Report on Image Denoising (VLG)-IIT-R

Presented by-
Utkarsh Verma
(21112116)

Table of Content:

Objective:

Digital images captured in real-world scenarios often suffer from various types of noise, such as sensor noise, compression artifacts, or environmental interference. This noise can degrade image quality and impact the performance of downstream tasks like object detection, image segmentation, and facial recognition. Denoising autoencoders offer an effective solution by learning to extract and reconstruct clean image features while filtering out unwanted noise.

The project begins with acquiring datasets of noisy and corresponding clean images from specified directories. These images undergo preprocessing steps, including resizing to a standard size of 128x128 pixels, conversion from BGR to RGB color space to ensure consistency, and normalization to scale pixel values to a range between 0 and 1. These steps prepare the data for effective model training and validation.

Dataset Description:

The dataset consists of two main categories of images:

Noisy Images: These images are captured in real-world scenarios where noise, such as sensor noise, compression artifacts, or environmental interference, has corrupted the original image quality.

Clean Images: These images are the corresponding ground truth versions of the noisy images, representing what the image would ideally look like without any noise or distortion.

Resolution: All images in the dataset are resized to a standardized resolution of 128x128 pixels. This ensures uniformity in input size for the denoising autoencoder model.

Colour Space: Images are represented in RGB (Red, Green, Blue) colour space after conversion from the default BGR (Blue, Green, Red) format typically used by OpenCV.

Data Pre-processing :

Data cleaning involves handling missing or incorrect data points that can adversely affect model performance. Common techniques include:

Handling Missing Data: Replace missing values with the mean, median, or mode of the feature. Alternatively, drop rows or columns with missing values if they are insignificant in number.

Handling Outliers: Outliers can skew model training. Techniques like winsorization (capping extreme values) or using robust statistical measures (median and interquartile range) can mitigate their impact.

Normalization and standardization aim to scale numerical features to a standard range, making them more comparable and improving model convergence:
Normalization: Scaling features to a range of [0, 1] using techniques like Min-Max scaling: [X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}).

Standardization: Transforming features to have a mean of 0 and a standard deviation of 1, ensuring that features have comparable scales: [X_{std} = \frac{X - \mu}{\sigma}).

Addressing class imbalance issues in classification tasks to prevent biased model training:
Sampling Techniques: Oversampling minority classes (e.g., SMOTE) or undersampling majority classes to balance class distribution.

Class Weighting: Assigning higher weights to minority class samples during model training to penalize misclassifications more effectively.

Data Splitting:

Training Set:

Purpose: The training set is the largest subset of the dataset used to train the machine learning model. It forms the basis for learning patterns and relationships between input features (independent variables) and the target variable (dependent variable).

Size: Typically, the training set comprises 60% to 80% of the total dataset, depending on the dataset size and complexity of the problem.

Usage: All model parameters (weights and biases) are adjusted during training to minimize the difference between predicted and actual outcomes.

Test Set:

Purpose: The test set serves as an independent dataset to evaluate the final model's performance after training and validation. It provides an estimate of how well the model will generalize to unseen data in real-world scenarios.

Size: Typically, the test set is 10% to 20% of the total dataset, similar in size to the validation set.

Usage: Once the model is trained and fine-tuned using the training and validation sets, its predictions are compared with the actual outcomes in the test set to compute performance metrics (e.g., accuracy, precision, recall, F1-score).

Model Architecture:

The model architecture refers to the specific arrangement and configuration of layers and operations within a machine learning model. It defines how data flows through the model, how features are extracted and transformed, and how predictions are generated. Here,

I'll provide a detailed explanation of a typical model architecture used in image denoising tasks, based on convolutional neural networks (CNNs).

1. Input Layer:

Purpose: Receives input data, typically images, with specified dimensions.

Configuration: Shape: (height, width, channels), where channels represent colour channels (e.g., RGB).

2. Convolutional Layers:

Purpose: Extracts features from input images using convolution operations.

Configuration:

Conv2D Layer: Applies convolution filters to the input. Each filter detects specific features (e.g., edges, textures).Parameters: Number of filters, filter size (kernel size), activation(e.g., ReLU).

MaxPooling2D Layer: Downsamples the feature maps to reduce spatial dimensions. Parameters: Pooling size (e.g., (2, 2)), padding (e.g., same).

3. Encoding Layers:

Purpose: Condenses and abstracts feature representation.

Configuration:

Conv2D Layers: Stacked layers to deepen feature extraction.

MaxPooling2D Layers: Downsampling to reduce spatial dimensions and retain important features.

4. Decoding Layers:

Purpose: Reconstructs the denoised image from encoded features.

Configuration:

UpSampling2D Layers: Increases spatial dimensions to reconstruct the image.

Conv2D Layers: Stacked layers for feature reconstruction.

5.Output Layer:
Purpose: Generates the final output, which is a denoised image.
Configuration:
Conv2D Layer: Applies convolution with sigmoid activation to generate pixel values in the range [0, 1].
Output shape: Same as input shape (height, width, channels).

6.Model Compilation:
Purpose: Specifies optimizer, loss function, and evaluation metrics.
Configuration:
Optimizer: Adam optimizer adjusts model weights based on gradient descent.
Loss Function: Mean Squared Error (MSE) measures the difference between predicted and actual pixel values.
Metrics: Typically, PSNR (Peak Signal-to-Noise Ratio) measures reconstruction quality.

Model Evaluation:

Model evaluation is a critical aspect of machine learning model development that assesses how well a trained model performs on unseen data. In this detailed explanation, I'll cover various metrics, techniques, and considerations involved in evaluating a model, particularly in the context of image denoising using convolutional neural networks (CNNs).

1.Loss Function:

Purpose: Measures the discrepancy between predicted and actual values during training.
Example: Mean Squared Error (MSE) calculates the average squared difference between predicted and actual pixel values.
Lower MSE indicates better model performance in minimizing reconstruction error.

2. Peak Signal-to-Noise Ratio (PSNR):

Purpose: Evaluates image quality by measuring the ratio of signal strength to noise interference.
Calculation: PSNR=20·log⁡10(max pixel valueMSE)\text{PSNR} = 20 \cdot \log_{10} \left(\frac{\text{max pixel value}}{\sqrt{\text{MSE}}} \right)PSNR=20·log10(MSEmax pixel value)
Higher PSNR values (in dB) indicate higher image quality and better denoising performance.

3. Structural Similarity Index (SSIM):

Purpose: Measures similarity between two images, considering luminance, contrast, and structure.
Range: SSIM values range from -1 to 1, where 1 indicates perfect similarity.
Usage: Complements PSNR by evaluating perceptual image quality.

4. Visual Inspection:

Purpose: Subjective evaluation by visually comparing denoised images with ground truth (clean images).
Criteria: Assess image clarity, sharpness, and preservation of details.

Results:

Explaining the results of a machine learning model, particularly in the context of image denoising using convolutional neural networks (CNNs), involves interpreting various metrics and assessing the overall performance. Here's a detailed explanation of how results are analyzed and interpreted:

Metrics Used for Evaluation:

1.Mean Squared Error (MSE):

Definition: MSE measures the average squared difference between predicted and actual pixel values.
Interpretation: Lower MSE values indicate better agreement between denoised (predicted) and clean (actual) images.

2.Peak Signal-to-Noise Ratio (PSNR):

Definition: PSNR quantifies image quality by evaluating the ratio of the maximum possible pixel value to the root mean squared error.
Interpretation: Higher PSNR values (expressed in decibels, dB) signify better preservation of image details and less noise interference.

3.Structural Similarity Index (SSIM):

Definition: SSIM assesses image similarity based on luminance, contrast, and structure.
Interpretation: SSIM values closer to 1 indicate greater similarity between denoised and clean images.

Steps in Result Interpretation

1. Quantitative Analysis

Evaluate MSE:
Calculate MSE for each test image to quantify the average pixel-wise error.
Interpret results: Lower MSE values indicate better denoising performance, reflecting reduced pixel-level discrepancies.
Assess PSNR:
ompute PSNR for each denoised image compared to its corresponding clean image.
Interpret results: Higher PSNR values indicate higher image quality and better denoising effectiveness.

Analyze SSIM:
Compute SSIM scores to evaluate perceptual image quality and similarity.
Interpret results: SSIM values closer to 1 indicate better preservation of image structure and details.

2. Visual Inspection
Compare Images:
Visually compare denoised images with clean images (ground truth).
Assess clarity, sharpness, and preservation of fine details.
Look for artifacts or distortions introduced during denoising.
Subjective Assessment:
Seek feedback from domain experts or stakeholders on the visual quality of denoised images.
Consider human perception of image fidelity and naturalness.

Interpretation Considerations:

1.Dataset Representation:
Ensure the test dataset covers a wide range of noise levels and realistic scenarios.
Verify that the model generalizes well to unseen data.

2.Model Performance:
Analyze trends in MSE, PSNR, and SSIM across epochs during training and testing.
Identify underperforming images or outliers that may require further investigation.

3.Hyperparameter Impact:
Assess how changes in learning rate, batch size, or network architecture affect model performance.
Optimize hyperparameters based on evaluation results to improve model robustness.

Conclusion:

Effective interpretation of results in image denoising involves a balanced analysis of quantitative metrics (MSE, PSNR, SSIM) and qualitative assessment (visual inspection). It ensures that the CNN model effectively reduces noise while preserving essential image details. By leveraging both objective metrics and subjective evaluation, machine learning practitioners can validate model efficacy, refine algorithms, and deliver high-quality denoising solutions tailored to specific application requirements

Thank You