình 6.8 Kết quả trên tập Test**

Thông số đánh giá trên ảnh gốc và ảnh làm mờ

- PSNR: 25.804217868496153
- MSE: 512.6014274691358
- SSIM: 0.7967242506776445

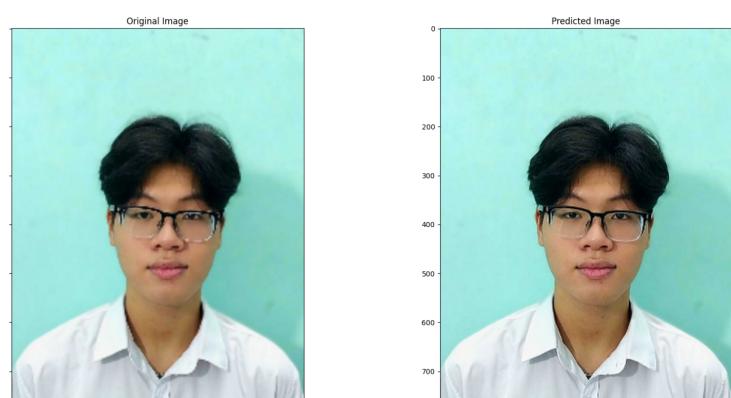
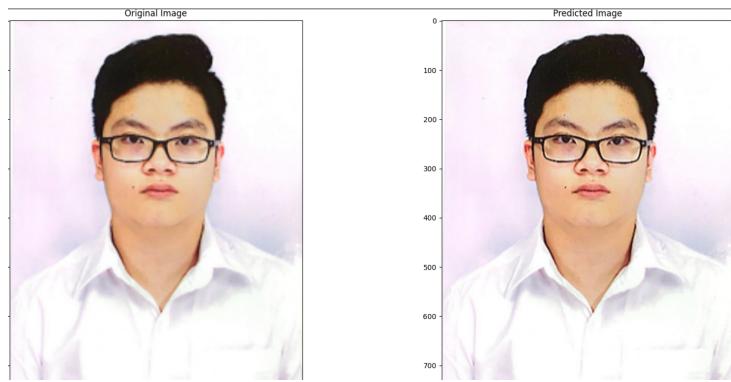
Thông số đánh giá trên ảnh gốc và ảnh output(đầu ra của model)

- PSNR: 26.432215326919476
- MSE: 443.5883101851852
- SSIM: 0.8434364090355023

Ta có thể nhìn thấy 2 thông số là độ tương đồng giữa 2 bức ảnh là PSNR, SSIM đã tăng lên, đồng thời chỉ số lỗi MSE (loss) giảm xuống. Điều này chứng tỏ mô hình đã học được và thực sự làm tăng chất lượng hình ảnh

## 7. Kết quả trên tập Validation

Chất lượng ảnh đầu ra đã được cải thiện đáng kể và tăng độ rõ nét các chi tiết. Độ sáng của các chi tiết ảnh, các vùng tương đồng đã cải thiện





## 8. Đề xuất cải tiến và kết luận

Tuy nhiên do chất lượng ảnh đầu vào với độ mờ, kích thước, độ nhòe khác nhau nên 1 số kết quả Model đưa ra chưa thực sự mang lại kết quả tốt. Quá trình tối ưu tham số của Model dựa trên Gradient Descent với learning = 0.0003.

Nhóm đề xuất thay đổi learning rate và tăng số lần train epochs để Model học tốt hơn. Ngoài ra do số lượng ảnh dùng cho tập Train còn ít và chưa đủ đa dạng nên nhóm sẽ tăng số lượng ảnh cho tập Train để cải thiện.

Một phần do quá trình Training trên máy tính cá nhân bằng CPU nên nhóm sẽ cố gắng khắc phục để có được kết quả tốt hơn

## 9. Reference

- [1] Image Super-Resolution Using Deep Convolutional Networks Chao Dong, Chen Change Loy, Member, IEEE, Kaiming He, Member, IEEE, and Xiaou Tang, Fellow, IEEE
- [2] Cui, Z., Chang, H., Shan, S., Zhong, B., Chen, X.: Deep network cascade for image super-resolution. In: European Conference on Computer Vision, pp. 49–64 (2014)
- [3] Dai, D., Timofte, R., Van Gool, L.: Jointly optimized regressors for image super-resolution. In: Eurographics. vol. 7, p. 8 (2015)