Project Report for Course EE5179 Deep Learning for Imaging

By: Sumeet Shekar (EE24D029) , V Divya Madhuri (EE24D004), Gunjan Parihar (EE24D035), and Athira K S (EE23D034)

Architecture details. Cite the paper (or the state-of-the-art model) you have used in your experiment. Mention whether you have made any modifications to the model for this task.

1. Architecture

The core architecture is a combination of a **pre-trained encoder** with an attention-based **U-Net decoder** for image denoising is shown in Fig.1. Below are the components and their functions:

1.1 Overview of U-Net with Attention

The architecture is based on an **Attention U-Ne**t structure with a pre-trained encoder. It combines the effective segmentation power of the traditional U-Net with advanced attention mechanisms to focus on critical regions.

1.2 Key Components in Detail

Encoder (Feature Extraction)

- The encoder is responsible for **extracting features** from the input image at multiple levels of abstraction.

- Popular encoders used:
    - **ResNet** (e.g., `resnet34`):
        - Uses convolutional blocks with residual connections.
        - **Residual Connections** prevent vanishing gradients, allowing deeper networks to learn efficiently.
    - **EfficientNet** (e.g., `efficientnet-b0`, `efficient net-b3`):
        - Uses **compound scaling** to balance depth, width, and resolution for efficient performance.
        - Achieves high accuracy with fewer parameters compared to traditional CNNs like ResNet.

- **Transfer Learning**:
    - The encoder is pre-trained on ImageNet, providing a strong baseline understanding of natural images.
    - Improves generalization on smaller datasets and speeds up convergence.

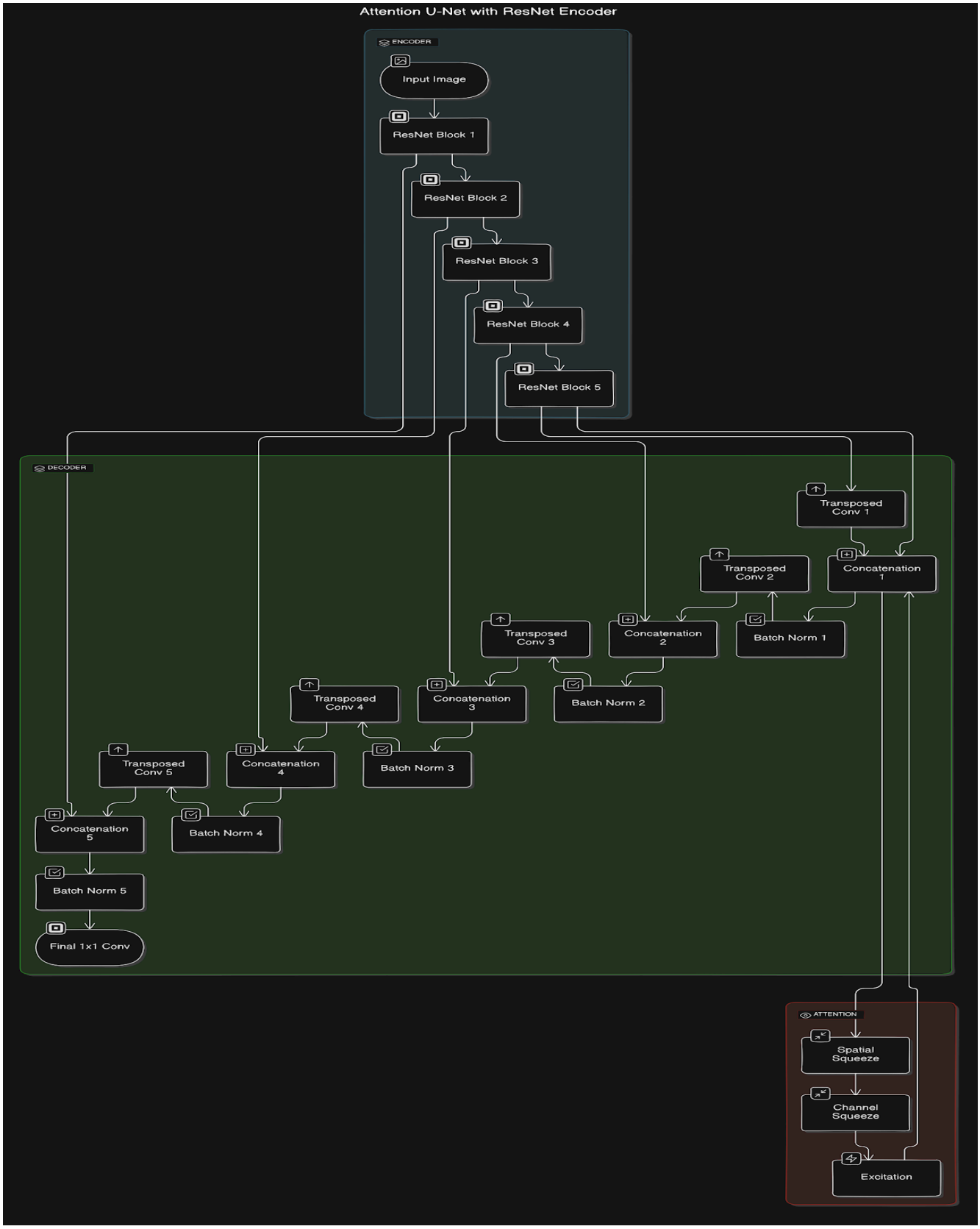


Fig.1. Architecture of our proposed model. (Top) Encoder, (Middle) Decoder, (Bottom) Attention

Skip Connections

- **Skip connections** bridge each corresponding level in the encoder and decoder, preserving spatial information from early layers during reconstruction.
- Crucial for preserving fine details and structural information.

Attention Mechanisms

- **SCSE (Spatial and Channel Squeeze & Excitation) Attention Block**:
    - **Channel Attention**: Focuses on relevant feature maps (channels).
    - **Spatial Attention**: Highlights specific regions within feature maps, ensuring important details are preserved.
    - Enhances the network's sensitivity to important features, helping retain small defects during denoising.

Decoder (Image Reconstruction)

- The decoder reconstructs the denoised image from the multi-scale features extracted by the encoder.
- Uses:
    - **Transposed Convolutions** for upsampling.
    - **Concatenation** with skip connections for detail retention.
    - **Batch Normalization** for stable training.

Final Convolutional Layer

- A final **1x1 Convolution** layer maps the output to a standard RGB image (3 channels).
- Produces the final denoised image, closely matching the dimensions of the ground truth.

2. Handling Variable Image Sizes

2.1 Challenges with Variable Image Sizes

- Real-world datasets often have images of varying resolutions and aspect ratios.
- Resizing to a fixed size can reduce or distort critical features.

Image shape in 'Train' category:
Shape (H x W): 1000 x 1000 - 255 images
Shape (H x W): 800 x 800 - 309 images
Shape (H x W): 1024 x 1024 - 1746 images
Shape (H x W): 840 x 840 - 192 images
Shape (H x W): 700 x 700 - 159 images
Shape (H x W): 900 x 900 - 135 images

Image shape in 'Val' category:
Shape (H x W): 1000 x 1000 - 15 images
Shape (H x W): 1024 x 1024 - 165 images
Shape (H x W): 800 x 800 - 33 images
Shape (H x W): 900 x 900 - 18 images
Shape (H x W): 700 x 700 - 18 images
Shape (H x W): 840 x 840 - 15 images

2.2 Adaptive Padding Approach

- Instead of resizing, **adaptive padding** is used:
    - **Padding** ensures images match the largest dimensions in a batch.
    - Maintains original aspect ratios and spatial details without distortion.
    - Zero-padding (`0` values) is used, avoiding artificial noise.

2.3 Custom Collate Function

- A custom `collate_fn` handles dynamic padding:
    - Determines the maximum width and height in each batch.
    - Pads each image to match these dimensions, preserving batch processing efficiency.

3. Dataset Management

3.1 Dataset Components

- **Degraded Images**: Noisy input images that need denoising.
- **Defect Masks**: Binary masks highlighting regions with critical defects.
- **Ground Truth**: Clean reference images used for supervised training.

3.2 Data Splitting

- Dataset is split into:
    - **Training Set (70%)**: Used to train the model.
    - **Validation Set (20%)**: Monitors performance during training.
    - **Test Set (10%)**: Evaluates final model performance.

3.3 Data Augmentation (Optional)

- Techniques like **random rotations, flipping, and brightness adjustments** can be added to enhance robustness to diverse noise patterns.
- Useful for increasing dataset size and variability.

4. Training Setup

4.1 Loss Function: Weighted Loss

- **Weighted Loss Function** to preserve important regions:
    - Base loss is **Mean Squared Error (MSE)**.
    - Gives higher weight to defect regions:
        - `Loss = MSE(non-masked areas) + weight * MSE(masked areas)`
    - Prevents over smoothing critical details.

4.2 Optimizer: AdamW

- **AdamW** optimizer:
    - Variant of Adam with decoupled weight decay for better regularization.
    - Handles noisy gradients well and stabilizes convergence.
    - Weight decay set to `1e-5`.

4.3 Learning Rate Adjustment

- Initial learning rate set to `0.001`.
- **Scheduler (`ReduceLROnPlateau`)** adjusts the learning rate:
    - Monitors validation loss and reduces the rate when performance plateaus.
    - Helps avoid overfitting and ensures stable convergence.

5. Evaluation Metrics

5.1 Peak Signal-to-Noise Ratio (PSNR)

- **PSNR** is the primary metric for evaluating denoising quality:
    - A higher PSNR indicates better quality and closer reconstruction to the ground truth.
    - Calculated after every epoch for progress monitoring.
    - Final evaluation on the test set for generalization.

5.2 Visual Inspection

- Visual inspection alongside quantitative metrics:
    - Compare **Degraded Image, Mask, Ground Truth, and Model Output**.
    - Assess how well the model retains small defects while reducing noise.

Summary

The architecture is a combination of a **pre-trained encoder** and an attention-enhanced U-Net decoder, tailored for **image denoising tasks with critical detail preservation**. The adaptive handling of variable image sizes ensures feature integrity, making the setup effective for complex denoising scenarios. Transfer learning and advanced attention mechanisms allow for good generalization, even with limited datasets, while providing visual and quantitative improvements over noisy inputs.

Advantages of the Approach

- **Flexibility**: Choice between ResNet and EfficientNet for encoders.
- **Efficiency**: Leverages pre-trained networks for faster convergence.
- **Detail Preservation**: Uses attention mechanisms and weighted loss to retain critical tiny features.
- **Modularity**: Easily extendable for other restoration tasks like super-resolution or inpainting.

Potential Improvements

- **Data Augmentation** to handle diverse noise patterns.
- Exploring **SSIM Loss** (Structural Similarity Index) for better perceptual quality.
- Using a **more advanced attention mechanism** (like Transformer-based attention) for finer control.

Test Results for Denoising with Defect Preservation

Sample outputs and PSNR/SSIM values (object-wise and overall) on the validation set: You can plot a bar chart showing PSNR/SSIM values for each object. Arrange the objects on the X-axis in alphabetical order of their names (bottle first, cable second, …). Also, report the average PSNR/SSIM value for the entire validation set.

Sample outputs:

Fig.2. Sample outputs

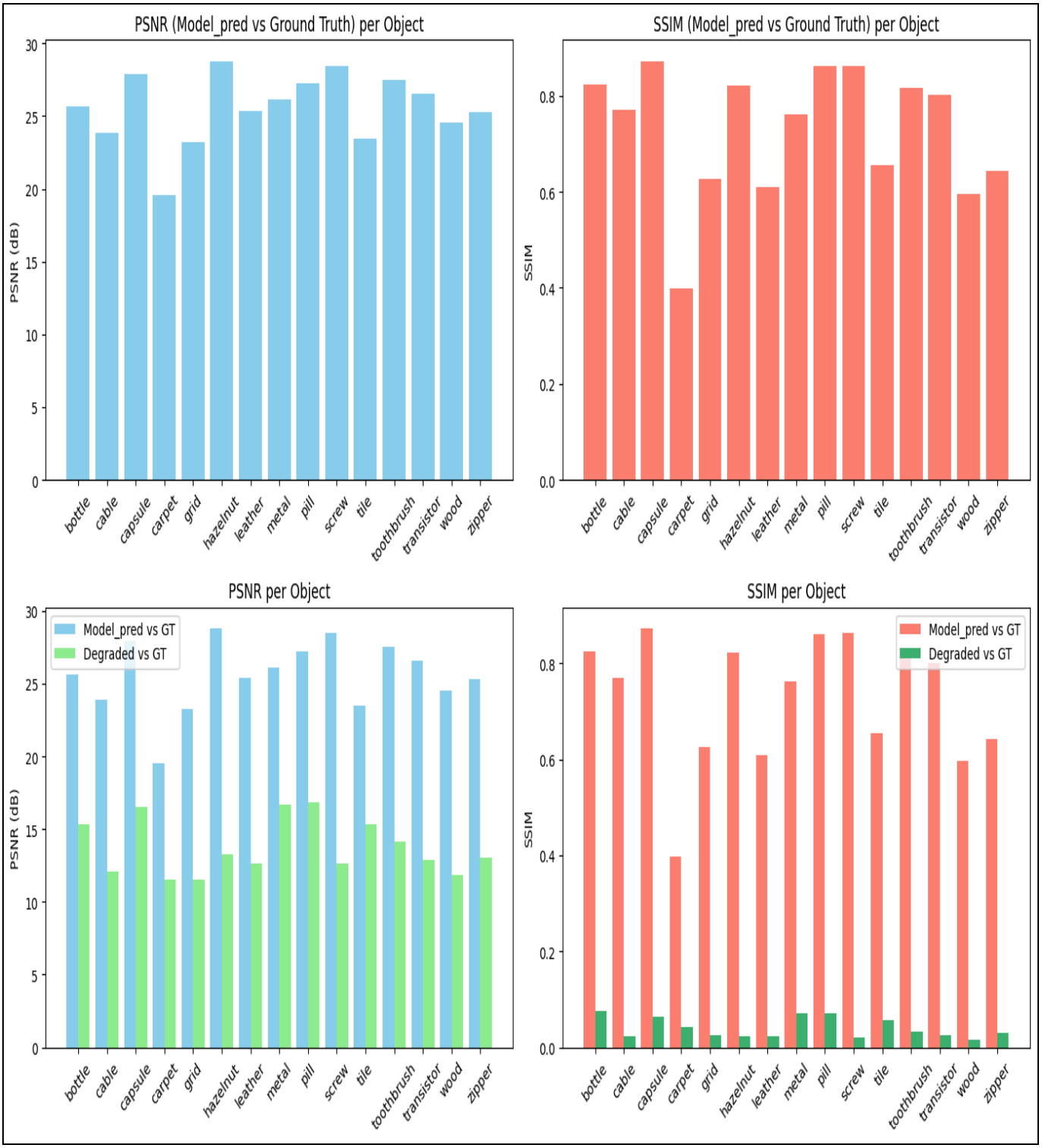


Fig.3. PSNR and SSIM plot for the entire validation set.

PSNR and SSIM Metrics

Per-Object Results

Object	Avg PSNR (Model vs GT)	Avg SSIM (Model vs GT)	Avg PSNR (Degraded vs GT)	Avg SSIM (Degraded vs GT)
Bottle	25.64	0.8233	15.37	0.0751
Cable	23.88	0.7696	12.13	0.0238
Capsule	27.92	0.8723	16.54	0.0644
Carpet	19.53	0.3980	11.57	0.0432
Grid	23.23	0.6259	11.52	0.0253
Hazelnut	28.79	0.8216	13.29	0.0239
Leather	25.36	0.6092	12.66	0.0221
Metal	26.12	0.7616	16.71	0.0718
Pill	27.22	0.8611	16.83	0.0706
Screw	28.46	0.8620	12.65	0.0211
Tile	23.48	0.6551	15.36	0.0562
Toothbrush	27.50	0.8161	14.11	0.0317
Transistor	26.56	0.8009	12.90	0.0246
Wood	24.54	0.5955	11.88	0.0164
Zipper	25.31	0.6425	13.01	0.0311

Table 1. Average PSNR and SSIM values for Model versus Ground Truth, and Degraded versus Ground Truth.

Overall Averages

- **Overall Average PSNR (Model_pred vs GT):** 25.57
- **Overall Average SSIM (Model vs GT):** 0.7277
- **Overall Average PSNR (Degraded vs GT):** 13.77
- **Overall Average SSIM (Degraded vs GT):** 0.0401

Results for capsule_faulty_imprint_001.png PSNR

(Whole): 28.28 dB, SSIM (Whole): 0.8767

PSNR (Masked): 52.18 dB, SSIM (Masked): 0.9993

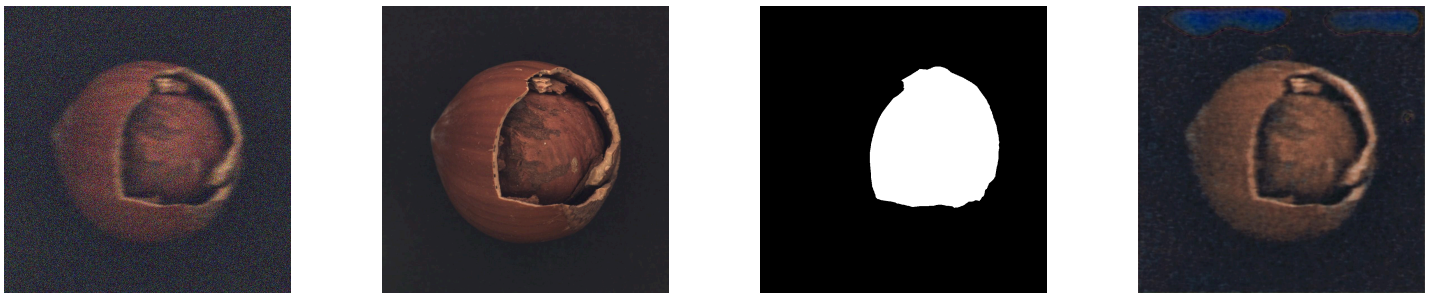
Fig.4 (a) Degraded Image, (b) Ground truth, (c) Masked Image, (d) Output for Capsule faulty imprint

Results for hazelnut_crack_002.png

PSNR (Whole): 28.02 dB, SSIM (Whole): 0.8254 PSNR

(Masked): 33.27 dB, SSIM (Masked): 0.9509

Fig. 5 (a) Degraded Image, (b) Ground truth, (c) Masked Image, (d) Output for hazelnut crack.

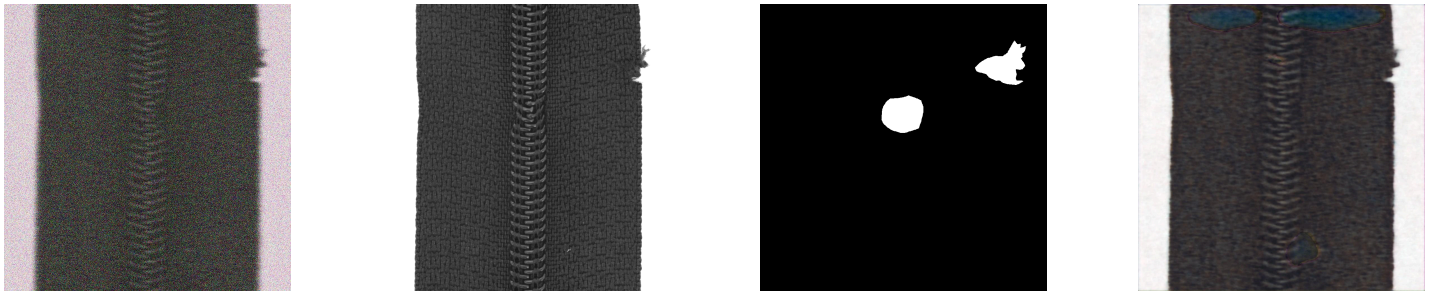


Results for zipper_combined_001.png

PSNR (Whole): 24.56 dB, SSIM (Whole): 0.6276 PSNR

(Masked): 37.90 dB, SSIM (Masked): 0.9887

Fig.6 (a) Degraded Image, (b) Ground truth, (c) Masked Image, (d) Output for Zipper combined

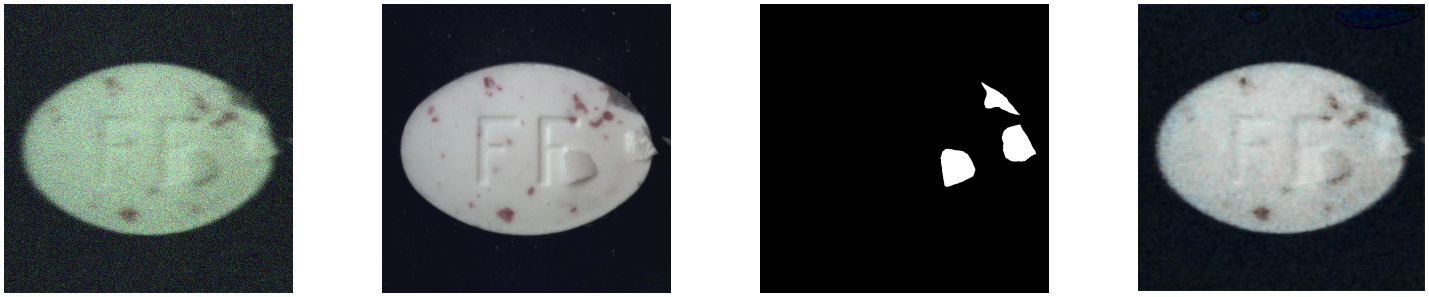


Results for pill_combined_001.png

PSNR (Whole): 27.53 dB, SSIM (Whole): 0.8700 PSNR

(Masked): 40.31 dB, SSIM (Masked): 0.9958

Fig.7 (a) Degraded Image, (b) Ground truth, (c) Masked Image, (d) Output for Pill combined

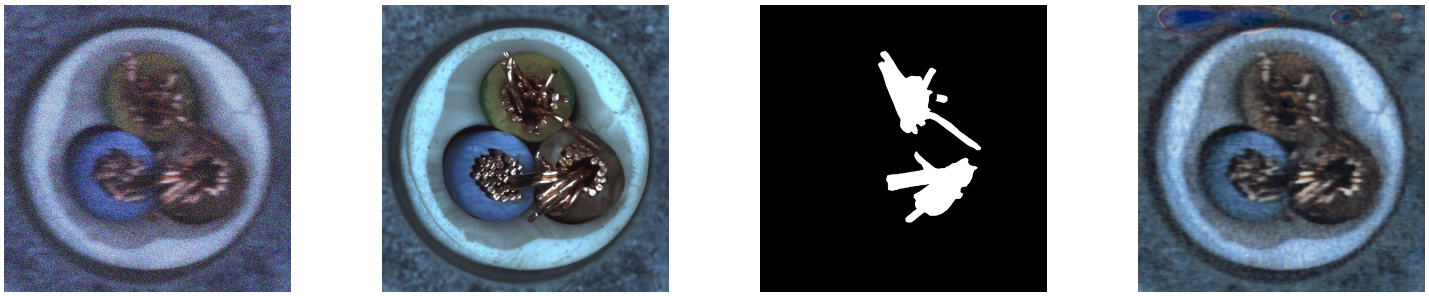


Results for cable_bent_wire_001.png

PSNR (Whole): 23.11 dB, SSIM (Whole): 0.7449 PSNR

(Masked): 29.72 dB, SSIM (Masked): 0.9710

Fig.8 (a) Degraded Image, (b) Ground truth, (c) Masked Image, (d) Output for cable bent wire



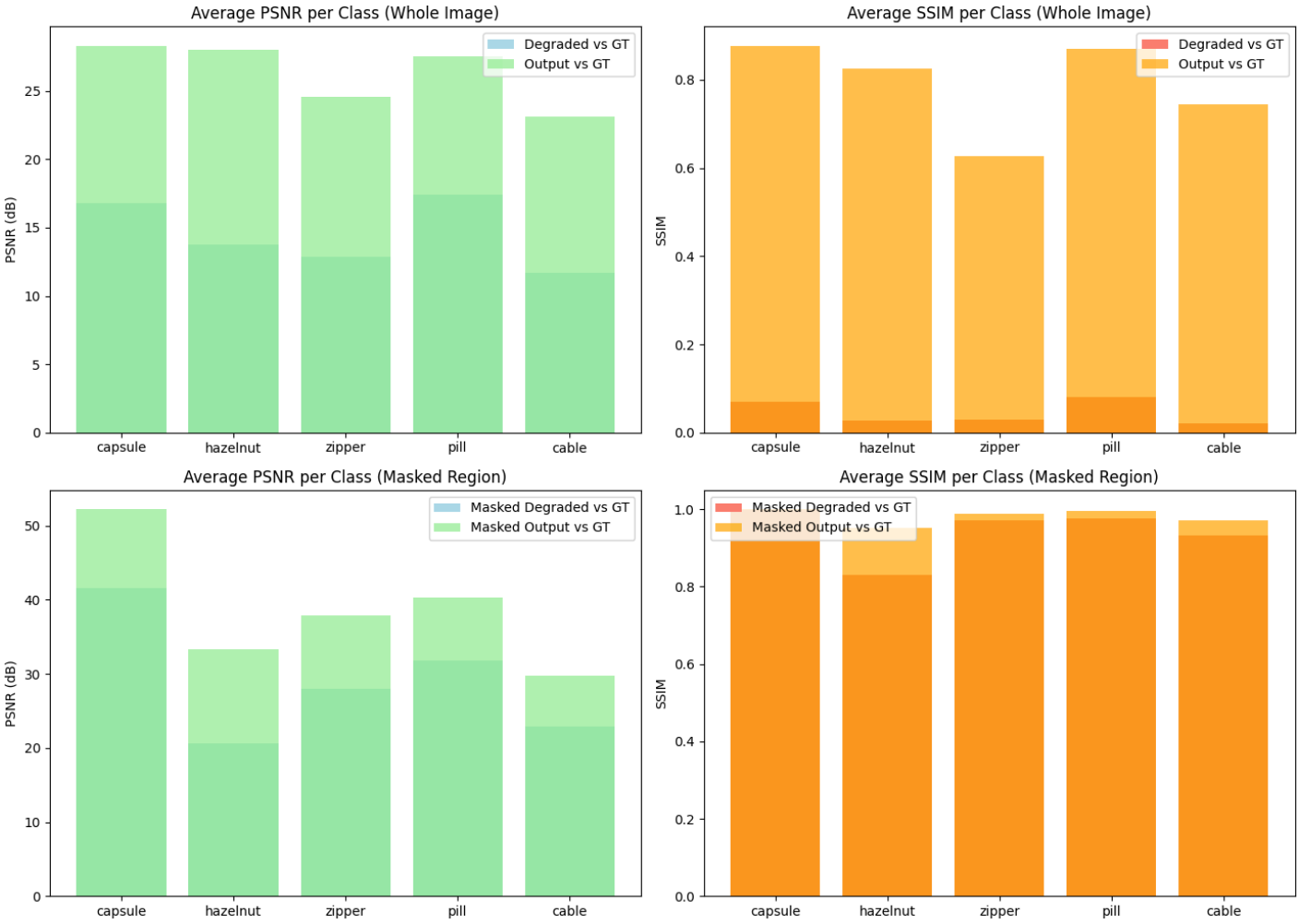


Fig. 9. PSNR and SSIM plot for five test samples.

The link to a Github project page (make it public) containing code files, model weights (or link to model weights), the list of required packages with versions to run your model, and test instructions. For each group, make one Github project page and add the members as collaborators.

Github Project Page

Model

List of required packages with versions:

torch>=1.8.0
torchvision>=0.9.0
torchmetrics>=0.6.0
Pillow>=8.0.0
numpy>=1.19.0
matplotlib>=3.3.0
segmentation-models-pytorch>=0.2.0
gdown>=3.12.2
fpdf>=1.7.2

Test instructions:

Usage

1. Rearrange Dataset Files: Use the rearrange_and_rename_files function to structure the dataset as required. This will organize and rename files within the dataset.

   ```
   src_dir = 'Dataset/Denoising_Dataset_train_val'
   dest_dir = 'Dataset/structured_data'
   rearrange_and_rename_files(src_dir, dest_dir)
   ```

2. Download Model Checkpoint: The code checks if the model checkpoint exists locally. If not, it will download it from Google Drive.

3. Run Inference and Evaluation: Use the infer_and_evaluate function to process the images, calculate metrics, and create a report. Example usage to process images and create a report:

   ```
   file_id = '1gGiza9UsHM679TDlvn-1fhhu6V2Y0hS2'
   url = f'https://drive.google.com/uc?id={file_id}'
   checkpoint_path = 'Model/checkpoint_epoch_18.pth'

   if not os.path.exists(checkpoint_path):
       os.makedirs(os.path.dirname(checkpoint_path), exist_ok=True)
       gdown.download(url, checkpoint_path, quiet=False)

   model = load_model(checkpoint_path, encoder_name='resnet34')
   input_folder = 'Dataset/structured_data/Val/degraded'
   ground_truth_folder = 'Dataset/structured_data/Val/ground_truth'
   mask_folder = 'Dataset/structured_data/Val/defect_mask'
   save_output_folder = 'output_results'

   # Run on the first n images, e.g., n=5
   infer_and_evaluate(model, input_folder, ground_truth_folder, mask_folder,
   save_output_folder, n_images=5)
   ```

4. View Results: Check the output_results/ directory for the generated PDF report and metric plots.

References

Key Papers

1. **U-Net: Convolutional Networks for Biomedical Image Segmentation**
- **Authors**: Olaf Ronneberger, Philipp Fischer, Thomas Brox
- **Link**: arXiv:1505.04597
- **Summary**: This paper introduces the U-Net architecture, which has become a foundational model for various segmentation and image restoration tasks. It emphasizes the importance of skip connections for preserving spatial information, a core concept used in the current denoising architecture.
- **Relevance**: The denoising model described uses a U-Net-based architecture with attention mechanisms for better feature retention.

2. **Attention U-Net: Learning Where to Look for the Pancreas**
- **Authors**: Olaf Oktay, Jo Schlemper, et al.
- **Link**: arXiv:1804.03999
- **Summary**: This paper introduces an attention mechanism in the U-Net, allowing the network to focus on relevant regions in the image. It combines spatial and channel attention, enhancing the model's ability to handle fine details.
- **Relevance**: The current model employs a similar approach by integrating attention mechanisms to emphasize important image regions.

3. **EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks**
- **Authors**: Mingxing Tan, Quoc V. Le
- **Link**: arXiv:1905.11946
- **Summary**: EfficientNet introduces a method to scale neural networks using a compound scaling method. It achieves state-of-the-art performance while maintaining efficiency in terms of computational resources.
- **Relevance**: EfficientNet serves as one of the encoder backbones, providing a highly efficient feature extractor for the denoising task.

4. **Squeeze-and-Excitation Networks**
- **Authors:** Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu
- **Link**: arXiv:1709.01507
- **Summary**: This paper introduces the concept of Squeeze-and-Excitation (SE) blocks, which recalibrate channel-wise feature responses through channel attention. This concept is extended in the SCSE (Spatial and Channel Squeeze & Excitation) blocks used in the current architecture.
- **Relevance**: The attention mechanism used in the denoising model is based on similar SE blocks, specifically focusing on both spatial and channel attention.

5. **Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections**
- **Authors**: Ying Tai, Jian Yang, Xiaoming Liu
- **Link**: arXiv:1707.05479
- **Summary**: This paper discusses deep convolutional encoder-decoder networks with symmetric skip connections for image restoration tasks. It emphasizes the importance of deep feature extraction while retaining spatial information.
- **Relevance**: The skip connections in the U-Net used for the denoising task are based on similar principles, aiming to maintain fine details throughout the network.

6. **Deep Residual Learning for Image Recognition**
- **Authors**: Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun
- **Link**: arXiv:1512.03385
- **Summary**: This paper introduces ResNet, a backbone used in many encoder-decoder architectures. It employs residual learning to ease the training of deep networks, which is critical in extracting robust features for tasks like denoising.
- **Relevance**: The residual learning concept is foundational to the ResNet encoder option in the denoising architecture.

Additional Resources

- **Segmentation Models Library**: A popular open-source library that provides access to pre-trained encoder-decoder models with attention mechanisms.
- **GitHub**: Segmentation Models
- **Relevance**: This library supports the U-Net variants with attention mechanisms and provides access to EfficientNet and ResNet backbones, as used in the denoising model.