

PROJECT REPORT

Image Denoising (Using CNN)

Image denoising is a technique used in image processing and computer vision to remove noise from a noisy image to recover the original, clean image to create a sharp and clear image. Noise in an image refers to the random variation of brightness or color information. It is typically unwanted visual distortion that can obscure important details or degrade the quality of the image.

Image denoising is primarily employed in medical imaging, where the final image often contains significant noise. This noise can occur due to the malfunctioning of machines or the precautions taken to protect patients from excessive radiation.

Objective

The primary objective of the image denoising project is to develop a deep learning model that can effectively remove noise from low-resolution images to restore their original high-resolution quality.

The key goals of this project include:

- 1.Noise Reduction
- 2.Image Quality Improvement
- 3.Effective Model
- 4.Performance Evaluation

Contents of this Notebook

1. DATA COLLECTION,LOADING AND SPLITTING
2. ADDING NOISE TO THE IMAGE
3. IMPLEMENTING CONVOLUTIONAL NEURAL NETWORK COMPILING THE MODEL

4. DEFINIGN EPOCHS AND OPTIMIZER
5. IMPLEMENTING DATA AUGMENTATION
6. TRAIN THE CNN MODEL
7. EVALUATING THE MODEL USING PSNR, MSE AND MAE
8. SAVING PREDICTED IMAGES,PLOT AND DISPLAY

Methodology

1. Data Collection and Preparation:

- Data Collection: Used the dataset which was given in zip file.
- Data Loading and Normalizing: Loaded the dataset into collab using OpenCV and resized them to (128*128 pixels) and normalized the pixel values in range of [0,1] for better performance during traing.
- Data Spliting:Splitting the dataset into Training,Validation and Test datasets for development and evaluation of model.
- Adding Noise: Adding noise into the low-resolution images to simulate real-world noise and make the denoising task more challenging.

2. Model Architecture:

- Convolutional Neural Network (CNN): Designing a deep learning model consisting of convolutional layers for feature extraction and upsmaling layers for image reconstruction.
- Using Encoderfor compression of input image into latent space using convolutional layers and max-pooling.Used four Conv2D layers with each layer uses 2 strides to reduce the spatial dimentions and then used MAx-pooling for further reduction of dimentions.
- And Decoder for reconstructing the image from latent space back to original size using upsmaling and convolutional layers.Used four Conv2DTranspose Layers and Upsampling2D layers with each of 2 strides.
- Mean Squared Error: This loss function is used to measure the difference between original high resolution and reconstructed image.
- Optimizer: Used Adam Optimize; learning_rate=1e-5 to optimize the efficient training.

3. Training and Validation:

- The dataset was split into training and validation sets using a 60-40 split.

- Data augmentation techniques such as rotation, shifting, and flipping were applied to the training set using ImageDataGenerator to improve model generalization.
- The model was trained for 50 epochs with a batch size of 64.

4. Evaluation:

Used the trained model to predict high resolution images from test dataset.

Evaluated the models performance using metrics such as PSNR, MSE and MAE.

Getting PSNR value of 13.28 db, MSE is 0.046, MAE is 0.178.

Dependencies

- Numpy
- Pandas
- SciKit-Learn
- Matplotlib
- Tensorflow
- Keras
- OpenCV

Evaluation

The performance of the denoising model was evaluated using the following metrics:

- Mean Squared Error (MSE): Measures the average squared difference between denoised and clean images which is about 0.046.
- Peak Signal-to-Noise Ratio (PSNR): Indicates the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation which is about 13.28 db.
- Mean Absolute Error (MAE): Represents the average absolute difference between denoised and clean images which is about 0.178.

Learnings

- In this project I learned and researched about different models such as CNN, Convolutional AutoEncoders, Vanilla AutoEncoders and many more. Also researched about metrics such as PSNR, MAE, MSE.
- Got to know how CNN works in real life application.
- Techniques like data augmentation (e.g., rotation, shifting, flipping) help CNNs generalize better from limited training data.
- Loss Functions: The choice of loss function, such as Mean Squared Error (MSE), plays a critical role in training CNNs for denoising. MSE is commonly used to measure the difference between noisy and denoised images, guiding the network towards learning optimal parameters.
- Optimization and Regularization: Techniques like Adam optimizer and dropout regularization are beneficial in training CNNs for denoising. Adam helps in efficiently updating network weights.

Problem Faced

1. Difficulty in Improving PSNR Value

- Designing a CNN architecture that is both deep enough to capture complex patterns and efficient enough to train effectively was challenging.
- Initial models did not significantly improve the PSNR value, indicating that the architecture might not be optimal for the denoising task.

2. Long Training Time

Hardware Limitations:

- Training deep CNN models on large datasets is computationally intensive.
- The available hardware (e.g., GPU) had limitations in terms of memory and processing power, which extended the training time significantly.

Convolution Operations:

- The large number of convolution operations in the model, especially in deeper layers, contributed to increased computational time.

Conclusion

The image denoising project aimed to enhance the quality of low-resolution, noisy images by leveraging convolutional neural networks (CNNs). Throughout the project, several key insights and learnings emerged, forming the basis for the final conclusions.

References

<https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/trit.2018.1054>

https://ieeexplore.ieee.org/abstract/document/8384706?casa_token=Clev-xMs4mMAAAAA:HQGkk-0JFOCdb7gYUNHB-aW9xQp-n9-0OJZbgQcD9FkR_t7bw330gICTL6OFvErgeAr84bbrAA