# Obfuscated Human Faces Reconstruction

Massimo Andreetta
massimo.andreetta@studenti.unipd.it

# Introduction

Human Face Reconstruction using Convolutional Neural Networks

Aim is to recover a more accurate representation of a human face from distorted or obfuscated images

Potential applications in the recovery of lost or damaged facial features and the identification of individuals in obscured photos or videos

Lack of attention to Loss Function in Image Processing

L2 loss is the default option in image reconstruction

# Pixel Loss

Pixel Loss used in Image Generation & Reconstruction

Goal is to minimize the difference between the generated and target image by comparing the values of their individual pixels

Mean Squared Error/L2 norm most commonly used method for calculating the difference

# L2 Loss Limitations

L2 loss fails to accurately predict human perception of image quality

Assumes that noise is independent of local image characteristics

Assumes white Gaussian noise, which is not always the case

These assumptions limit the effectiveness of L2 loss for predicting image quality

# Objective

Investigating alternative error metric (L1, PSNR, SSIM)

Define a new metric that combines PSNR and SSIM

Using PSNR and SSIM as evaluation metrics of the quality of image reconstruction

Analyzes the results obtained by three CNN models in detail

# Convolutional Neural Networks in Image Reconstruction

Success in computer vision tasks and effective use for denoising, deblurring, and demosaicking

Project  inspired by prior work on  image reconstruction tasks and Single Image Super-Resolution, including :

Hang Zhao et al. - “Loss Functions for Image Restoration with Neural Networks” (2018)

Jacob Conrad Trinidad. - “Reconstructing Obfuscated Human Faces” (2017)

Ledig, et al. -  “Photo-realistic single image super-resolution using a generative adversarial network.” (2017)

# Related Work

Hang Zhao et al. propose alternatives to L2 for the loss layer

Emphasized the importance of perceptually motivated losses

Introduced novel differentiable loss function based on MS-SSIM and L1

Demonstrated significant improvement in results quality with better loss functions, regardless of the network architecture

# Related Work

Trinidad examines CNNs to reconstruct obfuscated images of human faces

Investigates pixel loss and perceptual loss to improve image quality

Uses  CNN with  residual autoencoder architecture and 3 residual blocks between decoder and encoder

Ledig, et al. proposed SRResNet as the generator network for SRGAN

SRGAN can produce photo-realistic natural images for 4x upscaling factors

# Dataset

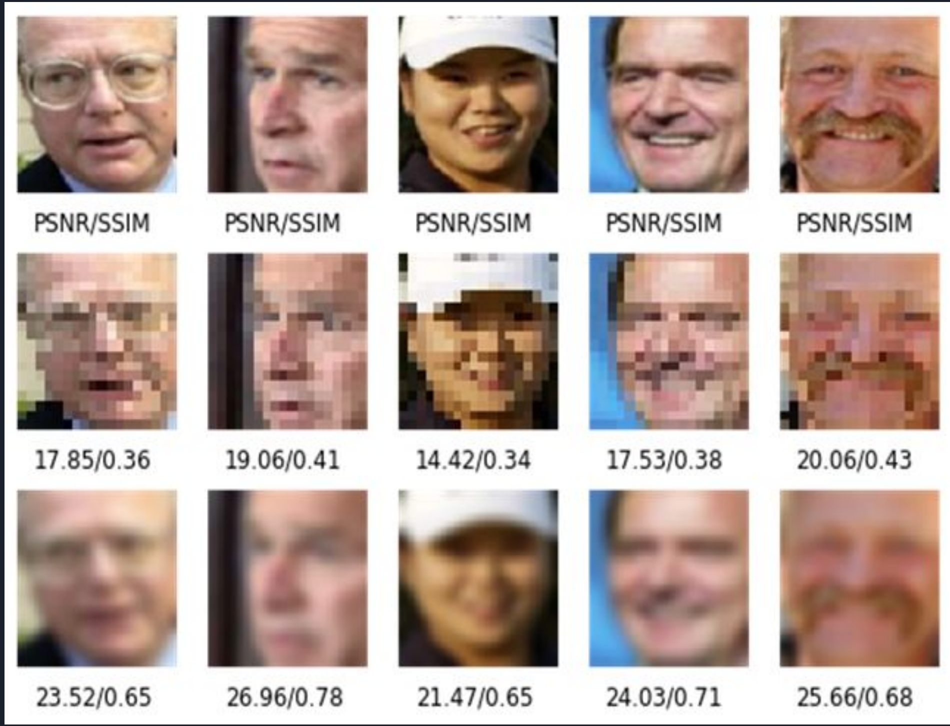
Labeled Faces in the Wild dataset

Dataset contains 13,233 images of 5,749 people, with faces already fully captured and centered

Input  images 62x47 pixels, resized to 112x112 using OpenCV

Two obfuscation methods: pixelation and Gaussian blurs

# Example



PSNR/SSIM
PSNR/SSIM
PSNR/SSIM
PSNR/SSIM
PSNR/SSIM
17.85/0.36
19.06/0.41
14.42/0.34
17.53/0.38
20.06/0.43
23.52/0.65
26.96/0.78
21.47/0.65
24.03/0.71
25.66/0.68

# Pre-processing

2500 images randomly split into 2000 training and 500 testing samples

Min-Max Scaling applied to scale the pixel values to the range [0,1]

Scaling helps improve model performance

Three CNN architectures used to reconstruct obfuscated human faces

Tested 5 different loss functions used in image reconstruction to asses the best ones

# L2 Loss Function

The Mean Squared Error/L2, is the most common error measure.

Loss obtained using the squared Euclidean distance between the pixels of the predicted output (x) and the ground truth (y)

$$L_{L2} = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)} - y^{(i)})^2$$

n = 112x112x3  number of pixels in our images.

# L1 Loss Function

Loss  obtained using the absolute Euclidean distance between the pixels of the network's predictions (x) and the ground truth (y).

$$L_{L1} = \frac{1}{n} \sum_{i=1}^{n} \left| x^{(i)} - y^{(i)} \right|$$

Networks trained with the Mean Absolute Error/ L1, perform better than L2 because it doesn't overly penalize larger errors like L2.

# PSNR Loss Function

Measure of image quality that compares the predicted image to the ground truth image

Calculates the difference between the two images as a logarithmic ratio of the MSE to the highest pixel value possible.

$$PSNR = 10 \log_{10} \frac{MAX^2}{MSE}$$

$$L_{PSNR} = 1/PSNR$$

# SSIM Loss Function

Based on the idea that the perceived quality of an image is related to the structural information present in the image

Output ranges between [-1,1]

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

$$L_{SSIM} = 1 - SSIM$$

# PSNR + SSIM Loss Function

PSNR and SSIM have poor relationship with how people perceive image quality

If used as loss functions for reconstruction tasks they can achieve even better results than L2 and L1

$$L_{PSNR+SSIM} = L_{SSIM} + 0.5 * L_{PSNR}$$

The SSIM loss is weighted twice as much as the PSNR loss in order to prioritize the quality of the restored images

# Training

Adam with learning rate = 0.0001 as optimizer

Trained with a batch size of 16 and early stopping with patience = 5

The input image of shape (112, 112, 3) undergoes a Min-Max scaling step before being fed into the model

The output image is then scaled back to the original scale using the min-max scaling values obtained previously from the input images

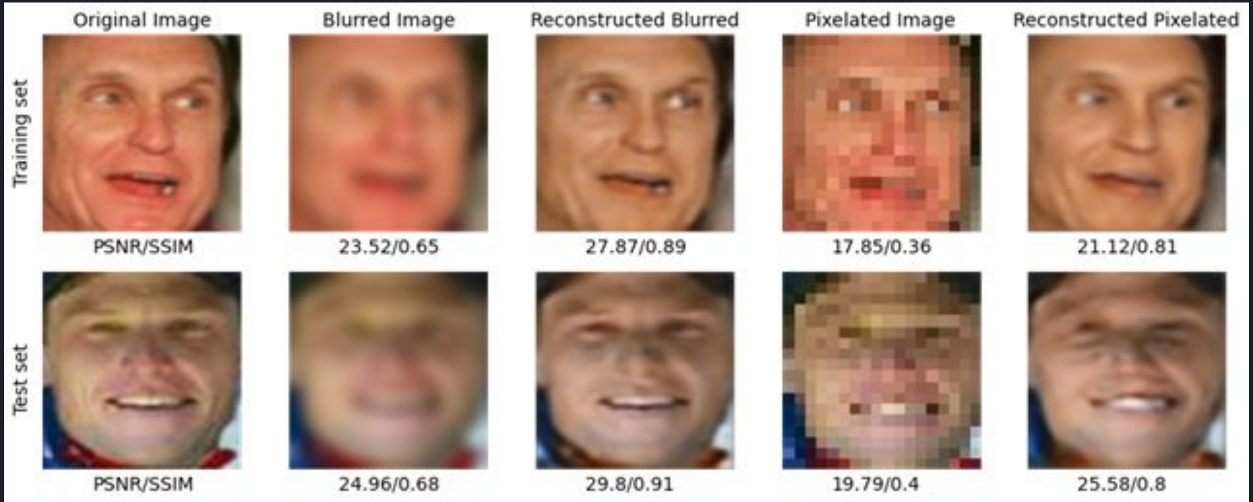
# Evaluation methods

PSNR  measure of how much the restored image differs from the original image, in terms of MSE

SSIM compares the luminance, contrast, and structure of both images to determine how similar the two images are in terms of perceptual structure

# Limitations

Models unable to include small face details like wrinkles
Better performance when reconstructing blurred images instead of pixelated ones



# Simple CNN

5 Convolutional layers:

First Convolutional layer: 128 filters 9x9

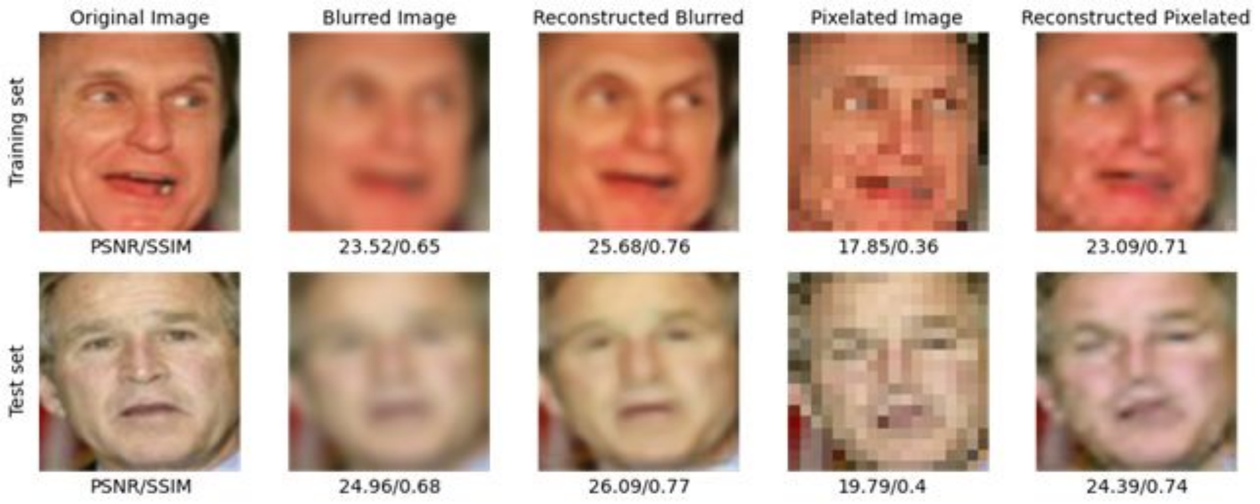
2-4 Convolutional layers: 128 filters 4x4

Each layers followed by:

- Batch Normalization layer
- Uses ReLU and zero padding

Output layer: 3 filters 4x4, sigmoid activation function

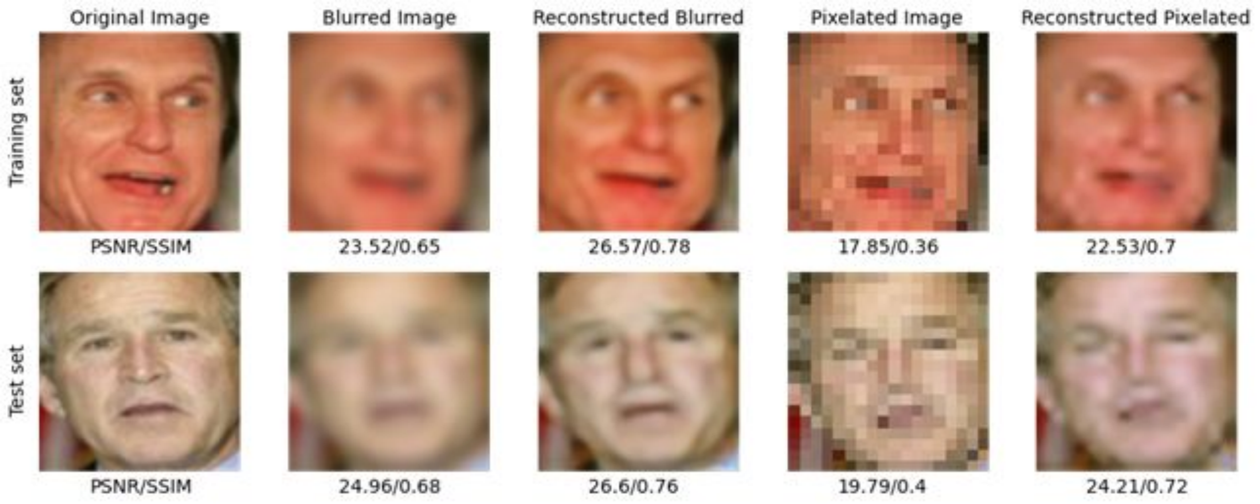
# L2 Loss




Blurred Image Reconstruction
Train Set: 26.5861 / 0.7900
Test Set: 26.5198 / 0.7868

Pixelated Image Reconstruction
Train Set: 23.9009 / 0.7562
Test Set: 23.8404 / 0.7529

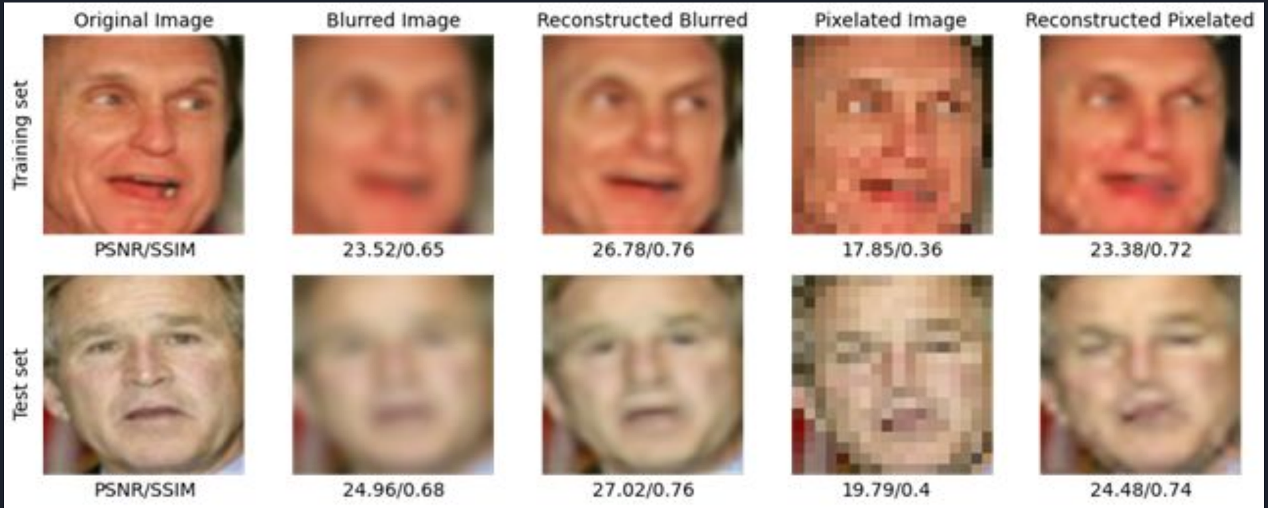
# L1 Loss




Blurred Image Reconstruction
Train Set: 26.7774 / 0.7948
Test Set: 26.7322 / 0.7918

Pixelated Image Reconstruction
Train Set: 23.7566 / 0.7513
Test Set: 23.7075 / 0.7481

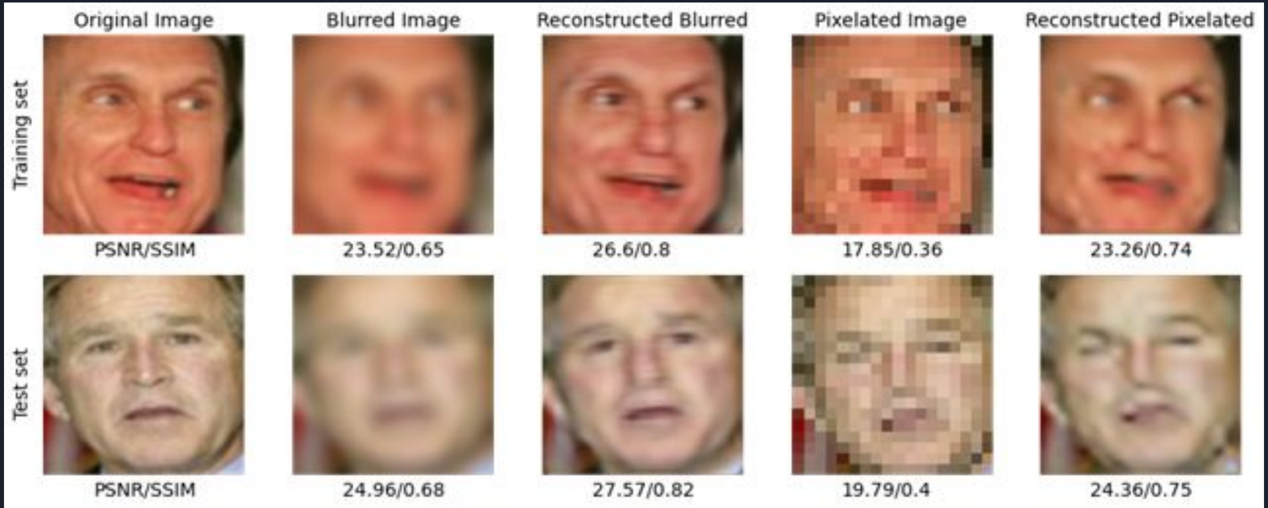
# PSNR Loss




Blurred Image Reconstruction
Train Set: 27.5200 / 0.7966
Test Set: 27.4431 / 0.7934

Pixelated Image Reconstruction
Train Set: 24.2428 / 0.7623
Test Set: 24.1503 / 0.7588

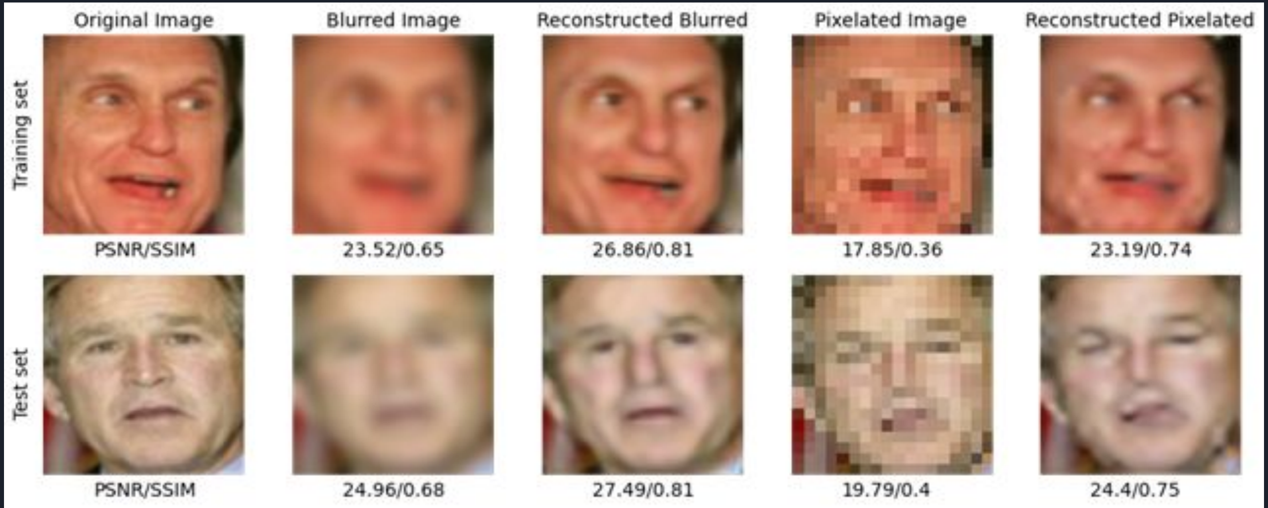
# SSIM Loss




Blurred Image Reconstruction
Train Set: 27.4119 / 0.8369
Test Set: 27.3281 / 0.8326

Pixelated Image Reconstruction
Train Set: 24.1874 / 0.7766
Test Set: 24.0753 / 0.7715

# PSNR + SSIM Loss

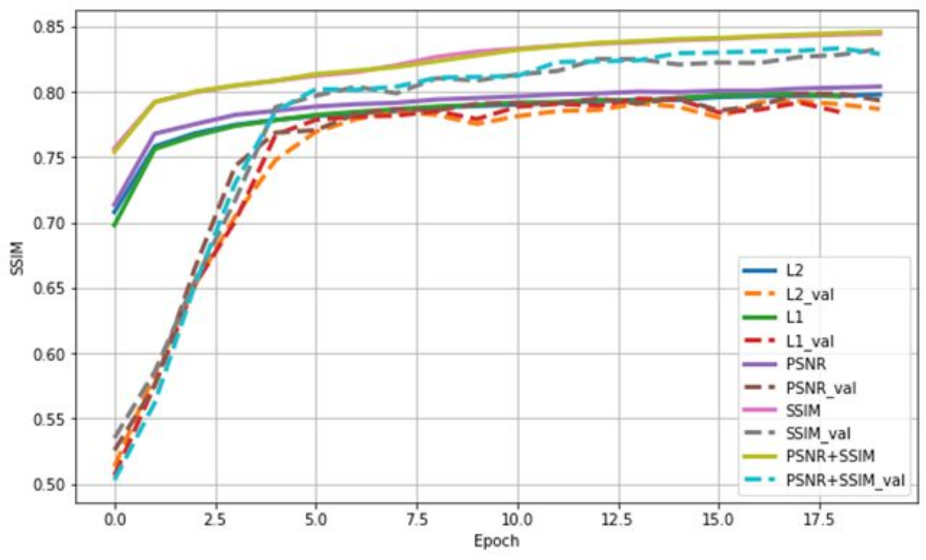



Blurred Image Reconstruction
Train Set: 27.8359 / 0.8336
Test Set: 27.7357 / 0.8290

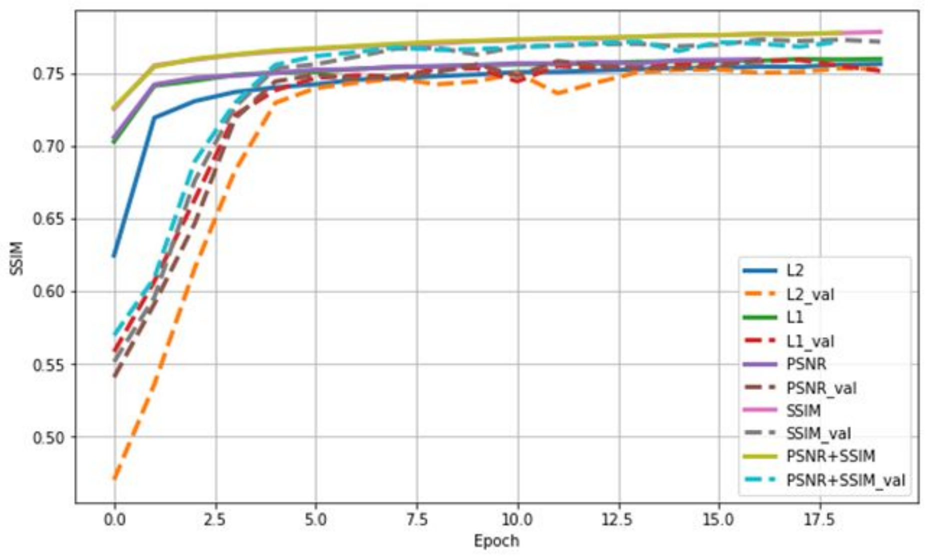
Pixelated Image Reconstruction
Train Set: 23.9799 / 0.7763
Test Set: 23.9083 / 0.7715

# Training and Evaluation

## Blurred Images




## Pixelated Images



# SRResNet

Removed 2 upsampling block at the end of the network

First Convolutional layer: 64 filters 9x9 + PReLU

5 Residual blocks, where the output of each block is added to its input

Each residual block consists:

- Convolutional layer: 64 filters 3x3
- Batch Normalization layer
- PReLU
- Convolutional layer: 64 filters 3x3
- Batch Normalization layer

# SRResNet

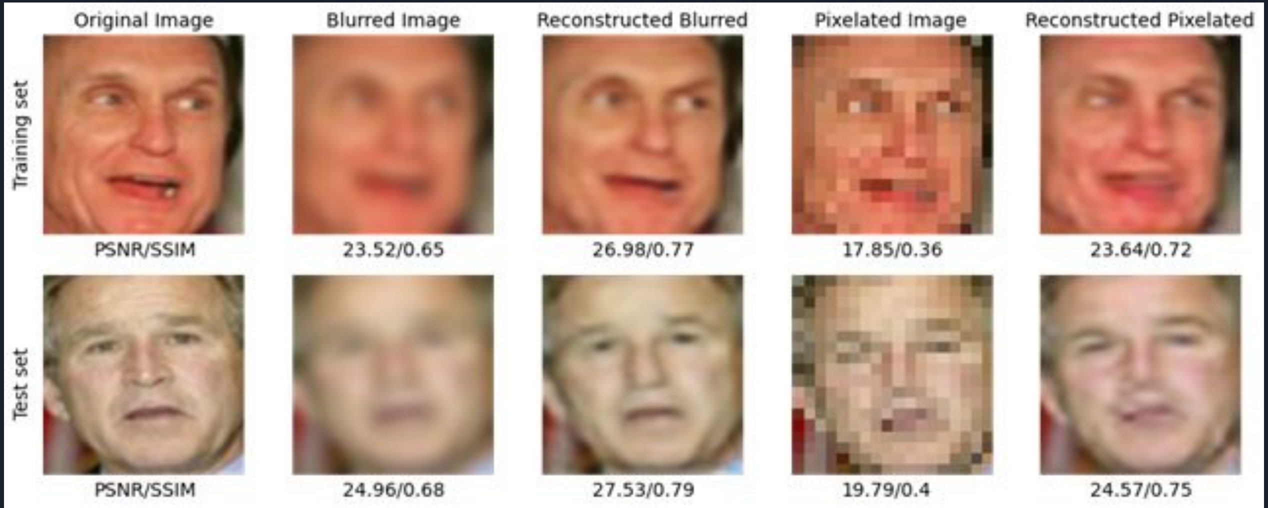
1 Convolutional layer: 64 filters 9x9

1 Batch Normalization layer

Output of this layer added to the output of the initial Convolutional layer

Output layer: 3 filters 9x9 + sigmoid activation function

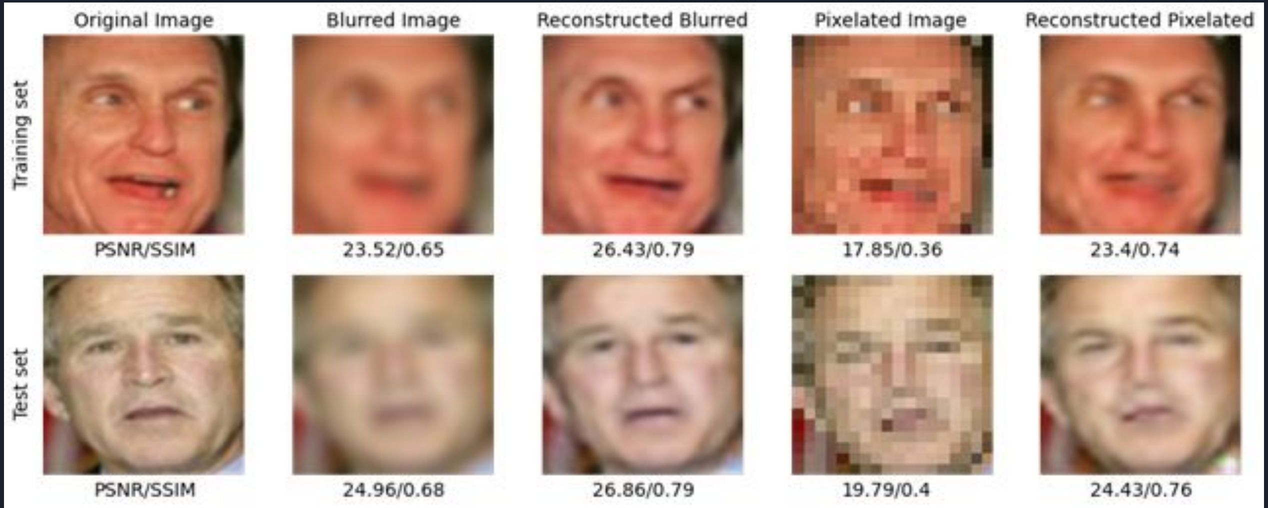
# L2 Loss




Blurred Image Reconstruction
Train Set: 27.9575 / 0.8101
Test Set: 27.7065 / 0.8000

Pixelated Image Reconstruction
Train Set: 24.3446 / 0.7650
Test Set: 24.1000 / 0.7555

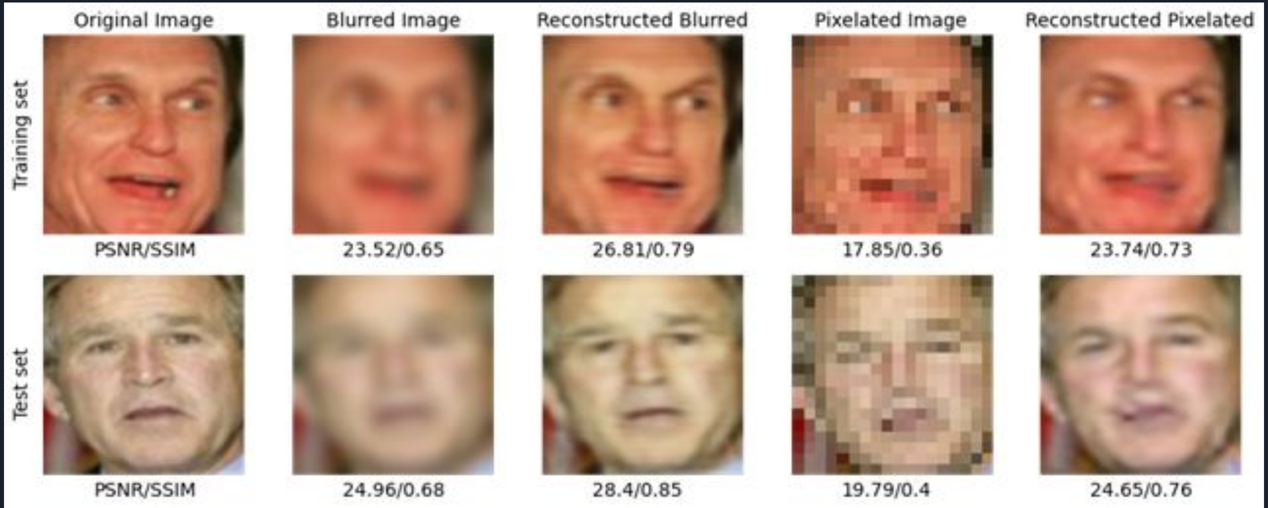
# L1 Loss




Blurred Image Reconstruction
Train Set: 27.3167 / 0.8171
Test Set: 27.1332 / 0.8076

Pixelated Image Reconstruction
Train Set: 24.1818 / 0.7813
Test Set: 24.0060 / 0.7729

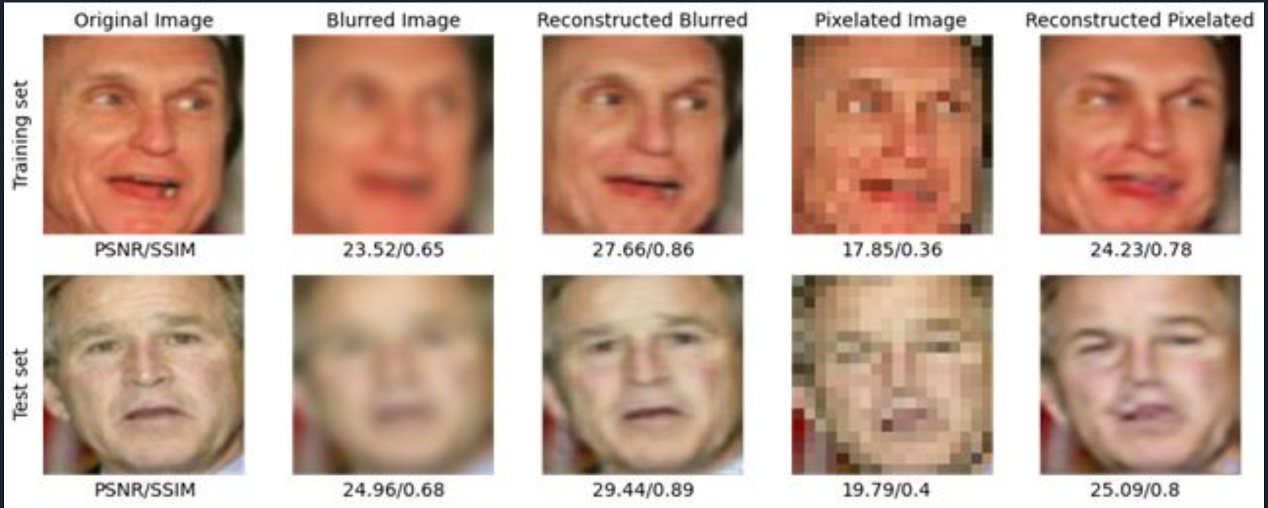
# PSNR Loss




Blurred Image Reconstruction
Train Set: 28.1864 / 0.8380
Test Set: 27.9722 / 0.8301

Pixelated Image Reconstruction
Train Set: 24.4730 / 0.7748
Test Set: 24.2071 / 0.7652

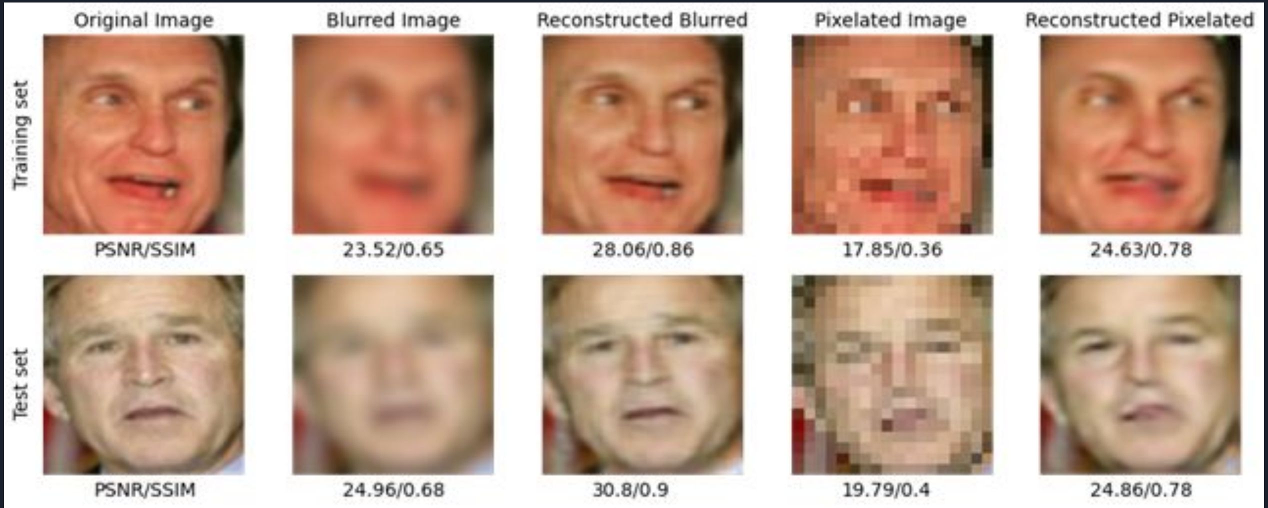
# SSIM Loss




Blurred Image Reconstruction
Train Set: 28.5495 / 0.8878
Test Set: 28.4434 / 0.8806

Pixelated Image Reconstruction
Train Set: 24.5532 / 0.8052
Test Set: 24.2078 / 0.7841

# PSNR + SSIM Loss

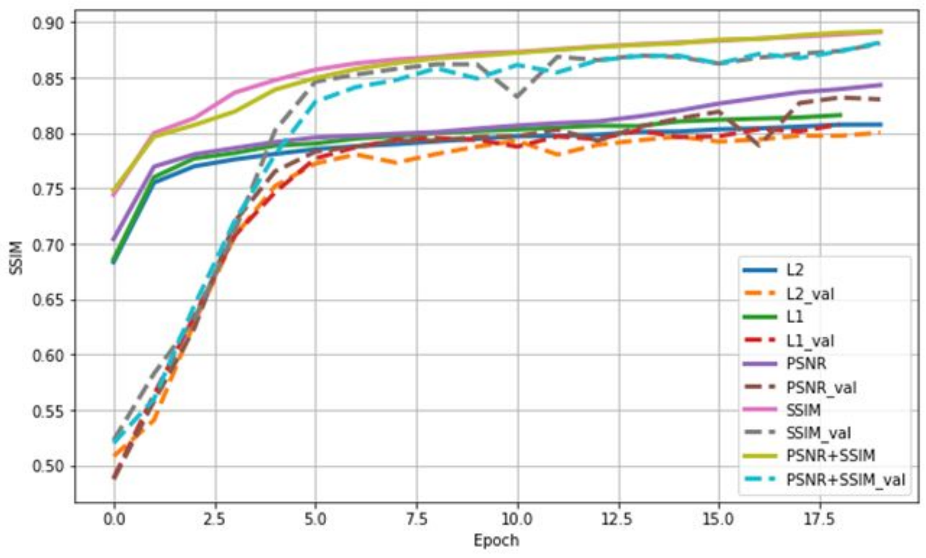



Blurred Image Reconstruction
Train Set: 29.2775 / 0.8892
Test Set: 29.1476 / 0.8817

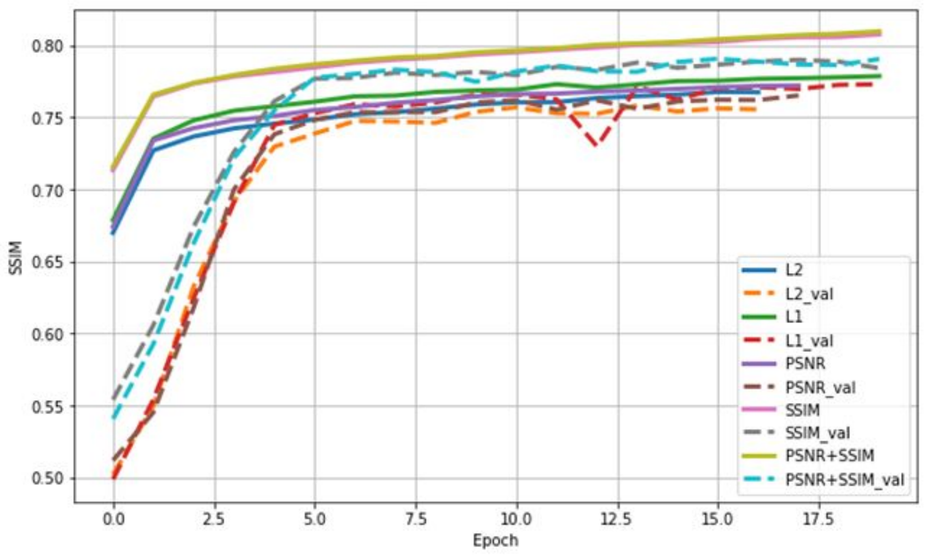
Pixelated Image Reconstruction
Train Set: 24.9042 / 0.8109
Test Set: 24.5279 / 0.7905

# Training and Evaluation

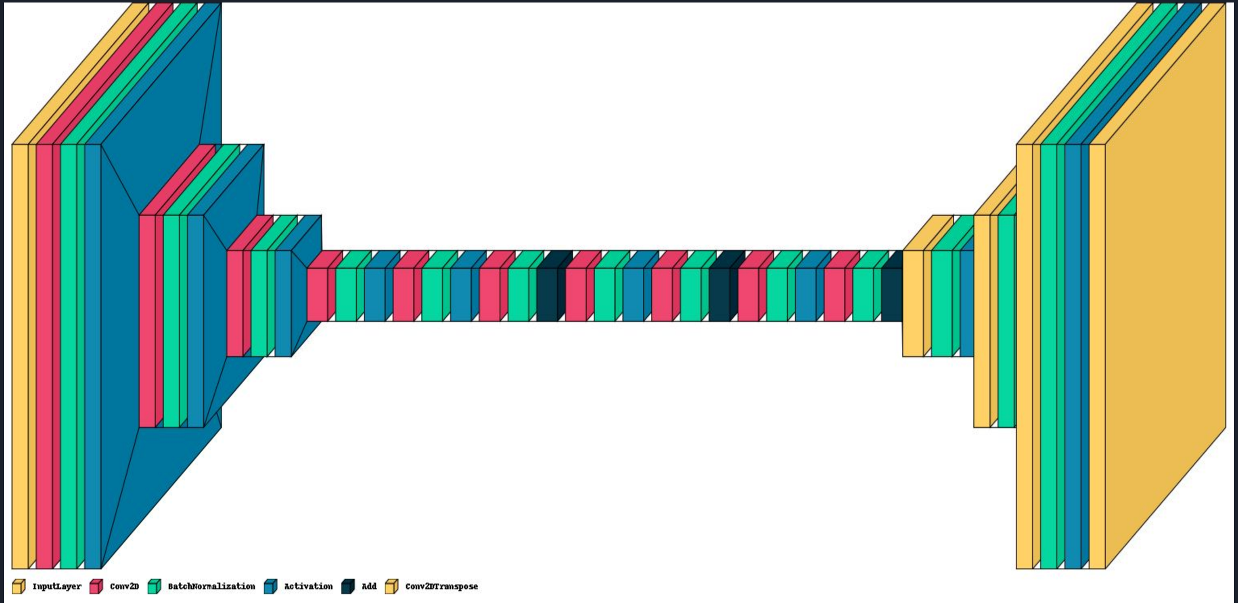
## Blurred Images




## Pixelated Images



# Residual Autoencoder




Multiple Convolutional and Transpose Convolutional layers, each followed by Batch Normalization layer and ReLu

# Residual Autoencoder

First Convolutional layer: 64 filters 9x9

2-4 Convolutional layers: 64-128-256 filters 4x4, stride=2

3 Residual blocks followed by an element-wise addition with input tensor

Each residual block consists:

- Convolutional layer: 64 filters 3x3
- Batch Normalization layer
- ReLU
- Convolutional layer: 64 filters 3x3
- Batch Normalization layer

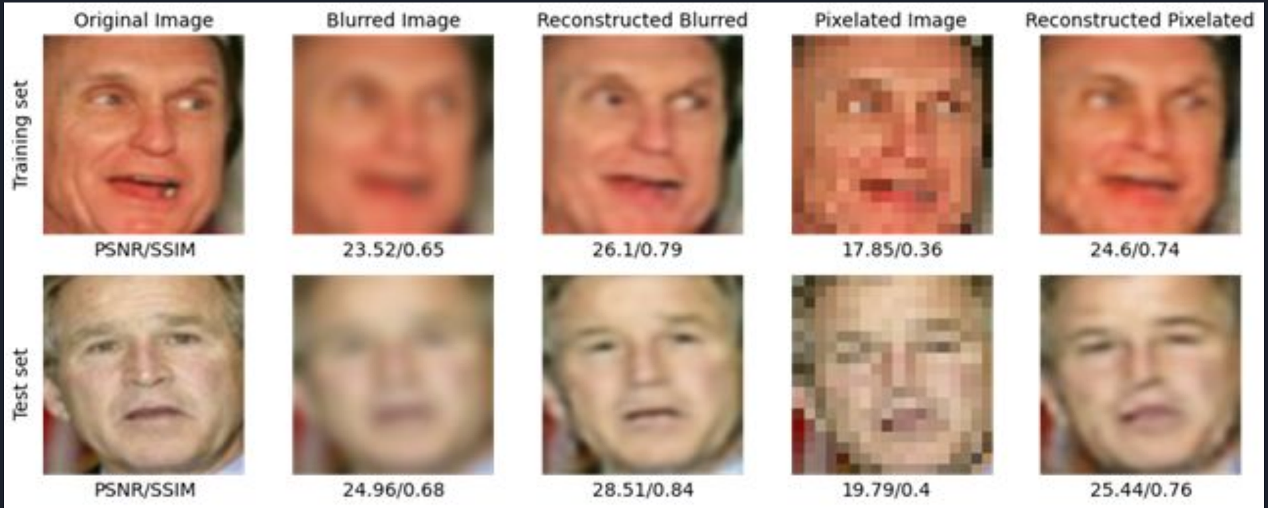
# Residual Autoencoder

4 Transpose Convolutional layers:

- 1-3 layers: 256-128-64 filters 4x4, stride = 2
- Last: 3 filters 9x9, stride = 1 and sigmoid activation function

Changed from tanh to sigmoid to account for the use of Min-Max Scaling

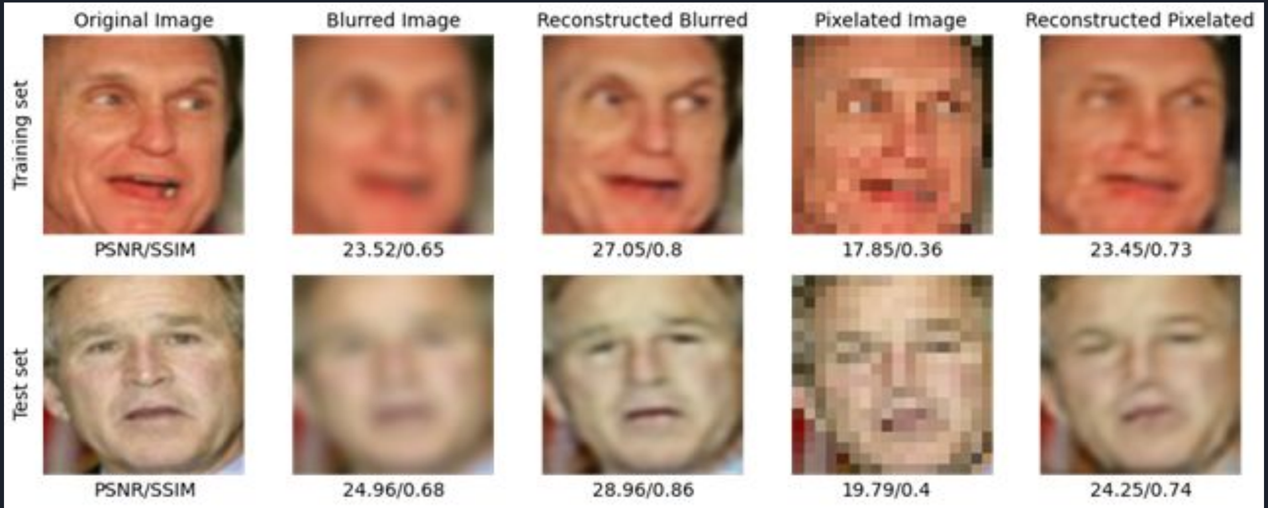
# L2 Loss




Blurred Image Reconstruction
Train Set: 27.5846 / 0.8301
Test Set: 27.3430 / 0.8224

Pixelated Image Reconstruction
Train Set: 26.1573 / 0.7866
Test Set: 24.3998 / 0.7492

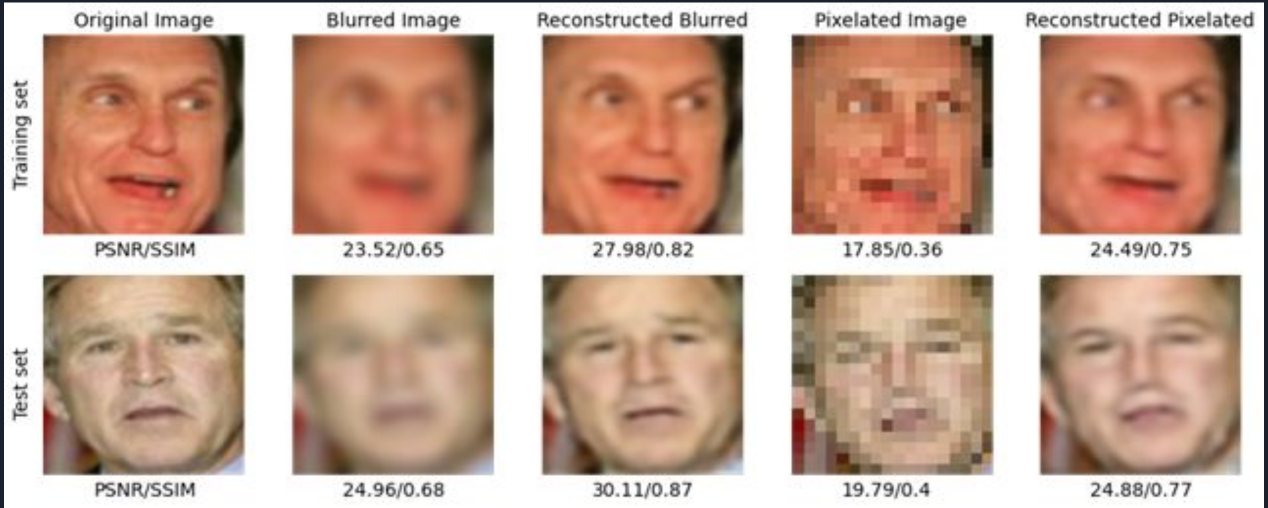
# L1 Loss




Blurred Image Reconstruction
Train Set: 28.0884 / 0.8368
Test Set: 27.9211 / 0.8301

Pixelated Image Reconstruction
Train Set: 24.2364 / 0.7689
Test Set: 23.9235 / 0.7541

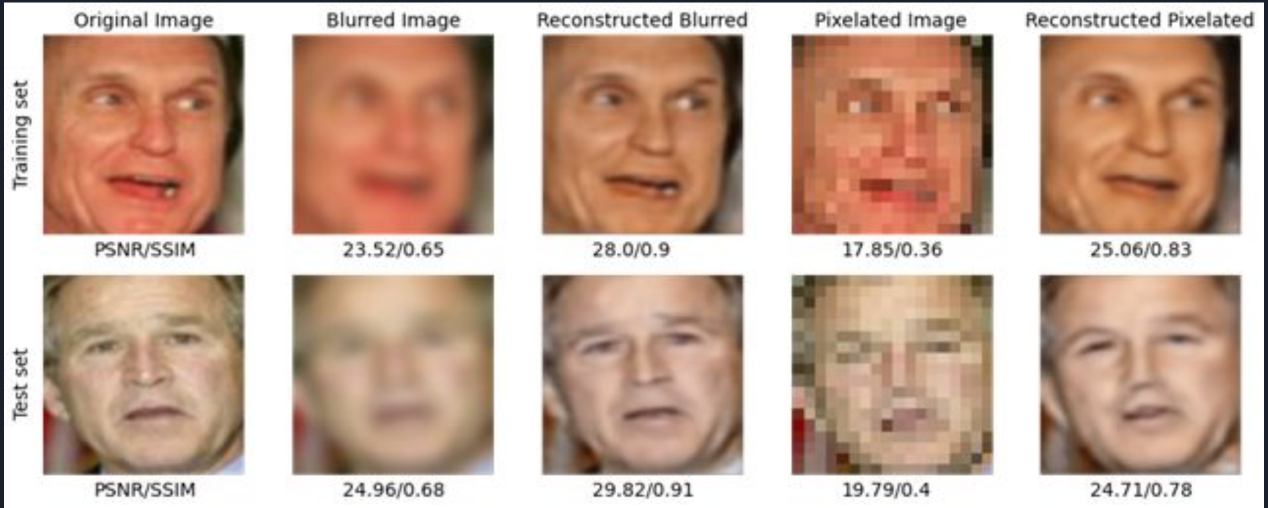
# PSNR Loss




Blurred Image Reconstruction
Train Set: 29.3583 / 0.8544
Test Set: 29.0591 / 0.8459

Pixelated Image Reconstruction
Train Set: 25.5830 / 0.7902
Test Set: 24.5705 / 0.7664

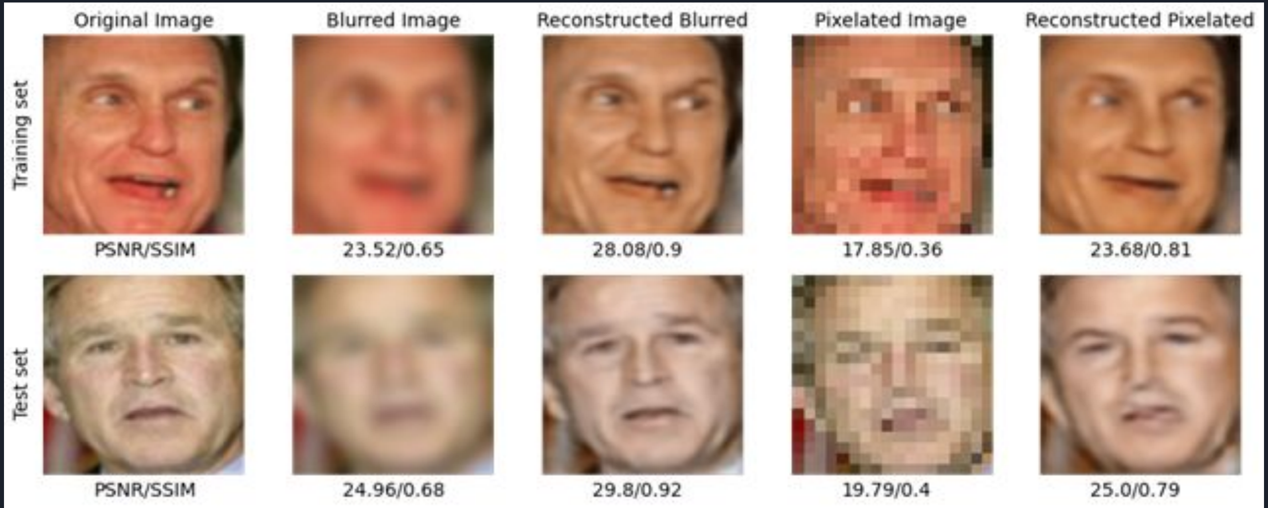
# SSIM Loss




Blurred Image Reconstruction
Train Set: 28.3404 / 0.9127
Test Set: 28.0636 / 0.8967

Pixelated Image Reconstruction
Train Set: 25.3908 / 0.8612
Test Set: 23.9914/ 0.7876

# PSNR + SSIM Loss

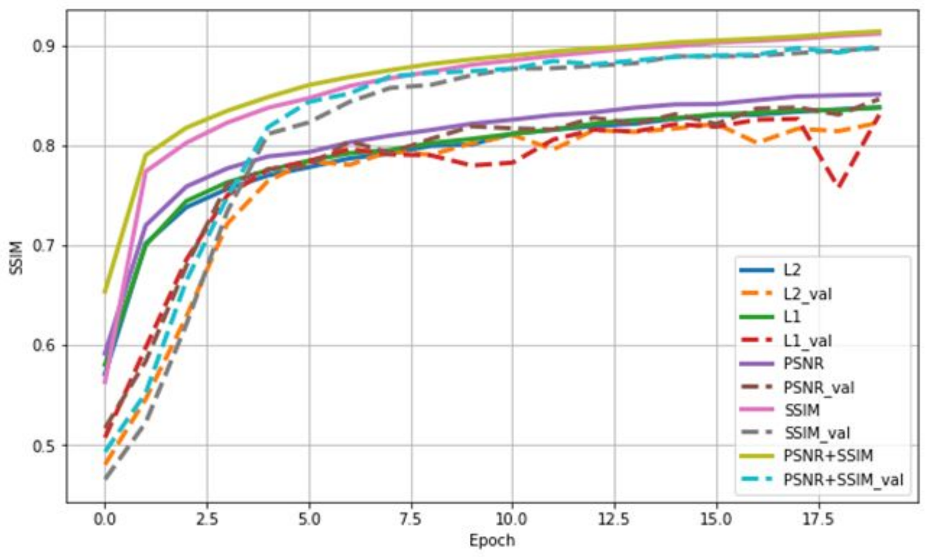



Blurred Image Reconstruction
Train Set: 28.6523 / 0.9155
Test Set: 28.3727 / 0.8990

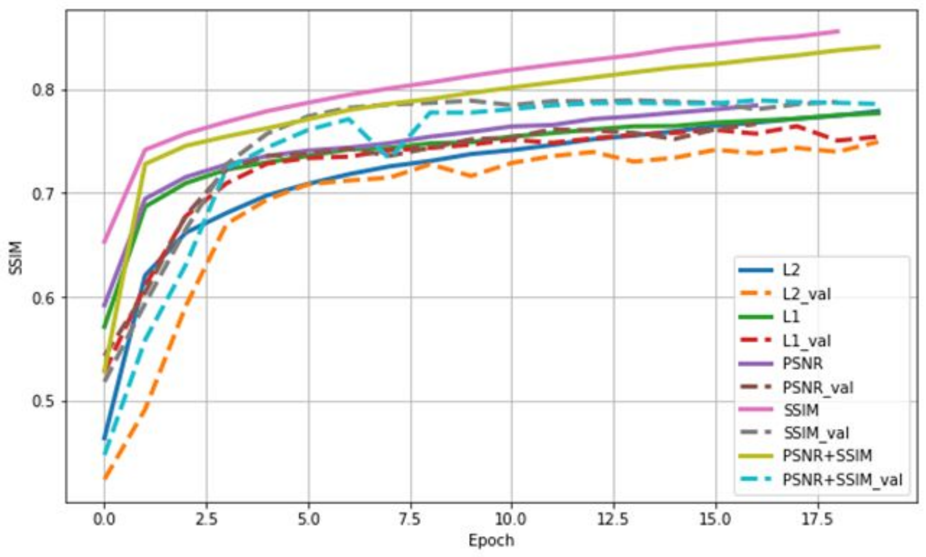
Pixelated Image Reconstruction
Train Set: 25.2175 / 0.8464
Test Set: 24.0194/ 0.7857

# Training and Evaluation

## Blurred Images




## Pixelated Images



# Results Comparison

| L2 Loss | Blurred Images | | | |
|---|---|---|---|---|
| Architecture | Train set PSNR / SSIM | | Test set PSNR / SSIM | |
| Simple CNN | 26.5861 | 0.7900 | 26.5198 | 0.7868 |
| SRResNet | 27.9575 | 0.8101 | 27.7065 | 0.8000 |
| Original Residual Autoencoder | 26.56 | 0.700 | 25.96 | 0.696 |
| Residual Autoencoder | 27.5846 | **0.8301** | 27.3430 | **0.8224** |
| | Pixelated images | | | |
| Architecture | Train set PSNR / SSIM | | Test set PSNR / SSIM | |
| Simple CNN | 23.9009 | 0.7562 | 23.8404 | 0.7529 |
| SRResNet | 24.3446 | 0.7650 | 24.1000 | **0.7555** |
| Original Residual Autoencoder | 23.94 | 0.600 | 23.52 | 0.585 |
| Residual Autoencoder | 26.1573 | **0.7866** | 24.3998 | 0.7492 |

| L1 Loss | Blurred Images | | | |
|---|---|---|---|---|
| Architecture | Train set PSNR / SSIM | | Test set PSNR / SSIM | |
| Simple CNN | 26.7774 | 0.7948 | 26.7322 | 0.7918 |
| SRResNet | 27.3167 | 0.8171 | 27.1332 | 0.8076 |
| Residual Autoencoder | 28.0884 | **0.8368** | 27.9211 | **0.8301** |
| | Pixelated images | | | |
| Architecture | Train set PSNR / SSIM | | Test set PSNR / SSIM | |
| Simple CNN | 23.7566 | 0.7513 | 23.7075 | 0.7481 |
| SRResNet | 24.1818 | **0.7813** | 24.0060 | **0.7729** |
| Residual Autoencoder | 24.2364 | 0.7689 | 23.9235 | 0.7541 |

# Results Comparison

| PSNR | Blurred Images | | | |
|---|---|---|---|---|
| Architecture | Train set PSNR / SSIM | | Test set PSNR / SSIM | |
| Simple CNN | 27.5200 | 0.7966 | 27.4431 | 0.7934 |
| SRResNet | 28.1864 | 0.8380 | 27.9722 | 0.8301 |
| Residual Autoencoder | 29.3583 | **0.8544** | 29.0591 | **0.8459** |
| | Pixelated images | | | |
| Architecture | Train set PSNR / SSIM | | Test set PSNR / SSIM | |
| Simple CNN | 24.2428 | 0.7623 | 24.1503 | 0.7588 |
| SRResNet | 24.4730 | 0.7748 | 24.2071 | 0.7652 |
| Residual Autoencoder | 25.5830 | **0.7902** | 24.5705 | **0.7664** |

| SSIM | Blurred Images | | | |
|---|---|---|---|---|
| Architecture | Train set PSNR / SSIM | | Test set PSNR / SSIM | |
| Simple CNN | 27.4119 | 0.8369 | 27.3281 | 0.8326 |
| SRResNet | 28.5495 | 0.8878 | 28.4434 | 0.8806 |
| Residual Autoencoder | 27.0003 | **0.9108** | 26.8274 | **0.8946** |
| | Pixelated images | | | |
| Architecture | Train set PSNR / SSIM | | Test set PSNR / SSIM | |
| Simple CNN | 24.1874 | 0.7766 | 24.0753 | 0.7715 |
| SRResNet | 24.5532 | 0.8052 | 24.2078 | 0.7841 |
| Residual Autoencoder | 25.3908 | **0.8612** | 23.9914 | **0.7876** |

# Results Comparison

| PSNR+SSIM | Blurred Images | | | |
|---|---|---|---|---|
| Architecture | Train set PSNR / SSIM | | Test set PSNR / SSIM | |
| Simple CNN | 27.8359 | 0.8336 | 27.7357 | 0.8290 |
| SRResNet | 29.2775 | 0.8892 | 29.1476 | 0.8817 |
| Residual Autoencoder | 28.6523 | **0.9155** | 28.3727 | **0.8990** |
| | Pixelated images | | | |
| Architecture | Train set PSNR / SSIM | | Test set PSNR / SSIM | |
| Simple CNN | 23.9799 | 0.7763 | 23.9083 | 0.7715 |
| SRResNet | 24.9042 | 0.8109 | 24.5279 | **0.7905** |
| Residual Autoencoder | 25.2175 | **0.8464** | 24.0194 | 0.7857 |

# Conclusion and future works

Focus of the project was on loss functions, which are often overlooked when exploring image restoration using neural networks

Proposed several alternative loss functions:  L1, PSNR, and SSIM

Analyzed their effects on 3 different CNN architectures to show their improved performance on the networks

Improved performance when using SSIM loss compared to L2

Introduced novel loss combining SSIM and  PSNR

Future work: investigating impact of state-of-the-art architectures like SRGAN, and exploring use of perceptual loss and Facial Attention Loss

Thank you for the attention