UEMF
الجامعة الأورومتوسطية بفاس
EUROMED UNIVERSITY OF FES
UNIVERSITÉ EUROMED DE FÈS

EIDIA
École d'Ingénierie Digitale
et d'Intelligence Artificielle

PFA Project Presentation

Image Restoration using Deep Learning

Elaborated by:
Kaoutar Lakdim
Chaimae Biyaye
Anas Belmessid
Monsef Berjaoui
Manal El kassim
Olaya Latoubi

Supervised by:
Dr.Alae Ammour

school year : 2022/2023

Outline

1. **Introduction**
2. **Overview of the Project**
3. **Traditional methods for image restoration**
   - Traditional methods for image restoration
4. **Different Architectures using deep learning**
5. **Our Model**
   - Data Description
   - DnCnn Architecture
   - CVT Architecture
   - PixNet Architecture
   - Model Evaluation
6. **Results Analysis**
7. **Conclusion**

Problematic

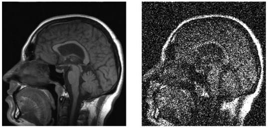
Types of noise

(a) Original Image (b) Noisy Image (c) Overregularised TV (d) Residual (Abs. Value)

(e) Filtered Residual (f) Gamma (Scaled) (g) TV_{pwL} Reconstruction SSIM = 0.796 (h) TGV Reconstruction SSIM = 0.801

Different types of noise applied to the same image

Domains of Application



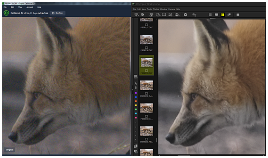
(a) Brain MR Image without noise (b) Brain MR Image with noise

Medical Field

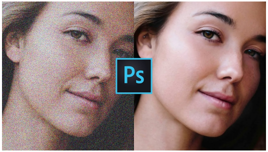
Historical Documents

Astronomy

Aerospace Field



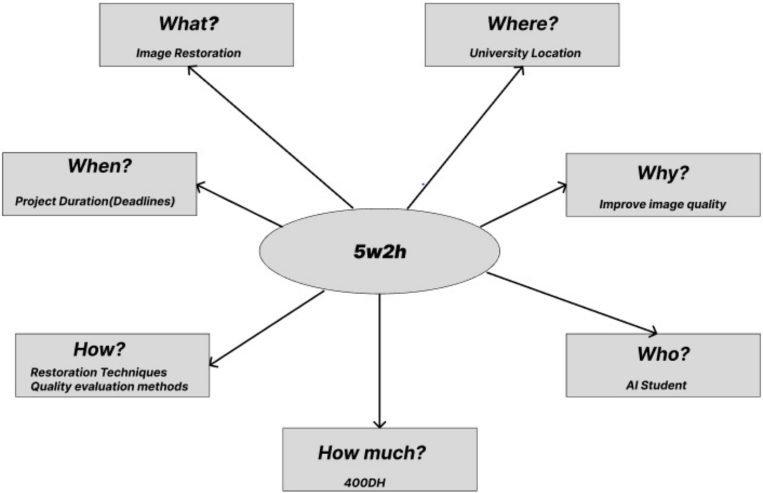
Videos



Artwork and Photography

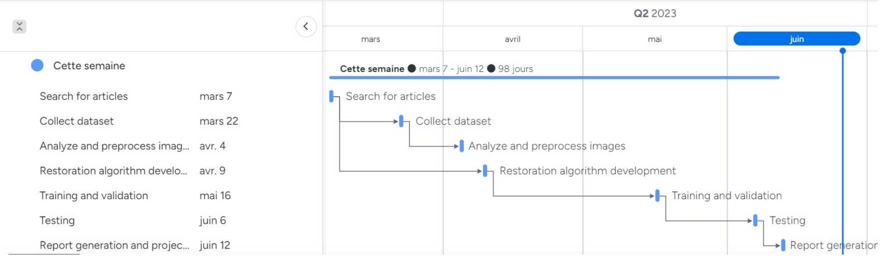
Overview of the Project

5W2H Diagram



Overview of the project

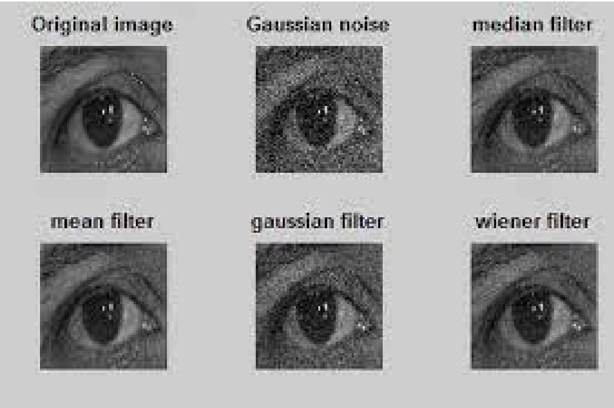
Gant Diagram



Traditional methods for image restoration

Traditional methods for image restoration

- Filters: Various filters, such as median filters, mean filters, and Gaussian filters and weiner filters.

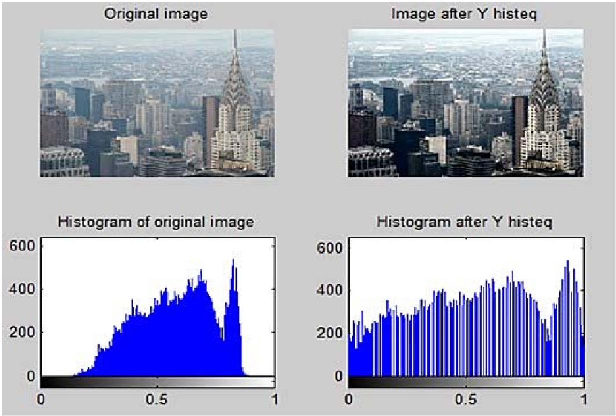


Traditional methods for image restoration

- Morphological Operations:Operations like erosion, dilation, opening, and closing.

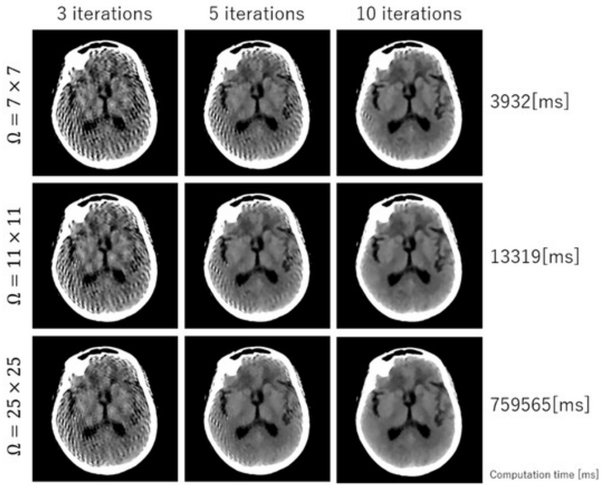
Traditional methods for image restoration

- Histogram Equalization:redistributing the pixel intensities.



Traditional methods for image restoration

- Variation (TV) Regularization: Mathematically, the TV of an image can be computed as the sum of the absolute differences between adjacent pixel values across the image. It can be defined as: TV(I) = —I—

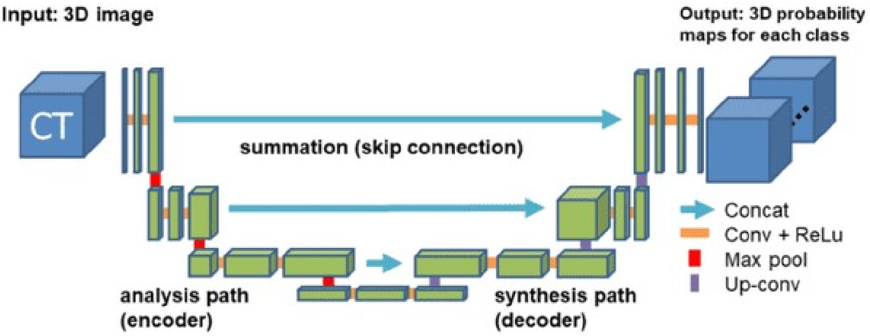


Traditional methods for image restoration

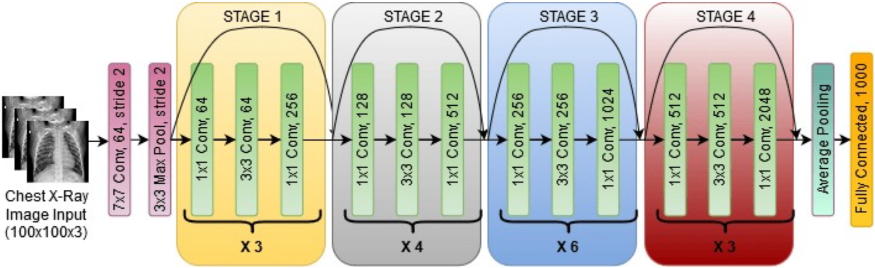
- Blind Deconvolution:Estimating the blur kernel.

Different Architectures using deep learning

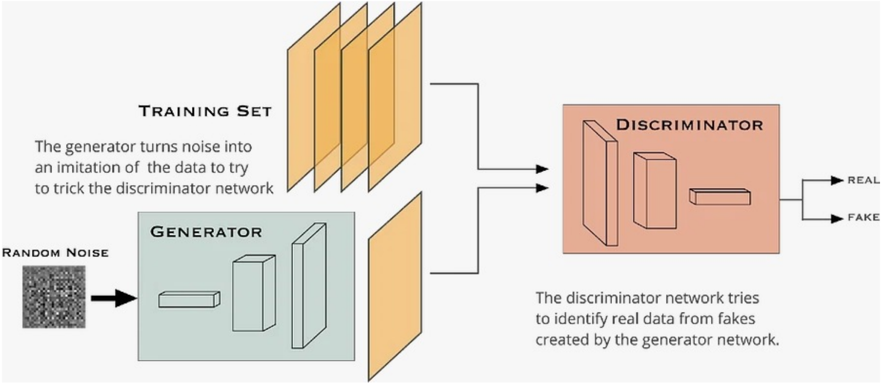
U-Net Architecture



ResNet Architecture

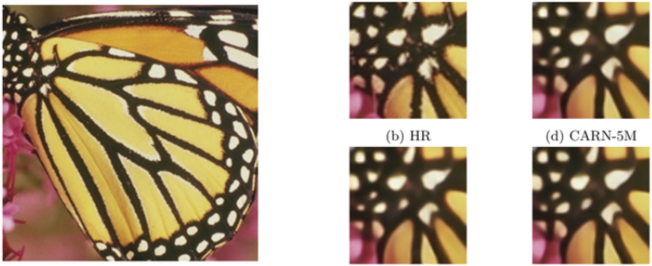


GAN Architecture



Our Model

Data Description



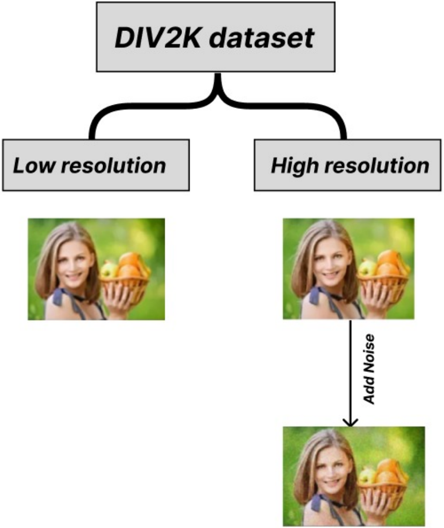
(a) Original butterfly.png
(b) HR
(d) CARN-5M
(c) Baseline
(e) MSRResNet20

DIV2K 1000 images

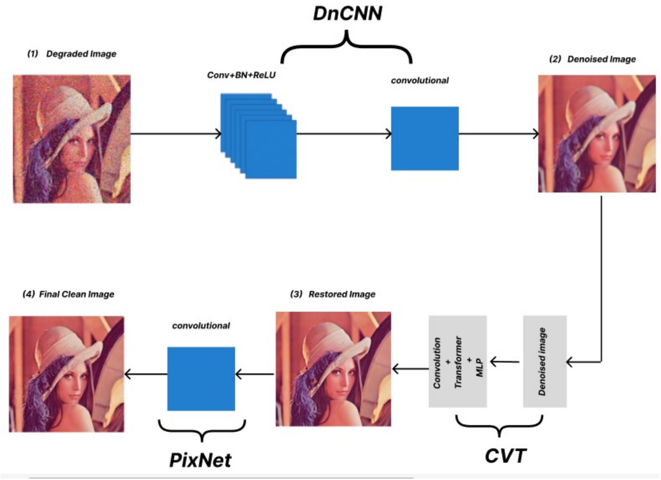
Resolution often exceeding 2K (2048x1080)

even 4K (3840x2160)

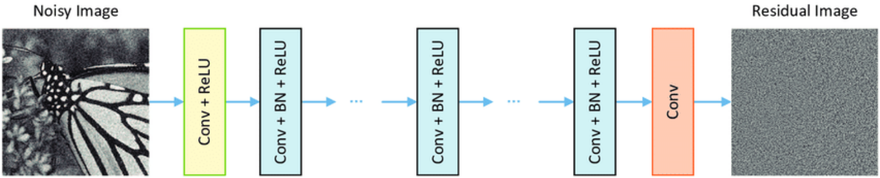
Data Description



Global Architecture:



DnCNN Architecture

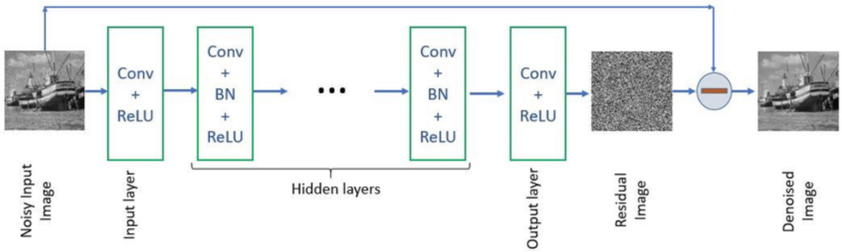


Convolutional Layer:Features Extraction

Relu Activation:Elimination of negative pixels

Batch normalization:Stability during the train

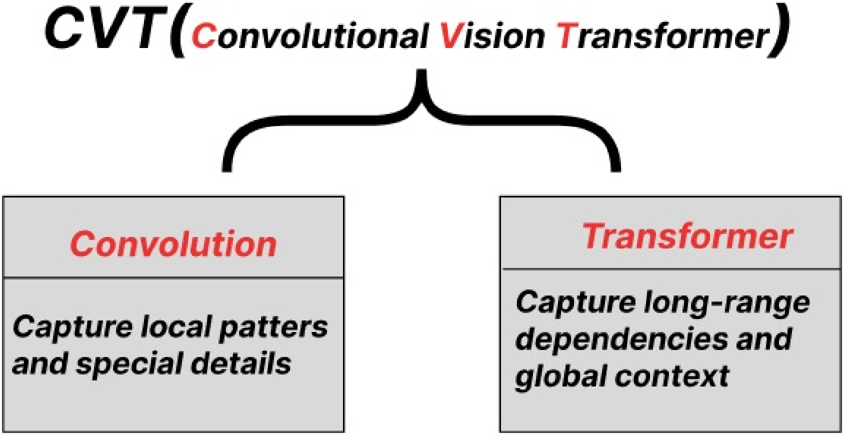
DnCNN Architecture



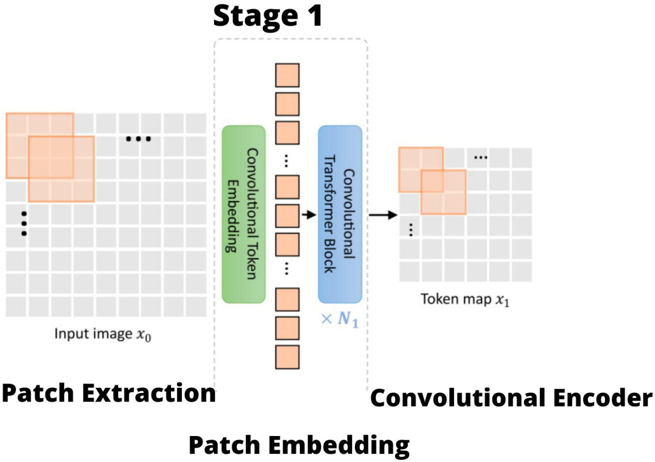
Residual Learning:Residual Image

Output:The denoised Image

CVT Architecture

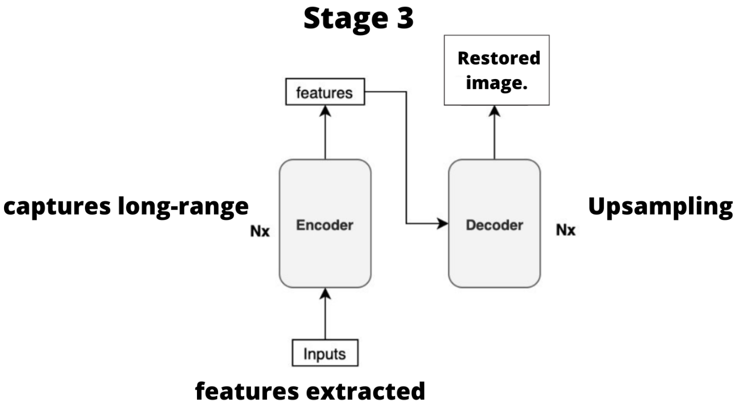


CVT Architecture



Embedding:Features Extraction suitable for the input of the transformer

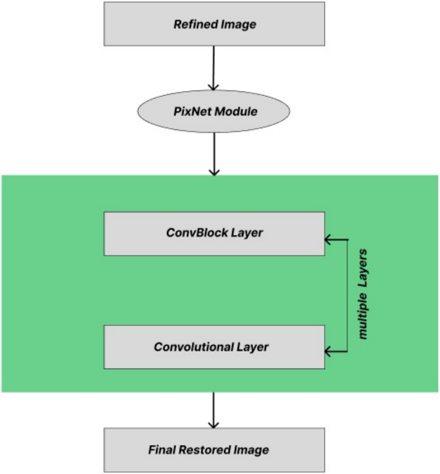
CVT Architecture



Transformer :Capture special relationships between pixels

PixNet Architecture

Enhancement of the quality:



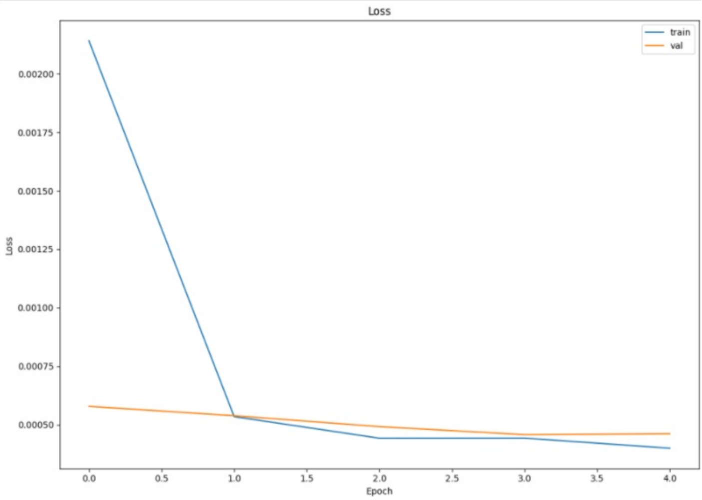
Model Evaluation

$$\mathrm{PSNR} = 10 \times \lg\left(\frac{255^2}{\mathrm{MSE}}\right)$$

$$\mathrm{MSE} = \frac{1}{M \times N}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[I(i,j) - I'(i,j)\right]^2$$

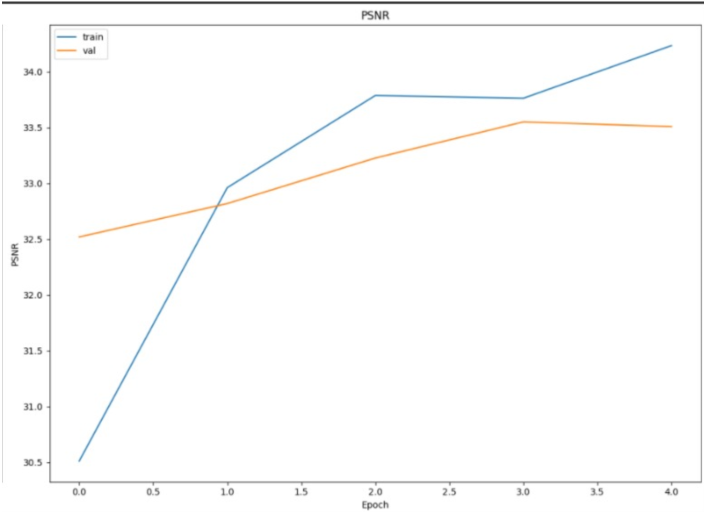
Results Analysis

Training Curves



Loss Curve

Training Curves



PSNR Curve

Comparison of our Model with Previous Models

Metrics of our Models:

	train_loss	val_loss	train_psnr	val_psnr
0	0.002141	0.000578	30.512136	32.520647
1	0.000535	0.000539	32.962687	32.820274
2	0.000442	0.000492	33.788875	33.228106
3	0.000443	0.000458	33.763961	33.552264
4	0.000400	0.000462	34.235739	33.509705

Comparison of our Model with Previous Models

Performance Metrics of different Architectures for Image Restoration:

Architecture	**Train_Loss**	**Val_Loss**
Single-layer Perceptron	0.320	0.530
Autoencoder	0.390	0.600
ResNet	0.430	0.640
U-Net	0.480	0.690
GAN (Generative Adversarial Network)	0.500	0.700

Comparison of our Model with Previous Models

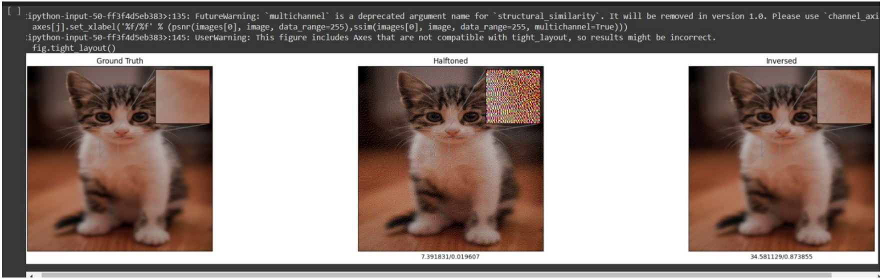
Performance Metrics of different Architectures for Image Restoration:

Architecture	Train PSNR	Val PSNR
Single-layer Perceptron	16.6	13.9
Autoencoder	14.8	12.3
ResNet	0.430	0.640
U-Net	12.6	10.3
GAN (Generative Adversarial Network)	12.3	10.1

Demo

Conclusion

Conclusion


:ipython-input-50-ff3f4d5eb383>:135: FutureWarning: `multichannel` is a deprecated argument name for `structural_similarity`. It will be removed in version 1.0. Please use `channel_axi
axes[j].set_xlabel('%f/%f' % (psnr(images[0], image, data_range=255),ssim(images[0], image, data_range=255, multichannel=True)))
:ipython-input-50-ff3f4d5eb383>:145: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.
fig.tight_layout()
Ground Truth
Halftoned
Inversed
7.391831/0.019607
34.581129/0.873855