



Smart Visual Optimizer: An Adaptive Human-in-the-Loop System for Intelligent Image Enhancement and Quality-Aware Decision Support

Maryam Alaa Zaki¹, Fathi A. Hamhoum^{*2}

Department of Computer Engineering and Sciences, Faculty of Science, University of Zawia, Libya

Corresponding Author's Email: Fathi.Hamhoum@zu.edu.ly

^{*2} <https://orcid.org/0009-0005-1903-0853>

Received: 25/8/2026 Revised: 30/8/2026 Accepted: 15/9/2026 Publication: 30/9/2026

Abstract

Image enhancement remains a challenging problem due to the diversity of image degradation types, varying user preferences, and different application requirements. Traditional enhancement methods often rely on fixed processing pipelines that may not adapt effectively to different image characteristics, which can produce over-enhanced or visually inconsistent results.

This paper presents *Smart Visual Optimizer*, a human-centered adaptive image enhancement framework that combines scene-aware analysis, mode-specific image processing pipelines, objective quality assessment, and interactive user control. The proposed system analyzes image characteristics and supports five enhancement modes: AI Super, Product, Social, Low Light, and Scan. Each mode applies a dedicated processing route designed for its target image category, such as product photographs, portrait images, low-light scenes, general images, and scanned documents. The framework computes quality indicators derived from brightness, contrast, sharpness, saturation, highlight distribution, and shadow distribution, then displays the enhanced result with quality metrics and system recommendations.

Experimental evaluation was conducted using 25 real-world images across the supported categories. The results showed consistent improvement in the average quality score across all tested categories, with the strongest improvements observed in Low Light, Document, and General images. Product and Social modes applied more conservative enhancement to preserve realistic appearance, natural colors, skin texture, and identity.

The proposed framework demonstrates that combining adaptive image processing with Human-in-the-Loop interaction can improve practical image enhancement systems by providing automated optimization, visual comparison, quality-aware feedback, and user-controlled refinement.

Index Terms

Image Enhancement, Image Restoration, Human-Computer Interaction, Human-in-the-Loop, Artificial Intelligence, Image Quality Assessment, Adaptive Image Processing, Computer Vision

I. INTRODUCTION

Image enhancement has become an essential component of modern computer vision systems and is widely used in digital photography, social media applications, e-commerce platforms, document digitization, surveillance systems, and mobile imaging devices. The primary objective of image enhancement is to improve visual quality, increase the visibility of important details, and preserve the natural appearance of the input image. With the rapid growth of visual content captured by mobile cameras and uploaded to digital platforms, there is an increasing need for image enhancement systems that are both adaptive and efficient.

Traditional image enhancement methods, such as Histogram Equalization (HE), Contrast Limited Adaptive Histogram Equalization (CLAHE), image sharpening, denoising filters, adaptive thresholding, morphological operations, and color correction, remain widely used because of their simplicity, interpretability, and low computational cost. These techniques are especially useful in real-time systems where fast processing and predictable behavior are required. In this work, several OpenCV-based operations are used to support contrast enhancement, denoising, document binarization, morphology-based cleanup, and scan

enhancement [1–4]. However, traditional approaches often depend on fixed parameters and predefined processing rules, which can limit their effectiveness when images differ significantly in lighting conditions, content type, noise level, color distribution, and sharpness.

Recent deep learning-based methods have achieved strong performance in image enhancement and restoration tasks. Transformer-based and all-in-one restoration models, such as Retinexformer [5], Restormer [6], and Multimodal Prompt Perceiver [7], have shown the ability to restore degraded images and enhance visual quality across different degradation types. In addition, Real-ESRGAN has been used for real-world blind super-resolution [8], while GFPGAN has been proposed for blind face restoration [9]. Although these methods are powerful, they may introduce unnatural details or hallucinated textures when applied to images that are already clear or when the image content does not match the model objective. For example, aggressive super-resolution can produce artificial-looking backgrounds in product images, while face restoration may alter facial identity or produce overly smoothed portraits.

Another important limitation is that many enhancement systems apply a single processing strategy to all types of images. In practice, different image categories require different enhancement objectives. Low-light images need shadow recovery and noise control, product photographs require edge clarity while preserving realistic backgrounds, portrait or social images need safe enhancement without altering facial features, and scanned documents require text readability, background normalization, and document-specific binarization. Document enhancement is particularly different from general image enhancement, since printed documents, handwriting, old manuscripts, and degraded archival pages may require different processing routes. Public benchmarks such as DIBCO/H-DIBCO, NoisyOffice, READ-BAD, and IAM highlight the diversity of document image problems, including binarization, noisy printed text enhancement, archival document analysis, and handwriting recognition [10–13].

To address these limitations, this paper proposes *Smart Visual Optimizer*, a Human-Centered Adaptive Image Enhancement Framework designed to enhance multiple image categories using scene-aware processing and user-guided control. The proposed framework first analyzes the input image using measurable visual characteristics such as brightness, saturation, sharpness, noise level, edge density, face presence, and document-like structure. Based on this analysis, the system selects an appropriate enhancement mode, including *AI Super*, *Product*, *Social*, *Low Light*, and *Scan*. Each mode applies a dedicated processing pipeline tailored to the visual requirements of that category. The final system also includes an enhancement strength control that allows the user to blend the enhanced result with the original image, supporting a Human-in-the-Loop interaction style consistent with human-centered AI principles [14].

The proposed framework is implemented as a complete real-time application using Python, OpenCV, FastAPI, and Flutter. The backend performs image analysis and enhancement, while the frontend provides an interactive interface for uploading images, selecting enhancement modes, comparing original and enhanced outputs, and adjusting enhancement strength. This design makes the system practical for real-time use while maintaining transparency and user control.

The main contributions of this paper are summarized as follows:

- Development of a scene-aware adaptive image enhancement framework that supports multiple image categories, including general images, product images, portraits, low-light images, and scanned documents.
- Design of dedicated enhancement pipelines for each mode, combining classical image processing techniques with optional AI-based restoration where appropriate.
- Implementation of a practical Scan mode that distinguishes between printed documents and old-paper or handwriting-like inputs, using document-specific preprocessing, binarization, morphology-based cleanup, and text readability enhancement.
- Integration of a Human-in-the-Loop enhancement strength control that allows users to adjust the balance between natural appearance and enhancement intensity.
- Implementation of a complete real-time system using Python, OpenCV, FastAPI, and Flutter, enabling interactive enhancement and visual comparison of results.

II. LITERATURE REVIEW

Image enhancement has evolved significantly from classical image processing techniques to modern learning-based restoration frameworks. Recent studies have focused on improving image quality under challenging conditions such as low illumination, noise, blur, compression artifacts, and document degradation. Since the proposed system is designed as a practical multi-

mode enhancement application, this section reviews the most relevant work related to traditional image processing, low-light enhancement, image restoration, document image enhancement, image quality assessment, and human-centered artificial intelligence.

A. Traditional Image Enhancement and OpenCV-Based Processing

Classical image enhancement techniques remain important in real-time applications because they are computationally efficient, interpretable, and suitable for deployment without large training datasets. Common methods include contrast enhancement, Histogram Equalization (HE), Contrast Limited Adaptive Histogram Equalization (CLAHE), denoising, sharpening, adaptive thresholding, color correction, and morphological operations. These techniques are widely used to improve visual clarity, reduce image noise, enhance local contrast, and extract readable structures from degraded images.

In the proposed framework, several OpenCV-based techniques are used directly in the enhancement pipelines. CLAHE is used to improve local contrast, especially in low-light images and document scans. Non-local means denoising is applied to reduce noise while preserving image details. Adaptive thresholding is used in the Scan mode to convert degraded document images into readable binary outputs under non-uniform illumination. Morphological operations are also used to remove small artifacts and strengthen text structures in scanned documents [1–4]. These operations are particularly suitable for the proposed system because they provide fast processing and predictable results in an interactive application environment.

However, traditional enhancement methods usually rely on fixed parameters and predefined rules. As a result, they may fail when the input image contains complex degradation, strong shadows, mixed lighting, weak text strokes, or content-specific requirements. This limitation motivates the need for adaptive pipelines that analyze image characteristics before applying enhancement operations.

B. Deep Learning–Based Low–Light and Image Restoration

Deep learning has become a major direction in image enhancement and restoration research. CNN–based methods first improved restoration quality by learning degradation patterns from data, while more recent Transformer–based methods have improved the ability to model long–range dependencies and global image context. These methods are especially effective in challenging tasks such as low–light enhancement, denoising, deblurring, and mixed degradation restoration.

Retinexformer proposed a Retinex–inspired Transformer architecture for low–light image enhancement and demonstrated strong performance in improving illumination while preserving image structure [5]. Customized learnable priors have also been introduced to improve the adaptability of low–light enhancement under different illumination conditions [15]. A recent survey further highlighted the progress of deep learning–based low–light enhancement and discussed the limitations of traditional enhancement approaches in complex real–world scenarios [16].

Restormer introduced an efficient Transformer architecture for high–resolution image restoration tasks such as denoising, deblurring, and artifact removal [6]. Multimodal Prompt Perceiver further extended all–in–one image restoration by using adaptive prompts to improve generalization across different degradation types [7]. These studies show the strength of modern restoration models; however, they often require high computational resources and may not always be suitable for lightweight real–time applications.

In the proposed system, learning–based restoration models are considered as optional enhancement components rather than being applied aggressively to every image. Real–ESRGAN is related to real–world blind super–resolution and detail recovery [8], while GFPGAN is related to blind face restoration [9]. During practical testing, however, these models were found to be useful only in specific conditions. Aggressive super–resolution may create artificial textures in product backgrounds, and face restoration may alter facial identity or over–smooth skin. Therefore, the final implementation adopts conservative and content–aware enhancement routes, applying AI–based restoration only when appropriate.

C. Document Image Enhancement and Scan Processing

Document image enhancement is different from general image enhancement because the main objective is not only visual attractiveness, but also readability. Scanned documents, handwritten notes, old manuscripts, and photographed pages often suffer from shadows, uneven illumination, paper texture, ink fading, background stains, and small text artifacts. For this reason, document enhancement usually requires specialized processing such as background normalization, binarization, adaptive thresholding, and morphology–based cleanup.

DocUNet addressed the problem of document image unwarping by correcting distortions caused by curved or folded document pages captured using mobile cameras [10]. This is related to document preprocessing because geometric distortion can reduce readability and affect later enhancement stages. DIBCO/H–DIBCO competitions have also played an important role in document image binarization research by providing benchmarks for printed and handwritten document binarization [11]. These benchmarks are directly related to the Scan mode in the proposed system, where the goal is to transform degraded document images into clearer and more readable outputs.

Handwritten document processing is also relevant to the proposed Scan mode. The IAM Handwriting Database provides scanned handwritten pages, lines, and words for offline handwriting recognition research [12]. Although the proposed system does not perform OCR, such datasets demonstrate the importance of preserving handwriting strokes and avoiding excessive filtering that may remove small but meaningful text details. Therefore, the implemented Scan pipeline focuses on improving readability while preserving the structure of handwritten and printed text.

D. Image Quality Assessment

Image Quality Assessment (IQA) is important for evaluating whether an enhancement operation improves or degrades the visual quality of an image. Full–reference metrics compare an enhanced image with a reference image, while no–reference metrics estimate quality without requiring a ground–truth image. In real–world enhancement applications, no–reference assessment is especially useful because reference images are usually unavailable.

The Structural Similarity Index Measure (SSIM) is one of the most widely used perceptual image quality metrics and evaluates image similarity based on luminance, contrast, and structure [13]. BRISQUE is a no–reference image quality assessment method that estimates visual quality from natural scene statistics without requiring a reference image [17]. In the proposed framework, quality–aware evaluation is used to support enhancement analysis and provide a more objective understanding of output quality. However, the system does not rely only on automatic quality scores, since user preference and content type can also affect the perceived quality of the final image.

E. Human–Centered Artificial Intelligence and User Control

Human–Centered Artificial Intelligence emphasizes reliability, transparency, and meaningful human control in intelligent systems. Shneiderman argued that AI systems should be designed to support human needs and provide reliable, safe, and trustworthy interaction [14]. This perspective is especially important in image enhancement, because the best result is not

always the strongest enhancement. Users may prefer a natural image, a sharper product photo, a brighter low-light image, or a cleaner document scan depending on the task.

Most image enhancement systems provide limited user control and often operate as black-box tools. In contrast, the proposed framework includes a Human-in-the-Loop mechanism through an enhancement strength slider. This allows the user to blend the enhanced result with the original image and adjust the final appearance according to personal preference. Such interaction helps balance automatic enhancement with user judgment and supports a more practical enhancement workflow.

F. Research Gap

Based on the reviewed literature, several gaps remain. First, many advanced restoration methods focus mainly on restoration accuracy and do not provide enough user control. Second, most enhancement systems use a single processing strategy, even though different image categories require different enhancement objectives. Third, document images require specialized processing that differs from general image enhancement, especially when dealing with handwriting, old paper, and degraded scans. Fourth, quality assessment is often used only for evaluation, rather than being integrated into the enhancement workflow. Finally, Human-in-the-Loop interaction remains underutilized in practical image enhancement systems.

To address these gaps, this research proposes *Smart Visual Optimizer*, a Human-Centered Adaptive Image Enhancement Framework. The system combines image feature analysis, scene-aware mode selection, dedicated enhancement pipelines, quality-aware evaluation, and interactive user control within a complete real-time application. Unlike black-box enhancement models, the proposed framework is designed to be practical, transparent, adaptive, and suitable for multiple image categories, including general images, product photos, social portraits, low-light images, and scanned documents.

TABLE I
SUMMARY OF RELATED WORK AND RELEVANCE TO THE PROPOSED SYSTEM.

Research Area	Representative References	Relevance to This Work
Low-light enhancement	Retinexformer [5], learnable priors [15], low-light survey [16]	Supports the Low Light mode for illumination recovery and noise control.
Image restoration	Restormer [6], MPerceiver [7], Real-ESRGAN [8]	Supports adaptive restoration and optional AI-based enhancement for degraded images.
Face restoration	GFPGAN [9]	Provides background for the Social mode and highlights the need to avoid identity
Document enhancement	DocUNet [10], DIBCO [11], IAM [12]	Supports the Scan mode for document readability, binarization, and handwriting
Image quality	SSIM [13], BRISQUE [17]	Supports objective evaluation and quality-aware enhancement analysis.
Human-centered AI	Shneiderman [14]	Supports the Human-in-the-Loop design and user-controlled enhancement
OpenCV-based processing	Thresholding, CLAHE, denoising, morphology [1–4]	Supports fast, interpretable, and real-time enhancement pipelines.

III. METHODOLOGY AND SYSTEM DESIGN

This research proposes *Smart Visual Optimizer*, a Human-Centered Adaptive Image Enhancement Framework designed to improve image quality across multiple image categories while preserving visual realism and allowing user interaction. The framework integrates image feature analysis, scene-aware mode selection, mode-specific enhancement pipelines, quality-aware evaluation, optional AI-assisted restoration, and Human-in-the-Loop (HITL) control within a unified real-time architecture.

Unlike single-filter enhancement systems, the proposed framework does not apply the same processing strategy to all input images. Instead, it analyzes the visual characteristics of the image and selects the most suitable enhancement mode. The final implementation supports five main modes: *AI Super*, *Product*, *Social*, *Low Light*, and *Scan*. Each mode is implemented as a dedicated enhancement route with different objectives and processing operations. The system was implemented using Python, OpenCV, FastAPI, and Flutter. OpenCV operations were used for image analysis, enhancement, denoising, thresholding, morphology, and document cleanup [1–4]. Real-ESRGAN was considered as an optional AI-based restoration component for selected low-detail images [8], while the final design avoids aggressive AI restoration when it may introduce artificial textures or alter important visual content.

A. System Overview

The proposed framework consists of six major components:

- 1) Image acquisition module.
- 2) Feature extraction and scene analysis module.
- 3) Scene-aware mode selection module.

- 4) Adaptive enhancement engine.
- 5) Quality-aware evaluation module.
- 6) Human-in-the-Loop interaction module.

The system receives an input image from the user, extracts measurable visual features, identifies the most suitable enhancement category, applies the corresponding enhancement pipeline, evaluates the output quality, and returns the enhanced image to the user interface. The user can then compare the original and enhanced images and adjust the enhancement strength according to personal preference.

B. System Architecture

The Smart Visual Optimizer framework follows a client-server architecture. The frontend application was developed using Flutter to provide a responsive and interactive user interface. The backend enhancement engine was implemented using Python and FastAPI. OpenCV was used for the core image processing operations, including contrast enhancement, denoising, sharpening, color correction, thresholding, and morphology-based document cleanup.

The workflow begins when the user uploads an image through the Flutter interface. The image is sent to the FastAPI backend through an HTTP request. The backend reads the image, extracts visual features, determines the suitable processing mode, applies the selected enhancement pipeline, and returns the enhanced result together with quality metrics. The frontend then displays the original and enhanced images using an interactive comparison interface and allows the user to adjust the enhancement strength.

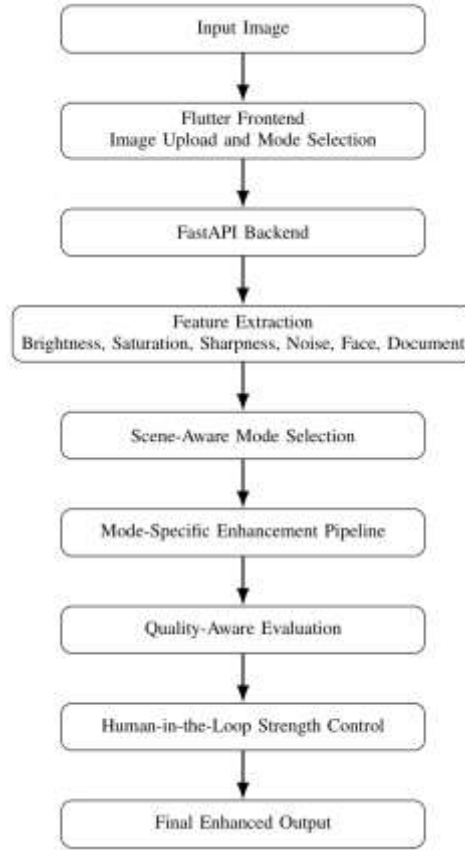


Fig. 1. Overall architecture of the proposed Smart Visual Optimizer framework.

Figure 1 illustrates the complete workflow of the proposed framework. The architecture combines automatic image analysis, adaptive mode selection, mode-specific enhancement, quality evaluation, and user-guided control within a unified enhancement system.

C. Feature Extraction and Scene Analysis

Different image categories require different enhancement objectives. Therefore, the proposed framework incorporates a feature extraction and scene analysis stage before applying enhancement. The purpose of this stage is to measure the visual properties of the image and guide the selection of the appropriate enhancement mode.

For each input image, the following features are extracted:

- **Brightness:** used to detect low-light or overexposed images.
- **Saturation:** used to analyze color intensity and identify images requiring color correction.
- **Sharpness:** used to estimate image clarity and detect blurred or low-detail images.
- **Noise level:** used to determine whether denoising is required.
- **Highlight ratio:** used to detect strong bright regions and protect overexposed areas.
- **Shadow ratio:** used to detect underexposed regions in low-light images.
- **Face presence:** used to activate safe portrait enhancement in the Social mode.
- **Document-like structure:** used to identify scanned documents, handwriting, printed pages, and old-paper inputs.

The extracted features are used to classify the image into one of the supported categories: general image, product image, portrait or social image, low-light image, or document image. This scene-aware analysis prevents the system from applying unsuitable enhancement operations to the wrong type of image.

D. Adaptive Enhancement Modes

The adaptive enhancement engine consists of five dedicated enhancement modes. Each mode has a specific objective and applies a different processing route. Table II summarizes the supported modes and their practical role in the proposed system.

TABLE II
ENHANCEMENT MODES USED IN THE PROPOSED FRAMEWORK.

Mode	Target Input	Enhancement Objective
AI Super	General, compressed, or low-detail	Improves general visual quality while avoiding unnecessary modification of already
Product	Product photographs	Enhances product clarity, preserves realistic colors, and avoids artificial-looking
Social	Portraits and social media	Improves visual appearance while preserving facial realism, skin texture, and
Low Light	Dark or poorly illuminated	Recovers shadows, improves brightness, and controls noise without
Scan	Printed documents, handwriting, old paper, and scanned pages	Improves readability using background normalization, binarization, denoising, and

E. AI Super Mode

The AI Super mode is designed for general-purpose image enhancement. It analyzes the input image and applies a conservative enhancement strategy according to the image condition. If the image is already clear and sharp, the system avoids aggressive processing and preserves the original appearance. If the image is soft, compressed, or low-detail, the system may apply stronger enhancement or optional AI-assisted restoration.

Real-ESRGAN was considered in this mode as an optional super-resolution component for real-world blind image restoration [8]. However, practical testing showed that applying super-resolution to every image can produce artificial textures, especially in natural backgrounds and product scenes. Therefore, the final design uses AI-assisted restoration selectively and prioritizes natural preservation when the image is already visually acceptable.

F. Product Mode

The Product mode is designed for e-commerce and advertising images. The goal is to improve product clarity while preserving realistic color, texture, and background appearance. The mode applies controlled denoising, local contrast enhancement, white balance correction, mild vibrance adjustment, highlight protection, and soft sharpening. The processing is intentionally conservative because product images require realism and should not contain hallucinated textures or artificial-looking details.

During practical testing, aggressive super-resolution was found to sometimes distort product backgrounds and produce painterly artifacts. Therefore, the final Product mode uses an OpenCV-based detail-preserving route rather than applying AI restoration by default. This design makes the mode more suitable for real product presentation where visual trust and accurate appearance are important.

G. Social Mode

The Social mode is designed for portraits and social media images. Its main objective is to improve appearance while avoiding identity alteration. The pipeline applies mild contrast enhancement, color correction, controlled vibrance adjustment, highlight reduction, and safe sharpening. When a face is detected, the system uses a conservative enhancement route to preserve facial structure, skin texture, and natural appearance.

GFPGAN is related to blind face restoration and can improve degraded face images [9]. However, in practical enhancement workflows, strong face restoration may alter facial identity or produce over-smoothed results. For this reason, the final Social mode avoids aggressive face restoration and focuses on safe visual enhancement.

H. Low Light Mode

The Low Light mode is designed for dark images and images captured under weak illumination. This pipeline aims to brighten shadow regions, improve local contrast, and control noise. It applies adaptive brightness recovery, CLAHE-based local contrast enhancement, denoising, and mild color correction. The enhancement is carefully controlled to avoid noise amplification and overexposure.

This design is motivated by the limitations of traditional fixed-parameter enhancement and by recent low-light enhancement research, which shows that illumination recovery must preserve structure and avoid unnatural brightness changes [5, 15, 16]. In the proposed system, the Low Light mode uses a practical OpenCV-based route suitable for real-time use.

I. Scan Mode

The Scan mode is designed for document images, including printed pages, handwriting, old paper, and photographed documents. Unlike general image enhancement, the main goal in this mode is readability rather than visual attractiveness. The pipeline includes grayscale conversion, background normalization, denoising, CLAHE-based contrast enhancement, ink enhancement, adaptive thresholding, connected component cleanup, and morphology-based text strengthening.

The Scan mode is related to document binarization research, where the objective is to convert degraded document images into clean and readable binary outputs. DIBCO/H-DIBCO benchmarks are widely used for evaluating document image binarization methods [11]. The proposed Scan pipeline also considers the importance of preserving handwriting strokes, which is relevant to handwritten document datasets such as IAM [12]. For old-paper and handwriting-like images, the system applies a stronger black-and-white route to improve text visibility and reduce background noise.

J. Quality Assessment Module

The quality assessment module computes objective indicators that describe the visual quality of the original and enhanced images. These indicators are used to analyze enhancement behavior and provide measurable feedback about the output. The final system does not rely only on automatic scores, since user preference and image category also affect perceived quality.

1) *Sharpness Estimation*: Image sharpness is estimated using the variance of the Laplacian operator. A higher value generally indicates stronger edge information and clearer details.

$$S = Var(\nabla^2 I) \quad (1)$$

where I represents the image intensity values, $\nabla^2 I$ represents the Laplacian response, and S represents the sharpness score.

2) *Brightness Estimation*: The average image brightness is computed using the mean intensity value:

$$B = \frac{1}{N} \sum_{i=1}^N I_i \quad (2)$$

where N denotes the total number of pixels and I_i represents the intensity value of pixel i .

3) *Contrast Estimation*: Image contrast is estimated using the standard deviation of pixel intensities:

$$C = \sigma(I) \quad (3)$$

where C represents the contrast score and $\sigma(I)$ is the standard deviation of the intensity values.

4) *Composite Quality Score*: A composite quality score is computed using a weighted combination of normalized quality indicators:

$$Q = w_1 S + w_2 C + w_3 B \quad (4)$$

where S represents sharpness, C represents contrast, B represents brightness, and w_1 , w_2 , and w_3 are weighting coefficients.

5) *Enhancement Improvement*: The relative improvement between the enhanced and original image is computed as follows:

$$Improvement(\%) = \frac{Q_{enhanced} - Q_{original}}{Q_{original}} \times 100 \quad (5)$$

where $Q_{enhanced}$ is the quality score of the enhanced image and $Q_{original}$ is the quality score of the input image.

K. Human-in-the-Loop Interaction

A key component of the proposed framework is the integration of Human-in-the-Loop interaction. Instead of forcing a fully automatic result, the system allows the user to control enhancement intensity through an interactive strength slider. This slider blends the enhanced output with the original image, allowing the user to choose a result that balances natural appearance and enhancement strength.

The blending operation is defined as:

$$I_{final} = (1 - \alpha)I_{original} + \alpha I_{enhanced} \quad (6)$$

where I_{final} is the final displayed result, $I_{original}$ is the original image, $I_{enhanced}$ is the processed image, and α is the user-controlled enhancement strength. When $\alpha = 0$, the output is identical to the original image. When $\alpha = 1$, the output is the fully enhanced image.

This interaction supports a human-centered design because the user remains involved in the final decision. It also reduces the risk of over-enhancement by allowing the user to adjust the result visually.

L. Research Workflow

The complete workflow of the proposed framework consists of the following stages:

- 1) The user uploads an image through the Flutter frontend.
- 2) The image is sent to the FastAPI backend.
- 3) The backend reads and prepares the input image.
- 4) Visual features are extracted from the image.
- 5) The system identifies the suitable enhancement mode.
- 6) A mode-specific enhancement pipeline is applied.
- 7) Quality indicators are computed for the original and enhanced images.
- 8) The enhanced image and metrics are returned to the frontend.
- 9) The user compares the original and enhanced images.
- 10) The user adjusts enhancement strength using the Human-in-the-Loop slider.
- 11) The final enhanced output is generated.

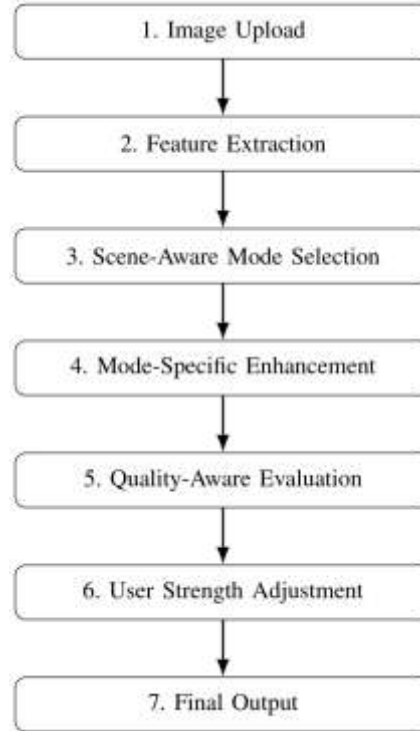


Fig. 2. Research workflow of the proposed enhancement framework.

TABLE III
IMPLEMENTATION SUMMARY OF THE PROPOSED SYSTEM.

Component	Implementation Details
Frontend	Flutter-based interface for image upload, mode selection, before-after comparison, and enhancement
Backend	FastAPI server responsible for receiving images, executing enhancement pipelines, and returning enhanced
Image Processing	Python and OpenCV operations, including CLAHE, denoising, sharpening, thresholding, morphology, and color
Enhancement Modes	AI Super, Product, Social, Low Light, and Scan modes.
AI-Assisted Restoration	Optional Real-ESRGAN-based restoration for selected low-detail images when appropriate.
Human-in-the-Loop	Strength slider that blends the original and enhanced images using a user-controlled parameter.
Output	Enhanced image, quality metrics, and visual comparison displayed in the frontend interface.

M. Implementation Summary

Table III summarizes the main implementation components used in the proposed system.

IV. IMPLEMENTATION

This section presents the implementation details of the proposed *Smart Visual Optimizer* framework, including the development environment, backend architecture, frontend interface, enhancement modules, system integration, and experimental testbed. The implementation was designed to support real-time image enhancement through a practical client-server architecture that combines Python-based image processing with an interactive Flutter user interface.

A. Development Environment

The proposed framework was implemented using open-source technologies to ensure flexibility, maintainability, and practical deployment. The backend was developed using Python because of its strong support for computer vision and image processing libraries. FastAPI was used to build the backend service and expose the image enhancement functionality through RESTful endpoints. OpenCV was used as the primary image processing library for feature extraction, denoising, contrast enhancement, thresholding, morphological processing, and document cleanup [1–4].

The frontend application was implemented using Flutter to provide a responsive and cross-platform user interface. The interface allows users to upload images, select enhancement modes, view original and enhanced results, compare images using a before-after slider, and control enhancement strength interactively. This design supports the Human-in-the-Loop concept by allowing the user to participate in the final enhancement decision rather than relying only on fully automatic processing.

Table IV summarizes the main technologies used in the implementation.

TABLE IV
DEVELOPMENT ENVIRONMENT AND SOFTWARE TECHNOLOGIES.

Component	Technology Used
Programming Language	Python for backend image processing and enhancement logic.
Backend Framework	FastAPI for receiving images, processing requests, and returning enhanced outputs.
Frontend Framework	Flutter for cross-platform user interface development.
Image Processing Library	OpenCV for enhancement, denoising, thresholding, morphology, and feature extraction.
Optional AI Module	Real-ESRGAN for selected super-resolution cases where AI-based restoration is appropriate [8].
Development Platform	Windows-based development environment.
Communication Method	RESTful API communication between the Flutter frontend and FastAPI backend.

B. Backend Architecture

The backend enhancement engine was implemented as a modular FastAPI service. The main endpoint receives an uploaded image together with the selected enhancement mode and strength value. The backend then reads the image, extracts visual features, applies the corresponding enhancement mode, computes quality indicators, blends the enhanced result with the original image according to the user-selected strength, and returns the final output to the frontend.

The backend implementation is organized into two main layers. The first layer is the API layer, which handles image upload, parameter reading, response formatting, and communication with the Flutter client. The second layer is the image processing layer, which contains the enhancement functions, feature analysis logic, mode-specific pipelines, and quality metric calculations. This modular structure improves maintainability and allows each enhancement mode to be modified independently.

The final backend supports the following enhancement modes:

- *AI Super* mode for general-purpose image enhancement.
- *Product* mode for product and e-commerce images.
- *Social* mode for portraits and social media images.
- *Low Light* mode for dark and poorly illuminated images.
- *Scan* mode for documents, handwriting, printed pages, and old paper.

Unlike the early prototype, the final implementation does not generate several enhancement candidates for each input image. Instead, it follows a single-pipeline optimization strategy. The system analyzes the image characteristics and then routes the input directly to the selected or most suitable processing pipeline. Each mode produces one optimized output through its dedicated enhancement route.

This design reduces repeated processing, lowers memory usage, and improves the response time of the FastAPI server. It also makes the system easier to control and more consistent with the real-time Human-in-the-Loop interaction requirement, where users expect fast visual feedback, quality metrics, and an adjustable enhancement result.

C. API Workflow

The backend workflow starts when the Flutter application sends an image to the FastAPI server. The server saves or reads the uploaded image, performs processing, and returns the enhanced output and quality metrics. The main processing stages are summarized as follows:

- 1) Receive image and enhancement parameters from the Flutter frontend.
- 2) Decode and validate the uploaded image.
- 3) Extract image features such as brightness, saturation, sharpness, and noise level.
- 4) Determine the active mode or use the user-selected mode.
- 5) Apply the mode-specific enhancement pipeline.
- 6) Compute quality indicators for the original and enhanced images.
- 7) Blend the enhanced image with the original image using the selected strength value.
- 8) Encode and return the final enhanced image and metrics to the frontend.

The practical backend workflow is shown in Fig. 3.

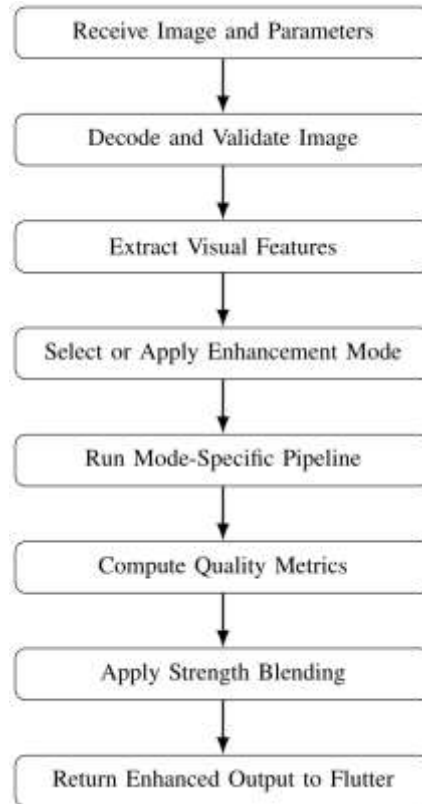


Fig. 3. Backend workflow of the Smart Visual Optimizer implementation.

D. Feature Extraction Module

Before applying enhancement, the system extracts visual features that describe the input image. These features help the system understand the image condition and support scene-aware processing. The extracted features include brightness, saturation, sharpness, contrast, noise level, highlight ratio, shadow ratio, face presence, and document-like structure.

Brightness is estimated from the average intensity of the image. Sharpness is estimated using the variance of the Laplacian response, where higher values indicate stronger edges and clearer details. Saturation is extracted from the color representation of the image and is used to determine whether color correction or vibrance adjustment is required. Shadow and highlight ratios are used to identify low-light or overexposed regions. For document images, structural cues such as edge density, background behavior, and text-like patterns are considered to support the Scan mode.

Table V summarizes the main extracted features and their implementation role.

TABLE V
EXTRACTED VISUAL FEATURES AND THEIR ROLE IN THE SYSTEM.

Feature	Role in Implementation
Brightness	Detects dark or overexposed images and supports Low Light processing.
Sharpness	Measures edge clarity and supports sharpening or detail-preservation decisions.
Contrast	Estimates intensity variation and supports contrast enhancement decisions.
Saturation	Measures color intensity and supports color correction and vibrance adjustment.
Noise Level	Indicates whether denoising should be applied.
Highlight Ratio	Detects bright regions and supports highlight protection.
Shadow Ratio	Detects underexposed regions and supports shadow recovery.
Face Presence	Supports safe Social mode enhancement and prevents aggressive face alteration.
Document Structure	Supports Scan mode activation for printed documents, handwriting, and old-paper inputs.

E. Adaptive Enhancement Pipeline

The adaptive enhancement engine applies a different processing route according to the active mode. Each mode is designed with a specific visual objective. This mode-specific design prevents unsuitable processing operations from being applied to the wrong image type.

The *AI Super* mode handles general images and compressed or low-detail images. It applies conservative enhancement and avoids unnecessary processing when the image is already clear. Real-ESRGAN is considered only for selected cases where super-resolution is useful [8]. This selective strategy reduces the risk of generating artificial details.

The *Product* mode is designed for e-commerce images. It applies controlled denoising, local contrast enhancement, mild sharpening, white balance correction, vibrance adjustment, and highlight protection. During implementation testing, aggressive AI super-resolution was avoided in this mode because it may produce unrealistic backgrounds or artificial textures. Therefore, the final Product mode uses a natural OpenCV-based enhancement route.

The *Social* mode is designed for portraits and social media images. It applies mild contrast enhancement, color correction, highlight reduction, and safe sharpening. Since face restoration models such as GFPGAN may alter facial identity when used aggressively [9], the final implementation uses a conservative face-safe route that preserves skin texture and natural appearance.

The *Low Light* mode is designed for dark images. It applies brightness recovery, CLAHE-based local contrast enhancement, denoising, and mild color correction. The goal is to brighten shadow regions while avoiding noise amplification and overexposure. This design is consistent with low-light enhancement research that emphasizes illumination recovery and structure preservation [5, 15, 16].

The *Scan* mode is designed for documents, handwriting, printed pages, and old paper. It applies grayscale conversion, background normalization, denoising, CLAHE-based contrast enhancement, ink enhancement, adaptive thresholding, connected component cleanup, and morphology-based text strengthening. This mode is directly related to document binarization and scan enhancement tasks [11, 12].

Table VI summarizes the main processing operations used in each enhancement mode.

F. AI-Assisted Restoration

The system includes optional AI-assisted restoration using Real-ESRGAN. Real-ESRGAN is a real-world blind super-resolution method designed to recover image details and improve visual resolution [8]. In the proposed implementation, this module is not applied as a default processing step for all input images. Instead, it is treated as a selective restoration component for images that are soft, compressed, or low-detail.

TABLE VI
IMPLEMENTATION OF ENHANCEMENT MODES AND MAIN PROCESSING OPERATIONS.

Mode	Main Operations	Implementation Objective
AI Super	Feature analysis, conservative enhance-	Improves general image quality while preserving already clear images.
Product	Denoising, CLAHE, white balance, vibrance adjustment, soft sharpening,	Improves product clarity while preserving realistic colors and backgrounds.
Social	Mild CLAHE, color correction, high-light reduction, safe sharpening,	Enhances portraits without altering identity or over-smoothing skin.
Low Light	Shadow recovery, CLAHE, denoising,	Brightens dark images while controlling noise and avoiding overexposure.
Scan	Grayscale conversion, background normalization, denoising, adaptive thresh-	Improves document readability and produces a clean scan-like output.

This design decision was based on practical testing. Although AI-assisted super-resolution can improve some degraded images, applying it aggressively to every image may introduce hallucinated textures, artificial details, or unrealistic visual patterns. This issue is especially noticeable in product images, where background realism and accurate texture preservation are important.

Therefore, the final framework prioritizes controlled OpenCV-based enhancement for most processing modes and uses AI-assisted restoration only when the image characteristics indicate that it is suitable. This conservative strategy helps preserve natural colors, realistic product appearance, and overall visual consistency while still allowing stronger restoration when it is beneficial.

G. Scan Mode Implementation

The Scan mode is one of the most important implementation components because it targets document readability rather than general visual attractiveness. The final Scan pipeline follows a black-and-white document enhancement route for handwriting, printed pages, and old-paper inputs.

The processing starts by converting the input image into grayscale. Background normalization is then applied to reduce uneven illumination and paper texture. Non-local means denoising is used to reduce noise while preserving text strokes [3]. CLAHE improves local contrast and makes weak text more visible [2]. Morphological black-hat processing is used to emphasize dark ink strokes. Adaptive thresholding converts the enhanced grayscale image into a binary image under non-uniform illumination [1]. Finally, connected component cleanup and morphological operations remove small artifacts and strengthen readable text structures [4].

The Scan implementation pipeline is summarized in Fig. 4.

H. Quality Assessment and Output Metrics

The backend computes several quality indicators for both the original and enhanced images. These indicators include brightness, contrast, sharpness, and other feature-based measurements. The purpose of these metrics is to provide objective feedback about the image condition and enhancement behavior.

The sharpness score is computed using the variance of the Laplacian operator. Brightness is computed using the average pixel intensity. Contrast is estimated using the standard deviation of pixel intensities. These measurements are combined into a composite quality score to describe the output quality. Although objective metrics are useful for analysis, the final system also considers user preference through the interactive strength control.

I. Human-in-the-Loop Interface

The Flutter frontend includes a Human-in-the-Loop interface that allows the user to control and inspect the enhancement result. The interface provides image upload, mode selection, enhancement preview, before-after comparison, quality metrics display, and strength adjustment.

The strength slider blends the original image and the enhanced image using a user-controlled parameter. This gives the user direct control over the final appearance and helps prevent over-enhancement. The blending operation is implemented as follows:

$$I_{final} = (1 - \alpha)I_{original} + \alpha I_{enhanced} \quad (7)$$

where I_{final} is the final output image, $I_{original}$ is the input image, $I_{enhanced}$ is the processed image, and α is the user-selected strength value. A value of $\alpha = 0$ preserves the original image, while $\alpha = 1$ applies the full enhanced result.

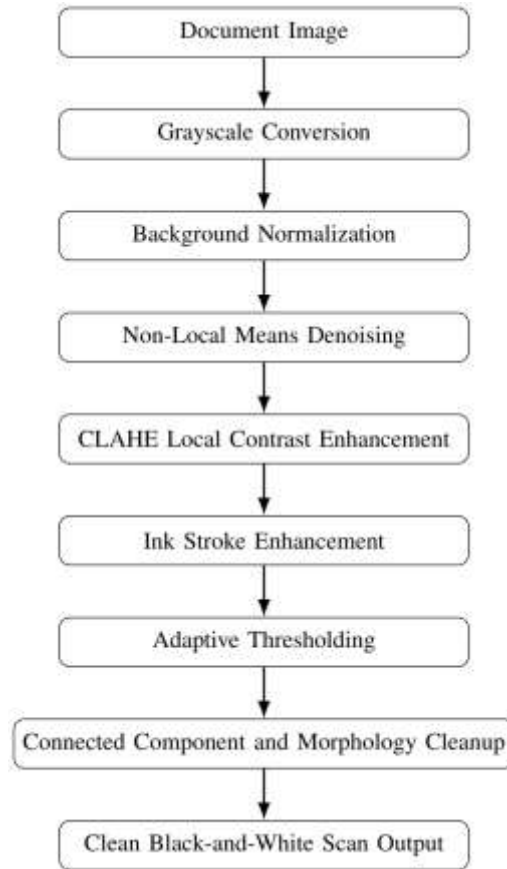


Fig. 4. Implementation pipeline of the Scan mode.

J. System Integration

Frontend and backend communication is performed using RESTful API calls. The Flutter frontend sends the selected image and enhancement parameters to the FastAPI backend. The backend processes the image and returns the enhanced output, static image path or encoded image data, mode information, and quality metrics. The frontend then displays the result and allows the user to interact with the output.

Table VII summarizes the integration between the main system components.

TABLE VII
FRONTEND AND BACKEND INTEGRATION SUMMARY.

Stage	Implementation Description
Image Upload	The user selects an image from the Flutter interface.
API Request	The image, selected mode, and strength parameter are sent to the FastAPI backend.
Image Processing	The backend extracts features and applies the appropriate enhancement pipeline.
Metric Calculation	The backend computes quality indicators for the original and enhanced images.
Response Generation	The backend returns the enhanced image and related metadata to the frontend.
Visualization	The Flutter frontend displays the original and enhanced images with comparison controls.
User Adjustment	The user modifies enhancement strength through the Human-in-the-Loop slider.

K. Experimental Testbed

The experimental testbed consisted of a Windows-based workstation running the backend and frontend development tools. The backend was executed locally using FastAPI and Python, while the frontend was tested using Flutter. The evaluation images were selected from real-world examples representing the main supported categories of the proposed system.

The test images included low-light photographs, product images, social media and portrait images, historical and scanned documents, handwritten pages, and general-purpose photographs. These categories were selected to evaluate the adaptability and practical effectiveness of the proposed framework across different enhancement scenarios.

Table VIII summarizes the image categories used in the experimental testbed.

TABLE VIII
IMAGE CATEGORIES USED IN THE EXPERIMENTAL TESTBED.

Image Category	Purpose of Evaluation
Low-light photographs	Evaluate brightness recovery, shadow enhancement, and noise control.
Product images	Evaluate product clarity, color preservation, and background realism.
Social and portrait images	Evaluate safe enhancement while preserving facial realism.
Historical and scanned doc-	Evaluate document readability, binarization, and cleanup performance.
ument pages	
Handwritten pages	Evaluate preservation of handwriting strokes and text visibility.
General-purpose photographs	Evaluate overall enhancement adaptability and natural appearance.

V. EVALUATION AND RESULTS

A. Experimental Setup

The proposed *Smart Visual Optimizer* framework was evaluated using 25 real-world images representing the five implemented enhancement modes. The dataset included product images, portrait and social media images, low-light photographs, scanned and historical document images, and general-purpose photographs. Each category contained five images to provide a balanced evaluation across the supported modes.

All experiments were performed on a Windows-based workstation using CPU-only execution. The backend was executed using Python, OpenCV, and FastAPI, while the frontend was tested using Flutter. The purpose of this setup was to evaluate the practicality of the system in a common development environment without requiring GPU acceleration.

Although some interface examples in this paper use the Low Light mode as a representative case, the experimental evaluation covered all five implemented enhancement modes: AI Super, Product, Social, Low Light, and Scan. Therefore, the reported results summarize the behavior of the complete multi-mode framework rather than a single enhancement case.

TABLE IX
EVALUATION DATASET DISTRIBUTION.

Category	Number of Images
Product	5
Social	5
Low Light	5
Document	5
General	5
Total	25

Table X summarizes the evaluation focus of each implemented enhancement mode.

TABLE X
EVALUATION COVERAGE ACROSS THE IMPLEMENTED ENHANCEMENT MODES.

Enhancement Mode	Evaluation Focus
AI Super	General visual enhancement and adaptive image optimization.
Product	Product clarity, color preservation, and realistic commercial appearance.
Social	Portrait enhancement while preserving facial realism, skin texture, and identity.
Low Light	Brightness recovery, shadow enhancement, contrast improvement, and noise control.
Scan	Document readability, binarization, background cleanup, and text visibility.

For each image, the original and enhanced outputs were analyzed using feature-based quality indicators. These indicators included brightness, contrast, sharpness, saturation, highlight distribution, and shadow distribution. For document images, visual readability and text clarity were also considered because the main goal of the Scan mode is to improve document readability

rather than only visual appearance.

B. Evaluation Metrics

Since the tested real-world images do not have ground-truth reference images, a feature-based no-reference evaluation strategy was used. This approach is suitable for practical enhancement systems where only the original and enhanced images are available. No-reference image quality assessment is commonly used when reference images are unavailable [13, 17].

The enhancement quality was measured using the following indicators:

- **Brightness:** measures the average intensity level of the image.
- **Contrast:** measures the variation of pixel intensities.
- **Sharpness:** estimates edge clarity using the variance of the Laplacian response.
- **Saturation:** measures color intensity.
- **Highlight distribution:** estimates the amount of very bright regions.
- **Shadow distribution:** estimates the amount of very dark regions.

A composite quality score was calculated using the weighted quality score defined in the methodology section. The percentage improvement was computed as:

$$Imp(\%) = \frac{Q_e - Q_o}{Q_o} \times 100 \quad (8)$$

where Q_o is the original quality score and Q_e is the enhanced quality score.

C. Quantitative Results

Table XI presents the average enhancement results for each image category. The results show that the proposed framework improved the average quality score across all tested categories.

TABLE XI
AVERAGE ENHANCEMENT RESULTS BY IMAGE CATEGORY.

Category	Original	Enhanced	Improvement
Product	33.4	35.4	6.0%
Social	39.3	40.2	2.3%
Low Light	20.2	33.5	65.8%
Document	38.2	47.8	25.1%
General	23.8	34.4	44.5%

Figure 5 compares the original and enhanced quality scores across the evaluated categories. The enhanced scores are higher than the original scores in all categories, indicating that the proposed framework improved the measured perceptual quality of the input images.

Figure 6 presents the percentage improvement achieved by the proposed framework. The highest improvement was observed in the Low Light category because the input images had weak illumination and low initial quality scores. The General and Document categories also achieved clear improvements due to contrast enhancement, detail improvement, and document-specific processing.

The Product and Social categories showed smaller numerical improvements because many images in these categories already had acceptable quality before enhancement. In these cases, the system intentionally applied conservative enhancement to preserve realistic product appearance, natural colors, skin texture, and facial identity.

D. Visual Interface Results

Figure 7 presents the main interface of the proposed *Smart Visual Optimizer* system. The figure combines two important interaction views: the before-after comparison interface and the Human-in-the-Loop control interface. The first view allows the user to inspect visual differences using an interactive slider, while the second view displays the enhancement strength control, AI recommendation, and quality metrics.

Although the visual example shown in the interface focuses on the Low Light mode, the same interaction design is used across the remaining modes, including AI Super, Product, Social, and Scan. This demonstrates that the proposed system is not limited to one enhancement case, but provides a unified interface for user-controlled multi-mode image enhancement.

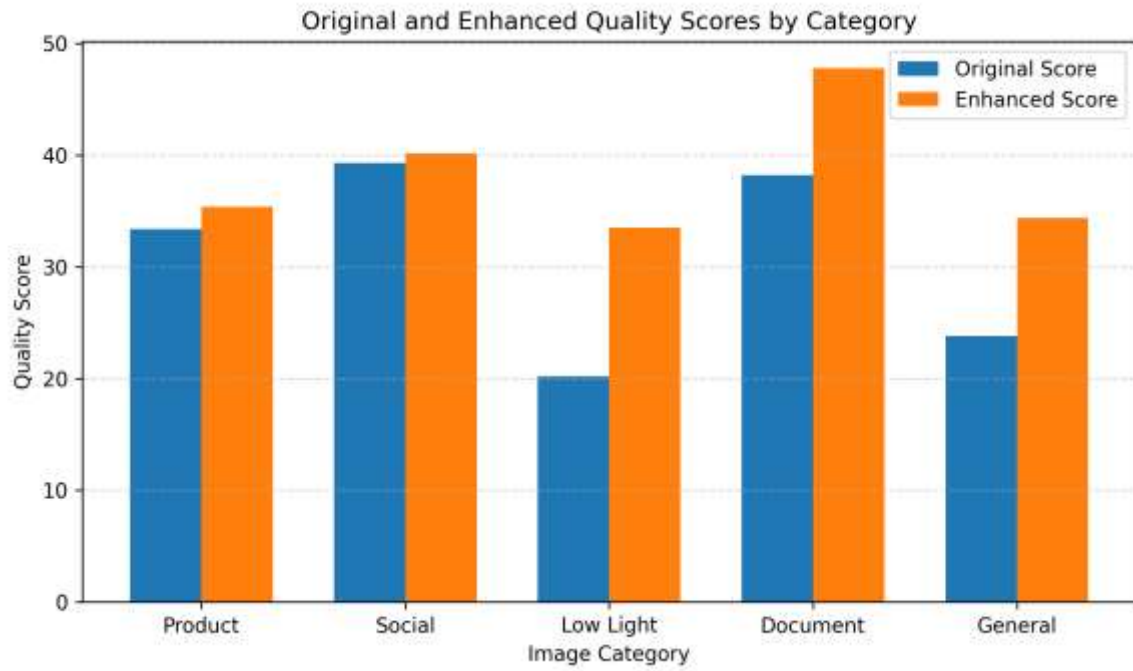


Fig. 5. Comparison between original and enhanced quality scores across image categories.

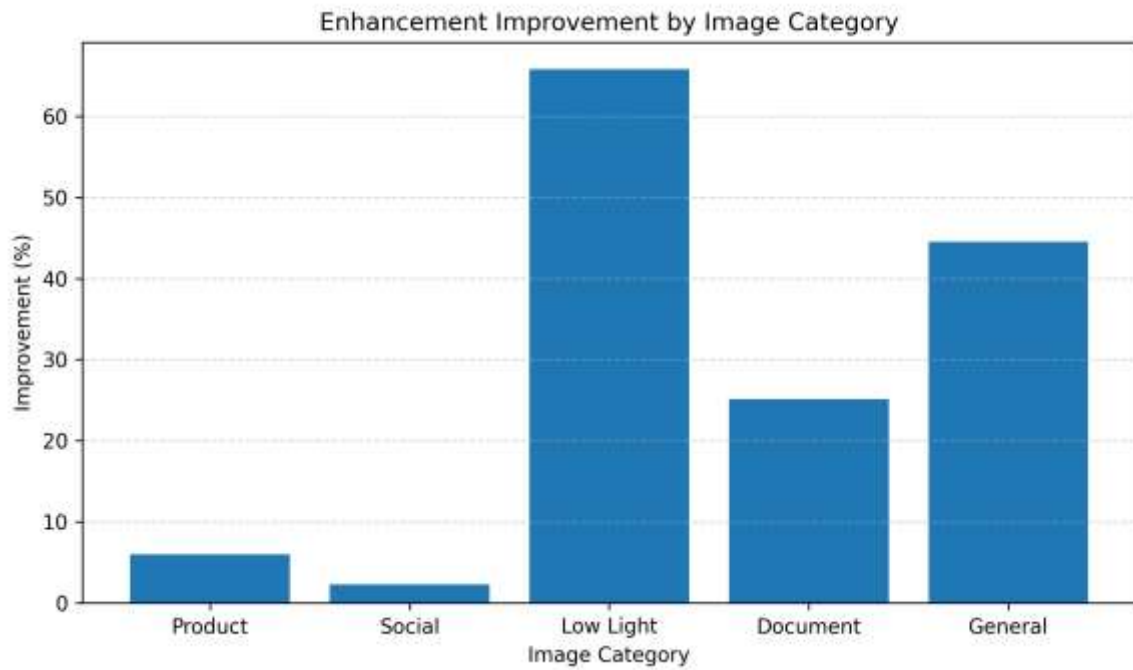


Fig. 6. Enhancement improvement percentage across image categories.

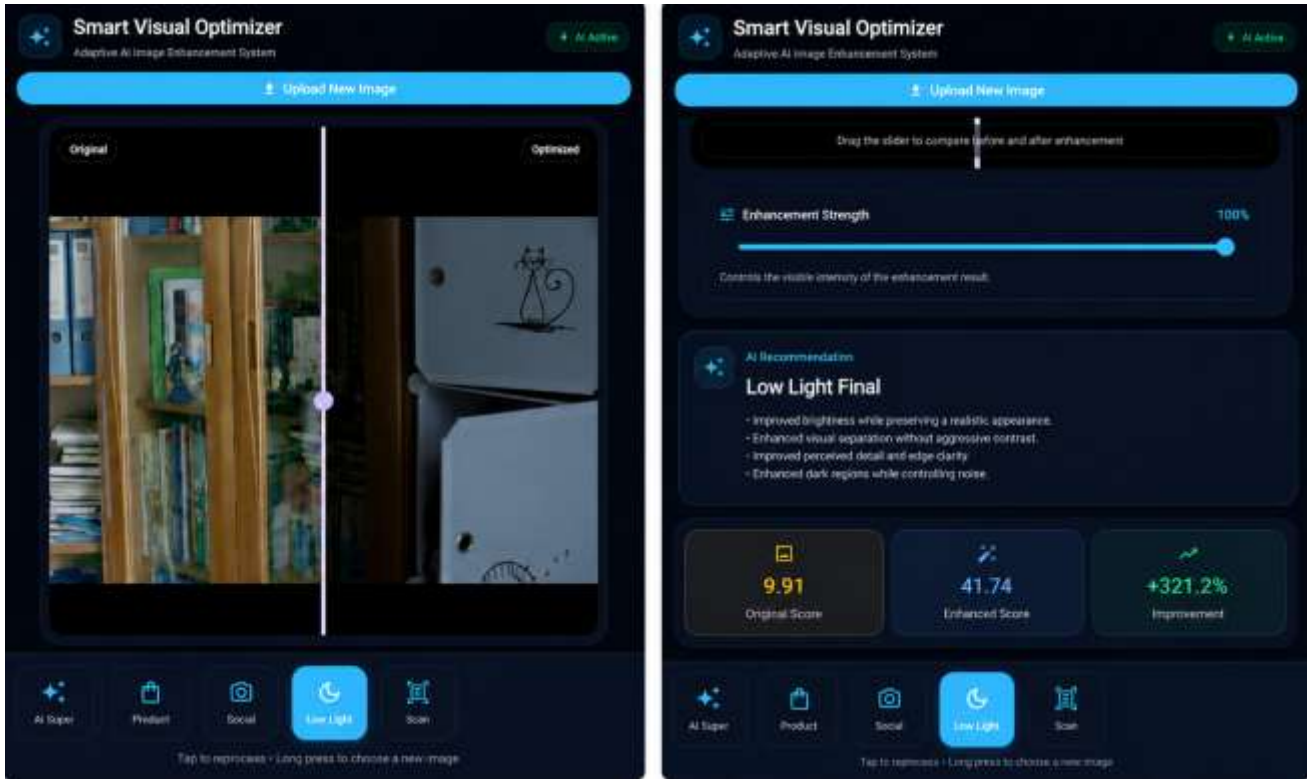


Fig. 7. Smart Visual Optimizer interface showing the before–after comparison slider, Human-in-the-Loop enhancement control, AI recommendation, and quality metrics.

E. Performance Across Enhancement Modes

The Product mode achieved stable results for commercial and e-commerce images by applying controlled contrast enhancement, white balance correction, mild sharpening, and highlight protection. The final implementation avoided aggressive AI super-resolution in this mode because practical testing showed that it may introduce artificial textures in product backgrounds.

The Social mode provided conservative enhancement for portrait and social media images. It improved visual appearance while avoiding strong facial alteration. This is important because face restoration models may alter identity-related details or over-smooth skin when applied aggressively [9].

The Low Light mode was effective for underexposed images by improving brightness and local contrast while applying denoising to reduce noise amplification. This behavior is consistent with low-light enhancement research, where illumination recovery should be balanced with noise control and structure preservation [5, 15, 16].

The Scan mode significantly improved document readability. It used grayscale conversion, background normalization, denoising, CLAHE, adaptive thresholding, connected component cleanup, and morphology-based text strengthening. These operations are related to document binarization and scan enhancement tasks [1–4, 11, 12].

The AI Super mode served as a general enhancement option. It preserved already clear images and applied stronger restoration only when the image characteristics indicated that additional enhancement was needed. Real-ESRGAN was treated as an optional restoration component for suitable low-detail images rather than a default stage for every input [8].

F. Human-Centered Evaluation

The proposed framework was also evaluated from a Human-Centered Interaction perspective. The system provides user control through mode selection, before–after comparison, displayed quality metrics, AI recommendation, and an enhancement strength slider.

The before–after comparison interface allows users to inspect enhancement effects directly. The strength slider allows users to adjust the final result according to personal preference. This helps reduce over-enhancement and supports a balance between automated processing and human judgment.

From an HCI perspective, the framework emphasizes three main principles:

- **User control:** users can select enhancement modes and adjust enhancement strength.
- **Visual feedback:** users can compare original and enhanced images directly.

- **Interactive decision support:** the system provides enhanced outputs, recommendations, and quality metrics while keeping the user involved in the final decision.

This design aligns with Human-Centered AI principles that emphasize reliability, transparency, and meaningful human control [14].

G. Informal User Feedback

In addition to the technical evaluation, informal user feedback was collected during practical demonstrations of the system. Several users interacted with the *Smart Visual Optimizer* interface and observed the difference between the original and enhanced images. The feedback was generally positive, especially regarding the clarity of the before–after comparison and the ability to control the enhancement strength.

Users reported that the comparison slider made it easier to notice the visual improvement, while the enhancement strength slider gave them a sense of control over the final result. This was important because users could reduce the enhancement intensity when they preferred a more natural appearance or increase it when stronger improvement was needed. Users also found the mode-based interface useful because it organized the enhancement process according to the image type.

Although this feedback was not collected through a formal large-scale usability study, it provides initial evidence that the proposed Human-in-the-Loop design improves user engagement and supports more transparent interaction with the image enhancement process. A more detailed user study with structured questionnaires and larger participant groups is planned for future work.

H. Discussion

The evaluation results show that *Smart Visual Optimizer* improves image quality across different image categories while preserving visual realism. The strongest numerical improvements appeared in Low Light, Document, and General images because these inputs had lower initial quality or readability problems.

The results also confirm the importance of using dedicated enhancement modes. Product and Social images require conservative processing to avoid unrealistic appearance, while document images require specialized scan enhancement focused on readability. Therefore, a single fixed enhancement pipeline would not be suitable for all categories.

The Human-in-the-Loop interface further improves the practical value of the system by allowing users to visually compare results, review quality indicators, and control enhancement strength. This makes the framework more transparent and user-centered than fully automatic black-box enhancement tools.

I. Limitations

Despite the promising results, several limitations remain. First, the evaluation dataset contains 25 real-world images, which is sufficient for initial testing but limited for large-scale validation. A larger and more diverse dataset would provide stronger evidence of generalization across different image types, lighting conditions, resolutions, and degradation levels.

Second, the quality score is based on handcrafted visual indicators such as brightness, contrast, sharpness, saturation, highlight distribution, and shadow distribution. Although these indicators are useful for practical evaluation and real-time feedback, they do not fully represent human visual perception. Therefore, future evaluation should include additional perceptual metrics and user-based assessment.

Third, CPU-only execution limits the practical use of heavy AI-based restoration models such as Real-ESRGAN. For this reason, the final system uses AI-assisted restoration selectively rather than applying it to all input images. GPU-based deployment could improve processing speed and support stronger restoration models in suitable cases.

Fourth, the Scan mode improves document readability through binarization, denoising, background cleanup, and morphology-based text enhancement, but it does not include OCR-based evaluation or automatic text recognition. Future work should evaluate the Scan mode using OCR accuracy, text recognition performance, and document readability metrics.

Finally, user feedback was collected informally during practical system demonstrations rather than through a large-scale structured usability study. Although the feedback was positive, especially regarding the before–after comparison and enhancement strength control, future work should include formal HCI evaluation using questionnaires, interviews, observation, and statistical analysis with a larger participant group.

J. Summary of Findings

The experimental evaluation demonstrates the following findings:

- The proposed framework improved the average quality score across all evaluated categories.
- The evaluation covered all five implemented enhancement modes: AI Super, Product, Social, Low Light, and Scan.
- The largest improvements were observed in Low Light, Document, and General images.
- Product and Social modes applied conservative enhancement to preserve realism and avoid artificial results.
- The Scan mode improved document readability through document-specific binarization and cleanup operations.

- Optional AI-assisted restoration is useful for selected low-detail images but should not be applied aggressively to all inputs.
- Human-in-the-Loop interaction improved transparency by allowing users to compare outputs, review recommendations, and control enhancement strength.

VI. CONCLUSION AND FUTURE WORK

This paper presented *Smart Visual Optimizer*, a Human-Centered Adaptive Image Enhancement Framework that combines scene-aware image analysis, mode-specific enhancement pipelines, feature-based quality evaluation, optional AI-assisted restoration, and Human-in-the-Loop interaction within a unified real-time system.

The proposed framework was designed to address the limitations of fixed image enhancement approaches by adapting the enhancement process according to image content and visual characteristics. Instead of applying a single enhancement strategy to all images, the system supports five dedicated modes: *AI Super*, *Product*, *Social*, *Low Light*, and *Scan*. Each mode applies a specific processing route according to the visual requirements of the input image. The implementation uses Python, OpenCV, FastAPI, and Flutter, while Real-ESRGAN is treated as an optional AI-assisted restoration component for selected low-detail images [8].

The experimental evaluation demonstrated that the proposed framework improved feature-based perceptual image quality across multiple real-world image categories, including low-light photographs, product images, portrait and social images, document scans, handwritten pages, and general-purpose photographs. The strongest improvements were observed in low-light, document, and general image categories, where the original images usually had lower brightness, weaker contrast, or reduced readability. Product and Social modes showed more conservative numerical improvements because the system intentionally avoids aggressive enhancement in order to preserve realistic product appearance, skin texture, and facial identity.

The Scan mode showed practical effectiveness for document enhancement by improving text visibility, reducing background noise, and producing clearer black-and-white scan-like outputs. This confirms that document images require a specialized processing route that differs from general image enhancement. Operations such as background normalization, adaptive thresholding, denoising, and morphology-based cleanup were important for improving readability in printed, handwritten, and old-paper inputs [1–4].

The Human-in-the-Loop design also contributed to the usability of the system. The Flutter interface allows users to upload images, select enhancement modes, compare original and enhanced outputs, inspect quality metrics, and adjust enhancement strength using an interactive slider. This gives users direct control over the final output and reduces the risk of over-enhancement. The design aligns with human-centered AI principles by combining automated processing with meaningful user control [14].

Despite the promising results, several limitations remain. First, the evaluation was performed on a moderate-sized real-world dataset, which limits the generalizability of the quantitative findings. Second, the quality assessment method relies mainly on handcrafted feature-based indicators such as brightness, contrast, and sharpness. Although these indicators are useful for practical analysis, they do not fully represent human visual perception. Third, AI-assisted restoration models such as Real-ESRGAN introduce higher processing latency under CPU-only execution, which limits their use in real-time scenarios without GPU acceleration. Finally, the Scan mode improves document readability, but it does not include OCR-based evaluation or automatic text recognition.

Future work will focus on expanding the evaluation dataset and comparing the proposed framework against standardized image enhancement and document enhancement benchmarks. Additional no-reference image quality assessment metrics such as BRISQUE and NIQE may be integrated to provide stronger perceptual evaluation. GPU acceleration can also be added to improve the execution speed of AI-based restoration models. For document images, future versions may include OCR-aware enhancement and readability-based evaluation to measure the effect of Scan mode on text recognition accuracy. Furthermore, user preference learning can be integrated to adapt enhancement strength and mode selection according to individual user behavior.

Overall, the proposed *Smart Visual Optimizer* framework demonstrates that combining adaptive image enhancement, feature-based quality analysis, optional AI-assisted restoration, and Human-in-the-Loop control can provide a practical and user-centered solution for improving diverse real-world images. The system offers a balanced approach between automation and user control, making it suitable for applications such as digital photography, social media content, e-commerce product presentation, low-light image correction, and document digitization.

REFERENCES

- [1] OpenCV, “Image thresholding,” https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html, accessed : 2026–06–17.
- [2] —, “Histogram equalization and clahe,” https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html, 2026, accessed : 2026–06–17.
- [3] —, “Image denoising,” https://docs.opencv.org/4.x/d5/d69/tutorial_py_noise_removal.html, 2026, accessed : 2026–06–17.
- [4] —, “Morphological transformations,” https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html, 2026, accessed : 2026–06–17.

- [5] Y. Cai, H. Bian, J. Lin, H. Wang, R. Timofte, and Y. Zhang, "Retinexformer: One-stage retinex-based transformer for low-light image enhancement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [6] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5718–5729.
- [7] Y. Ai, H. Huang, X. Zhou, J. Wang, and R. He, "Multimodal prompt perceiver: Empower adaptiveness, generalizability and fidelity for all-in-one image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 25 432–25 444.
- [8] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-esrgan: Training real-world blind super-resolution with pure synthetic data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 1905–1914.
- [9] X. Wang, Y. Li, H. Zhang, and Y. Shan, "Towards real-world blind face restoration with generative facial prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9168–9178.
- [10] K. Ma, Z. Shu, X. Bai, J. Wang, and D. Samaras, "Docunet: Document image unwarping via a stacked u-net," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4700–4709.
- [11] I. Pratikakis, K. Zagoris, X. Karagiannis, L. Tsochatzidis, T. Mondal, and I. Marthot-Santaniello, "Icdar 2019 competition on document image binarization (dibco 2019)," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 1547–1556.
- [12] U.-V. Marti and H. Bunke, "The iam-database: An english sentence database for off-line handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, pp. 39–46, 2002.
- [13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [14] B. Shneiderman, "Human-centered artificial intelligence: Reliable, safe and trustworthy," *International Journal of Human-Computer Interaction*, vol. 36, no. 6, pp. 495–504, 2020.
- [15] N. Zheng, M. Zhou, Y. Dong, X. Rui, J. Huang, C. Li, and F. Zhao, "Empowering low-light image enhancer through customized learnable priors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 12 525–12 535.
- [16] C. Tian, Y. Xu, and W. Zuo, "A survey of deep learning-based low-light image enhancement," *Sensors*, vol. 23, no. 7, p. 3567, 2023.
- [17] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.