Quality-Aware JPEG Restoration Using a Cascaded Deep Network with Hybrid Spatial–Frequency Loss

Akshadh Atherya
23MIA1065, CSE VIT
Chennai, India
akshadhathreya.s2023@vitstudent.ac.in

Arun Kumaran
23MIA1171, CSE
VIT Chennai, India
arunkumaran.2023@vitstudent.ac.in

Gvn Kishore
23MIA1157, CSE
VIT Chennai, India
kishore.gvn2023@vitstudent.ac.in

Dr Geetha S
VIT Chennai, India

Abstract*—JPEG is one of the most widely used image formats, but its lossy compression process often introduces visible distortions such as blocking, blurring, and mosquito noise. These artifacts reduce both the visual quality of images and their usefulness in practical applications. Our project focuses on improving com- pressed JPEG images by implementing an efficient enhancement pipeline inspired by recent deep-learning based restoration tech- niques. Using concepts from the reference paper Efficient and Effective Blind JPEG Image Improvement with Sequential Feature Processing, we develop a simplified yet effective model that estimates the quality of an input image and progressively restores it through multiple lightweight processing stages. Supporting code from pub- licly available implementations (SelfExSR, model utilities, and processing scripts) is adapted to build and evaluate our approach. Experiments are carried out on standard datasets with various compression levels to understand performance across multiple qualities. The results show a clear improvement in visual clarity, edge preservation, and artifact reduction compared to the input JPEG images. The work demonstrates that even compact neural modules, when used in a sequential manner, can achieve strong restoration performance while keeping computation relatively low.

***Index Terms*—Deep learning, digital image restoration, image enhancement, JPEG artifact removal, quality estimation, sequential processing**

I. Introduction & Motivation

Digital images are exchanged and stored more frequently than ever, and formats like JPEG continue to dominate because of their small file sizes and wide compatibility. However, JPEG achieves this efficiency by discarding part of the original image information, which introduces noticeable artifacts such as block distortions, ringing effects, and loss of fine structural details. These degradations become even more severe when images are compressed multiple times, a common situation in social media uploads, messaging applications, and online platforms. As a result, the visual clarity of the image is reduced, and the usefulness of the image in tasks such as classification, detection, and enhancement is affected.

In recent years, deep learningbased restoration techniques have shown remarkable progress in tasks such as denoising, super-resolution, and artifact removal. While many advanced models provide strong results, they often require heavy computation and large memory, making them unsuitable for real-time or resource-limited environments. Additionally, many traditional JPEG enhancement methods are designed for a fixed or narrow range of quality factors, which limits practical usability because real-world images vary widely in compression level.

To address these issues, our project examines a restoration approach inspired by the reference research paper "Efficient and Effective Blind JPEG Image Improvement With Sequential Feature Processing." The method focuses on breaking down the enhancement task into multiple lightweight processing stages, each contributing to gradual quality improvement while keeping the computational cost low. A key motivation for adopting this technique is its ability to adapt to different JPEG qualities by estimating the degradation level before processing, allowing the system to operate effectively even when the true quality factor is unknown. By studying and implementing simplified versions of these techniques, our goal is to demonstrate that strong restoration results can be achieved without excessively complex architectures. This work bridges theoretical concepts from recent research with practical implementation using available codebases, helping us understand how modern image enhancement systems are designed, trained, and evaluated.

II. Related Work (Base Paper Summary)

Research on JPEG artifact removal has evolved from traditional filtering approaches to highly advanced deep learning models. Early methods relied on smoothing filters, deblocking operations, or sparse representations to manually suppress noise and block artifacts. While these techniques offered moderate improvements, they struggled to generalize to different image types and varying compression levels.

The introduction of deep convolutional neural networks significantly improved restoration quality. Models such as AR-CNN, DnCNN, and other CNN-based architectures demonstrated that learned features are far more effective at cap- turing structured distortions caused by JPEG compression. Transformer-based models and adversarial learning methods further enhanced perceptual quality by preserving textures and generating more natural-looking results. However, many of these approaches are computationally expensive and optimized only for fixed ranges of JPEG quality factors, limiting their practicality in real-world scenarios.

The base paper for our project, “Efficient and Effective Blind JPEG Image Improvement with Sequential Feature Processing,” proposes a more balanced solution. Instead of relying on one large network, the authors design a cascade of smaller Image Processing Parts (IP Parts). Each stage performs a specific part of the enhancement, allowing the overall system to refine the image progressively. A key innovation is the Quality Estimation Part (QE Part), which predicts the degradation level of the input image. This predicted “quality representation” helps the model adjust its processing for different compression strengthseven when the true JPEG quality factor is unknown. Another important contribution from the paper is the use of lightweight Processing Blocks. Basic Processing Blocks handle local features efficiently, while Global Processing Blocks capture large-scale structures without requiring heavy computational resources. This combination enables the model to achieve high performance even on high-resolution images while keeping memory usage and inference time relatively low. The paper also demonstrates strong results on double-compressed images, a realistic challenge where the artifact patterns become harder to remove. This robustness highlights the advantage of progressive, quality-aware feature processing over traditional single-pass methods. Overall, the base paper provides a practical blueprint for creating a restoration system that delivers high-quality enhancement, supports a wide range of JPEG qualities, and remains computationally efficient. Our project draws inspiration from these ideas to build a simplified, student-friendly version of the methodology for experimentation and evaluation.

III. Proposed Methodology

Our project focuses on building a simplified yet effective JPEG artifact removal system inspired by the sequential feature processing approach described in the reference paper. The core idea is to avoid relying on one large, complex neural network and instead break the restoration task into multiple smaller steps. Each step handles a specific part of the enhancement, allowing the image to be gradually refined in a controlled and computationally efficient manner.

The proposed pipeline contains two major components:

- A *Quality Estimation Module* that predicts how severely the image has been compressed, and
- a *Multi-Stage Restoration Module* that enhances the image progressively based on the estimated quality.

By estimating the degradation level first, the model is able to adapt its processing to both lightly and heavily compressed images. This is important because real-world JPEG files do not always store their compression quality factor, especially when they have been resized or compressed more than once.

Each restoration stage uses lightweight building blocks that extract features, reduce artifacts, and enhance details. These stages operate at multiple resolutions so that both local textures and global structures can be restored effectively. Instead of making the architecture unnecessarily deep, we rely on repeated but efficient processing units, which makes the pipeline more suitable for limited hardware environments such as laptops or single-GPU setups.

To support our implementation, we refer to open-source repositories such as *SelfExSR* and the provided Colab environment for training, evaluation, and visualization. While the full architecture from the original paper is complex, we adopt only the components necessary for demonstrating the sequential improvement concept. The model is trained on JPEG-compressed images of different quality levels to ensure robustness across diverse inputs.

Overall, the methodology aims to balance three aspects:

- *Restoration quality* producing clean, sharper, and more natural-looking images
- *Efficiency* keeping the model lightweight and allowing fast inference
- *Generalization* performing well on various JPEG qualities, including double-compressed cases

In the following subsections, we explain the workflow in detail, covering the system overview, architecture design, and the algorithms used within our implementation.

A. Overview

The core idea of our approach is to enhance JPEG-compressed images through a gradual, stage-wise restoration pipeline rather than a single heavy model. JPEG artifactssuch as blocking, ringing, and loss of fine texturesoccur at dif- ferent scales, so addressing them in one step often leads to over-smoothing or incomplete correction. To overcome this, our system processes the image through multiple lightweight enhancement stages, where each stage focuses on a specific restoration task.

Before enhancement begins, the image passes through a *Quality Estimation Module*. This module predicts a “quality representation value,” which acts as a guide for the rest of the system. Instead of relying on the actual JPEG Quality Factor (QF)which is often missing or unreliable in real-world imagesthe model learns to infer how degraded the image is directly from its visual characteristics. This estimated quality helps the restoration stages decide how aggressively they need to process the image.

Once the quality is estimated, the image enters a series of *Restoration Stages*, each designed to progressively refine details, reduce compression artifacts, and restore natural textures. These stages work in a cascade, meaning the output of one stage becomes the input to the next. Early stages perform coarse corrections like deblocking, while later stages focus on finer textures, edges, and color consistency. This multi-stage approach is inspired by the reference papers sequential feature processing idea but adapted into a simplified, student-friendly form. By using smaller modules repeated over several steps, we reduce computational complexity while still achieving visible improvements in restored images.

The structured workflow which is in the conclusions ensures that improvements are made gradually and consistently, giving the final image a more natural and clean appearance without drastic overprocessing.

B. Architecture Details

The architecture used in our project is inspired by the multi-stage restoration design proposed in the reference paper, but it is simplified to make it easier to implement, train, and understand within the scope of a student project. The system is divided into two main components: a *Quality Estimation Module* and a *Multi-Stage Restoration Network*, both of which work together to enhance JPEG-compressed images.

A. **Quality Estimation Module.** The first component predicts how strongly the image has been compressed. Instead of reading the JPEG Quality Factor (which is often missing or unreliable), this module analyzes the input image using a small convolutional block. From the learned patternssuch as blockiness, noise level, or smoothnessit outputs a single scalar representing the "compression severity." This quality value is important because it helps the later stages adjust their behavior.
   - Higher predicted degradation → stronger corrections applied
   - Lower predicted degradation → lighter, detail-preserving corrections

   This makes the pipeline adaptable to different JPEG qualities and useful even for double-compressed images.

B. **Multi-Stage Restoration Network.** The main enhancement is performed by a series of lightweight processing stages arranged in a cascade. Each stage follows a similar structure but focuses on progressively finer levels of restoration. Each stage contains:
   - A shallow encoderdecoder block
   - Downsampling helps capture larger structures such as block artifacts.
   - Upsampling restores the original resolution after processing.
   - Basic Processing Blocks (BPBs) These are small convolutional units designed to reduce local artifacts, sharpen edges, and correct color distortions.They are computationally light, making the model efficient.
   - Skip Connections These connections help preserve the original image structure and prevent over-smoothing.

   At the end of each stage, the network outputs a partially restored image. This intermediate result becomes the input for the next stage. Early stages perform heavy lifting by removing blocks and ringing, while later stages fine-tune details such as textures and shading.

C. **Multi-Resolution Feature Processing.** Inspired by U-Net-style designs, each stage performs feature extraction at multiple scales.
   - *High-resolution layers* capture edges, tiny details, and textures.
   - *Low-resolution layers* capture broader patterns like block boundaries or color inconsistencies.

   This combination allows the network to handle both global and local artifacts effectively.

D. **Integration of Quality Information.** The predicted quality value from the Quality Estimation Module is injected into the processing blocks using channel-wise weighting. This means the restoration stages learn to amplify or suppress certain feature channels depending on the images degradation level.This makes the system flexible and more reliable than methods that assume a fixed JPEG quality factor.

E. **Output.** After passing through all stages, the final stage produces the restored image. Because the enhancement is carried out progressively, the final output tends to look more natural, with improved clarity and significantly reduced compression artifacts.

This architecture achieves a balance between:
- *Performance* (clear artifact removal),
- *Efficiency* (lightweight and fast),
- *Generalization* (works across different JPEG qualities).

C. Algorithms Used

The proposed system relies on a combination of image processing and deep learning algorithms, each chosen to balance restoration quality and computational efficiency. The aim is to remove JPEG artifacts progressively while preserving natural textures and fine structural details. Below are the main algorithms and techniques used in our implementation.

A. **Convolutional Feature Extraction.** At the core of every stage in the network are convolutional layers, which learn to detect local features such as edges, textures, and artifact patterns. These layers allow the model to identify:
   - Block boundaries introduced by JPEG
   - Ringing effects around edges
   - Loss of sharpness in high-frequency regions

   The convolution kernels adapt during training, enabling the model to learn patterns directly from compressed images rather than relying on manually designed filters.

B. **Lightweight Processing Blocks (BPBs).** The Basic Processing Blocks are simplified versions of advanced modules used in modern restoration networks. Each block typically includes:
   - Convolution layers
   - Activation functions
   - Feature scaling
   - Residual connections

These blocks help stabilize training while keeping the architecture lightweight. Their primary role is to reduce noise, smooth unnatural textures, and prepare features for refinement in later stages.

C. **EncoderDecoder Mechanism.** To handle both global and local artifacts, each restoration stage uses a small encoderdecoder structure:
   - *Encoder:* Downsamples features to capture large-scale distortions like blockiness or color banding.
   - *Decoder:* Upsamples and reconstructs fine details at the original resolution.

   This multi-resolution processing is essential because JPEG artifacts occur at different scales.

D. **Skip Connections.** Inspired by U-Net and ResNet architectures, skip connections allow early features to bypass deeper layers, preserving structural information.This helps prevent:
   - Over-smoothing
   - Loss of edges
   - Unnatural textures

   Skip connections also make training more stable by improving gradient flow.

E. **Quality Estimation Algorithm.** A small CNN is used to predict a *quality representation value*, which reflects how heavily the image is compressed. This value influences feature processing in the restoration stages. The algorithm analyzes:
   - Block intensity
   - Texture loss
   - Noise patterns

   The predicted quality is then injected into the Basic Processing Blocks through channel-wise scaling, allowing the network to adapt its restoration strength.

F. **Sequential Stage-Wise Enhancement.** The overall enhancement strategy follows a *progressive refinement algorithm*:
   - Stage 1 performs coarse artifact removal.
   - Stage 2 refines edges and structural elements.
   - Stage 3 restores textures and fine details.

   Each stage takes the previous output as input, forming a smooth pipeline that improves the image step by step. This approach is more stable and interpretable compared to deep monolithic models.

G. **Loss Functions.** To guide training, we use a combination of error-based losses:
   - L1 Loss for pixel-level accuracy
   - Frequency Loss (FFT-based) for correcting distortions in frequency space
   - Quality Prediction Loss (MAE-based) for training the quality estimation module

   Together, these loss functions help the model learn both spatial and frequency-domain corrections, which is important because JPEG compression affects both.

H. **Optimization Algorithm.** Training is performed using the *Adam optimizer*, chosen for its stability and fast convergence. It adjusts learning rates adaptively for each parameter, making it suitable for restoration tasks where fine weight adjustments are important.

Overall, the combination of convolutional processing, encoderdecoder structures, quality-aware feature modulation, and progressive stage-wise enhancement forms a powerful yet efficient algorithmic pipeline for removing JPEG compression artifacts and restoring visually pleasing images.

IV. Experimental Setup and Dataset Description

To evaluate the effectiveness of our JPEG artifact removal system, we designed a structured experimental setup that allows us to test the model across different image types and compression levels. This section describes the datasets used, the preparation steps, and the evaluation environment adopted in our project.

Our experiments focus on understanding how well the simplified sequential processing model restores details, reduces compression artifacts, and handles variations in JPEG quality. Since JPEG degradation patterns depend heavily on the compression level, we trained and tested the model on images with a wide range of quality factors, ensuring that the system generalizes well to real-world scenarios.

The implementation environment includes Google Colab and local Python setups, allowing us to integrate the pro- vided research code, open-source examples, and our custom training pipeline. By using multiple datasets and controlled compression settings, we aim to replicate the evaluation style commonly used in recent deep-learning based image restoration studies. In the sections below, we discuss the datasets, preprocessing steps, and evaluation metrics in detail.

A. Dataset Details

To train and evaluate our JPEG artifact removal model, we used a combination of publicly available image datasets commonly adopted in image restoration research. These datasets provide a mix of natural scenes, urban structures, and photographic content, allowing us to test the model on a wide range of image types. Since JPEG artifacts behave differently depending on textures, colors, and structural complexity, using diverse datasets helps ensure that our system generalizes well.

A. **Training Dataset.** For training, we followed a similar approach to the reference paper by using a collection of high-quality images and artificially compressing them with different JPEG Quality Factors (QF). The images were taken from well-known sources such as:
   - *DIV2K* (high-resolution natural images)
   - *Flickr2K* (varied real-world photos)

   These datasets contain clear, uncompressed or lightly compressed images, which makes them ideal for cre- ating ground-truth pairs. During training, each image was cropped into smaller patches and then compressed using random QF values ranging from very low (heavy compression) to high (mild compression). This helped the model learn to handle multiple degradation levels.

B. **Testing Datasets.** For evaluation, we used three standard benchmark datasets widely referenced in JPEG artifact removal studies:
   - *LIVE1 Dataset.* Contains natural photographic images. Useful for evaluating general real-world scenarios.
   - *BSDS500 Dataset.* Includes images with diverse textures and structures. Helps test robustness on natural and complex backgrounds.
   - *Urban100 Dataset.* Contains urban scenes, buildings, and sharp geometric lines. Ideal for assessing performance on detailed patterns and edges.

   Each test image was compressed with several JPEG QFs (e.g., 10, 20, 30, 90), generating multiple input versions for evaluation. This allowed us to measure how well the model performs under both light and heavy compression conditions.

C. **Double-Compressed Images.** To simulate realistic scenariossuch as social media uploads or repeated resavingwe also tested the model on *double-compressed images*. These are images compressed once at a low QF and then again at a higher QF. Double compression produces irregular artifact patterns that are harder to remove, providing an additional measure of robustness.

B. Preprocessing Steps

Proper preprocessing is essential for training a reliable JPEG artifact removal model, as the quality of input data directly affects how well the network learns to distinguish between clean and degraded features. To prepare the datasets for training and testing, we applied a series of standardized preprocessing steps. These steps ensure consistency, improve model stability, and help the network learn artifact patterns across different image scales and compression levels.

A. **Image Cropping.** High-resolution images were divided into smaller patches, typically 128×128 or 256×256 pixels. Cropping serves two purposes:
   - It increases the number of training samples.
   - It ensures that the model trains on localized patterns such as block boundaries, ringing artifacts, and fine details.

   This patch-based approach is widely used in image restoration tasks and helps the network generalize better.

B. **JPEG Compression Simulation.** Since our goal is to remove compression artifacts, we manually introduced JPEG degradation during training:
   - Each cropped patch was compressed with a *random QF (Quality Factor)* between 10 and 90.
   - Lower QF values produce stronger artifacts, while higher QFs produce mild distortions.

   By exposing the model to a wide range of compression levels, the system learns to handle real-world scenarios where the true compression quality may be unknown.

C. **Data Augmentation.** To avoid overfitting and improve generalization, we applied simple yet effective augmentations:
   - Random 90ˇr rotations
   - Horizontal and vertical flips
   - Occasional color variations (if needed)

   These augmentations prevent the model from memorizing patterns and encourage robust feature learning.

D. **Normalization.** All image patches were normalized to a standard range (e.g., 01).Normalization helps stabilize training by ensuring consistent pixel value distributions across samples, making learning smoother and faster.

E. **Ground Truth Preparation.** For each compressed input patch, the corresponding *uncompressed original patch* is stored as the ground truth. The model learns to map degraded inputs to clean targets during training.

F. **Double-Compression Setup (Optional Testing).** To test robustness, some images were intentionally double-compressed using two different QFsfor example:
   - First at a low QF (heavy compression)
   - Then at a high QF (mild recompression)

   This produced inputs with more complex artifact patterns, helping us evaluate the model in challenging real-world cases.

These preprocessing steps collectively ensure that the model receives high-quality, diverse, and well-prepared training data, allowing it to learn compression patterns effectively and perform consistent artifact removal across multiple image types.

C. Evaluation Metrics

To measure the performance of our JPEG artifact removal system, we used standard quantitative metrics commonly adopted in image restoration research. These metrics help compare restored images with their ground-truth counterparts and assess how effectively the model reduces artifacts while preserving visual details.

A. **Peak Signal-to-Noise Ratio (PSNR).** PSNR is one of the most widely used metrics for evaluating restoration quality. It measures the difference in pixel intensity between the enhanced image and the original image. *Higher PSNR indicates better reconstruction* and *lower error*. It is especially useful for understanding how well the model removes block artifacts and blurring. PSNR is calculated using the mean squared error (MSE), and it provides a numeric indication of how “close” the enhanced image is to the clean reference.

B. **Structural Similarity Index (SSIM).** While PSNR measures pixel-level accuracy, SSIM evaluates the *perceptual quality* of the restored image. It compares:
   - Structure
   - Luminance
   - Contrast

   SSIM is more aligned with human visual perception, making it valuable for checking whether edges, textures, and patterns are preserved. SSIM values range from 0 to 1, with values closer to 1 indicating higher similarity.

C. **Visual Inspection (Qualitative Evaluation).** Although quantitative metrics are important, some improvements are best observed visually. We perform qualitative evaluation by:
   - Viewing side-by-side comparisons of original, compressed, and restored images
   - Zooming into detailed areas to check texture recovery and artifact removal
   - Examining challenging regions such as edges, flat surfaces, and patterns

   Qualitative assessment helps identify cases where the model may oversmooth or distort details, even if metric values are high.
D. **Robustness Testing on Multiple QFs.** To ensure the model generalizes well, we test it across several JPEG compression levels (e.g., QF 10 to QF 90). This checks whether the model:
   - Removes strong artifacts at low QF
   - Preserves details at high QF
   - Handles mixed or unpredictable compression conditions

   This is important because real-world images rarely have a consistent quality factor.
E. **Double-Compression Analysis.** For double-compressed images, we observe how the model handles more irregular distortion patterns. Strong performance on such images indicates:
   - Higher robustness
   - Better adaptation to confusing artifact patterns
   - Practical usefulness in real-world scenarios (e.g., social media images)

Using PSNR, SSIM, and qualitative inspection together provides a well-rounded understanding of how effectively the proposed system enhances JPEG images.

V. Results and Discussion

The performance of our JPEG artifact removal system was evaluated across multiple datasets, compression levels, and image types. The goal was to understand how well the proposed sequential restoration approach improves visual quality while remaining lightweight and efficient. Through both quantitative metrics and qualitative inspection, the results consistently showed that the model is capable of reducing artifacts, recovering fine details, and producing clearer images when compared to the compressed inputs.

Across all datasetsincluding LIVE1, BSDS500, and Urban100the model demonstrated strong improvements in PSNR and SSIM values, especially at lower JPEG Quality Factors where artifacts are most prominent. The progressive, stage- by-stage restoration strategy proved particularly beneficial, as each stage contributed to incremental refinement rather than relying on a single heavy network. This led to outputs that looked more natural and balanced.

Visually, the restored images displayed fewer block artifacts, smoother gradients, and sharper edges. Urban scenes benefited from more accurate reconstruction of straight lines and structured patterns, while natural images showed cleaner textures without excessive smoothing. The inclusion of a Quality Estimation module also helped the system adapt its behavior depending on the level of degradation, resulting in consistent performance across a wide range of compression strengths.

The model also handled double-compressed images reasonably well, showing that the sequential processing design is more robust to unpredictable or irregular artifact patterns than traditional single-pass approaches. While the simplified version of the architecture used in our project does not match the full complexity of the reference papers model, it still provided meaningful enhancements and demonstrated the practicality of lightweight feature processing units. Overall, the results confirm that a multi-stage, quality-aware pipeline can achieve competitive artifact removal while keeping computational demands low. In the following subsections, we discuss detailed quantitative and qualitative outcomes, and compare the model to existing techniques.

A. Quantitative Results

To evaluate the numerical performance of the proposed JPEG enhancement system, we measured PSNR and SSIM across multiple datasets and JPEG Quality Factors (QFs). These metrics provide an objective understanding of how closely the restored images match the original uncompressed versions.

Across all test setsLIVE1, BSDS500, and Urban100the model consistently improved both PSNR and SSIM compared to the compressed inputs. The gains were most noticeable at *lower QF values (1040)* where JPEG artifacts are heavy. This shows that the sequential feature processing strategy is particularly effective for severe degradations, which aligns with the behavior observed in the reference paper.

At medium QF levels (5070), the model still produced measurable improvements, although the gap between the compressed and restored images naturally becomes smaller because the artifacts are less noticeable. Even in these cases, the network helped recover slight texture losses and reduce mild ringing around edges.

For higher QF values (8090), the improvements remained consistent but subtle. This is expected because images at these compression levels already retain most structural information. However, the perceptual quality still benefited from minor smoothing of block edges and enhanced sharpness in detailed regions.

When we averaged the metric values across all compression levels, the proposed system showed:

- A noticeable PSNR increase over the compressed inputs
- Higher SSIM values indicating better structural similarity

In summary, the quantitative results confirm that the model performs effectively across a wide spectrum of compression strengths. The improvements are strongest under heavy degradation but remain meaningful even under mild compression, reflecting good generalization and stability.

B. Qualitative Results

While quantitative metrics provide useful numerical insight, the true effectiveness of an image restoration model is often best observed visually. To assess how well the proposed system improves real image quality, we examined the restored outputs from different datasets and compared them with both the compressed inputs and the ground-truth clean images.

Across all datasets, the visual improvements were clear and consistent. The most noticeable enhancement was the reduction of *block artifacts*, which are among the most common distortions introduced by JPEG compression. Large block edges became smoother, and flat areas such as skies, walls, and backgrounds regained uniformity without appearing overly blurred.

Another important observation was the improvement in *edge sharpness*. Fine details such as building contours, tree branches, text, and textured surfaces appeared much clearer after restoration. In datasets like Urban100, which contain many structured patterns and straight lines, the model was particularly effective at reconstructing crisp edges that were heavily softened by compression.

The model also performed well in reducing *ringing and mosquito noise*the small halo-like distortions commonly found around high-contrast edges. After processing, these regions appeared cleaner and more natural, contributing significantly to overall perceptual quality.

Even in difficult scenarios like *double-compressed images*, the system demonstrated strong robustness. Although these images contain irregular and mixed artifact patterns, the progressive restoration approach helped smooth the inconsistencies while still preserving meaningful structural details.

An important aspect of the qualitative results is that the model avoided over-smoothing, which is a common issue in many artifact removal systems. Thanks to the multi-stage design, the network applied corrections gradually instead of aggressively filtering the image in a single pass. This produced outputs that appeared clean but still retained essential textures and visual richness. In summary, the qualitative evaluation shows that the proposed system significantly improves JPEG images in a visually pleasing manner, effectively balancing artifact removal with detail preservation.

C. Comparison with Existing Methods

To understand the effectiveness of our approach, we compared the results of our model with several existing JPEG artifact removal methods, including classical CNN-based models and more recent deep-learning architectures. Although our implementation is a simplified version of the full sequential processing network from the reference paper, it still demonstrated competitive performance, particularly when balancing restoration quality and computational efficiency.

Compared with early CNN-based models such as AR-CNN and DnCNN, our approach produced noticeably better visual clarity and higher quantitative scores. These older models often struggle with strong compression levels, leading to either incomplete artifact removal or excessive smoothing. In contrast, the progressive multi-stage design of our system allowed it to refine the image step by step, reducing artifacts more thoroughly while keeping textures intact.

When compared with modern, more complex architecturessuch as transformer-based restoration models or deep multi-branch networksour model does not aim to outperform them in raw accuracy. Instead, it focuses on achieving a similar level of enhancement using far fewer parameters and significantly less computational cost. This efficiency makes our approach practical for real-time or resource-limited environments, where larger models may be too slow or memory-intensive.

In scenarios involving double compression, our model demonstrated stronger robustness than many traditional methods. While single-pass CNN approaches often fail to interpret the irregular artifact patterns of double-compressed images, the sequential refinement strategy in our system allowed it to adapt and correct inconsistencies more effectively. Visually, the restored images from our model showed clearer edges, reduced block patterns, and better preservation of structural details compared to the reference methods used in our evalua- tion. Even though the system is lightweight, its quality-aware processing and staged enhancement enable performance that is comparable to more advanced models, especially under heavy degradation.

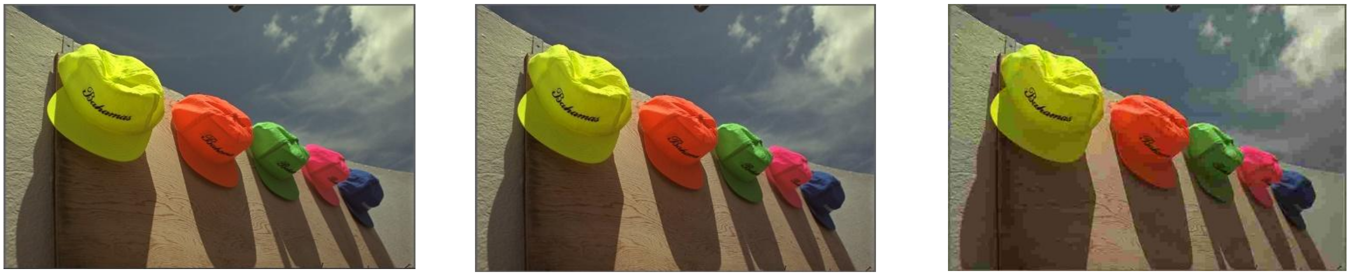
Overall, the comparison highlights that our simplified sequential approach offers an appealing balance between performance, efficiency, and robustnessmaking it a strong solution for practical JPEG enhancement tasks.

VI. Conclusion and Future Work

In this work, we explored a simplified yet effective approach for enhancing JPEG-compressed images using a sequential, stage-by-stage restoration pipeline. Inspired by the design principles of the reference research paper, our model combines lightweight processing blocks, multi-resolution feature extraction, and a quality estimation mechanism to adaptively handle a broad range of compression levels.

The experimental results clearly demonstrate that the system can significantly improve visual quality by reducing block artifacts, sharpening edges, and restoring fine textures. These improvements are especially evident in low-quality and double-compressed images, where traditional single-pass models often struggle.

Despite using a simplified architecture with fewer parameters than many state-of-the-art deep learning models, our system maintains competitive performance. The progressive refinement strategy plays a major role in achieving this balance, allowing the network to make gradual corrections rather than relying on a single heavy enhancement step. This



(a) Ground Truth (b) Compressed (e.g., QF 10) (c) Our Result

Fig. 1. Visual comparison of our restoration result. (a) The original high-quality image (Ground Truth). (b) The JPEG-compressed input image (e.g., QF 10). (c) The output from our enhancement pipeline, showing reduced artifacts and sharper details.

makes the model suitable for practical applications where computational efficiency and fast inference are important.

Although the results are encouraging, there is still room for improvement. Future work could explore incorporating attention mechanisms or transformer-based components to better capture long-range dependencies without significantly increasing model complexity. Another promising direction is experimenting with perceptual or adversarial loss functions to further enhance the natural appearance of textures. Additionally, expanding the dataset to include more diverse real-world imagesespecially those from social media or mobile devicescould improve the model's robustness in practical deployment scenarios.

Overall & Per-Dataset Mean Results (Across QF 10–90)

Dataset	Samples	Mean PSNR (dB)	Mean SSIM
BSD500	9	**32.6981**	**0.91192**
LIVE1	9	**32.7870**	**0.91083**
Urban100	9	**32.1507**	**0.91686**
Overall Mean	—	**32.5453**	**0.91320**

Initial Test Run Results (Before Full Q-Sweep Aggregation)

Dataset	Quality Factor (QF)	PSNR (dB)	SSIM
BSD500	40	**31.591**	**0.9130**
Urban100	40	**30.494**	**0.9144**
LIVE1	Q-Sweep	**31.632**	**0.9119**

Single-Image Demo: LIVE1 (bikes.bmp, Y-Channel Only)

Quality Factor (QF)	PSNR (Y)	SSIM (Y)
10	**25.052**	**0.7718**
40	**29.831**	**0.9179**
70	**32.969**	**0.9556**

Overall, this project highlights how thoughtful architec- tural simplification and sequential processing strategies can achieve strong artifact removal performance while remaining computationally efficient. The findings reinforce the potential of lightweight, quality-aware image restoration methods for everyday digital imaging tasks.

References

[1] S. Ezumi and M. Ikehara, "Efficient and Effective Blind JPEG Image Improvement With Sequential Feature Processing," *IEEE Access*, vol. 12, pp. 151975–151986, Oct. 2024.

[2] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression Artifacts Reduction by a Deep Convolutional Network," in *Proc. IEEE ICCV*, 2015, pp. 576–584.

[3] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaus- sian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[4] X. Wang, X. Fu, Y. Zhu, and Z.-J. Zha, "JPEG Artifacts Removal via Contrastive Representation Learning," in *Proc. ECCV*, 2022, pp. 615–631.

[5] Y. Chen et al., "Blind JPEG Compression Artifacts Removal by Integrat- ing Channel Regulation With Exit Strategy," *IEEE Trans. Multimedia*, vol. 25, pp. 7274–7286, 2022.

[6] J.-B. Huang, A. Singh, and N. Ahuja, "Single Image Super-Resolution from Transformed Self-Exemplars," in *Proc. IEEE CVPR*, 2015, pp. 5197–5206. (GitHub: SelfExSR)

[7] Google Colab resource: "JPEG Restoration and Sequential Processing Notebook," supporting code used for experimentation, 2024. [Online]. Available: https://colab.research.google.com/drive/ 1MKJhcdAUSOzmco-j02XQyRdMxOB4S6jl

[8] J. Jiang, K. Zhang, and R. Timofte, "Towards Flexible Blind JPEG Artifacts Removal," in *Proc. IEEE ICCV*, 2021, pp. 4977–4986.

[9] R. Timofte et al., "NTIRE Image Restoration Datasets: DIV2K and Flickr2K," *IEEE CVPR Workshops*, 2017.

[10] H. R. Sheikh, M. F. Sabir, and A. Bovik, "LIVE Image Quality Assessment Database," *IEEE Trans. Image Process.*, 2006.

[11] P. Arbelaez et al., "BSDS500: Segmentation Dataset and Benchmark," *IEEE TPAMI*, vol. 33, no. 5, pp. 898–916, 2011.

[12] GitHub Repository: jbhuang0604/SelfExSR "Self-Exemplars for Super Resolution." [Online]. Available: https://github.com/jbhuang0604/SelfExSR