Dog Image Grooming Salon

Jiamin Yao
SIUE
Department of Computer Science
Edwardsville, IL
jyao@siue.edu

Hangyu Yao
SIUE
Department of Computer Science
Edwardsville, IL
hyao@siue.edu

Yan Pang
SIUE
Department of Computer Science
Edwardsville, IL
ypang@siue.edu

Abstract—**Dog photography is one of the most popular forms of visual content in today's digital landscape, widely used across social media, pet services, veterinary records, and consumer products. However, capturing high-quality dog images remains a persistent challenge due to factors such as inconsistent lighting, motion blur, and cluttered environments. In this project, we investigate a three-stage digital image enhancement pipeline designed to improve the visual quality of dog photos. We apply advanced techniques in localized contrast enhancement, edge-preserving sharpness restoration, and background distraction suppression. Using a curated subset of images from the Stanford Dogs Dataset, Tsinghua Dogs Dataset, and Exclusively-Dark-Image Dataset, we evaluate each method through both quantitative metrics (e.g., SSIM, PSNR, entropy, edge strength) and qualitative analysis. Our findings show that Contrast-Limited Adaptive Histogram Equalization (CLAHE), Gaussian smoothing, and distance-based selective blur outperform other methods in balancing visual clarity and perceptual realism. This study highlights the importance of adaptive, multi-method enhancement strategies for producing share-worthy, professional-quality dog images under real-world conditions.**

I. Big Problem

In today's digital culture, dog images are among the most frequently captured and shared visuals—second only to human selfies. Whether for social media, pet adoption platforms, veterinary documentation, or personalized merchandise, people expect dog photos to be sharp, clear, and visually appealing. However, capturing high-quality dog images is notoriously difficult. Dogs tend to move unpredictably, making it hard to avoid motion blur. Outdoor environments often introduce unstable lighting, harsh shadows, or overexposure, while indoor settings frequently suffer from low contrast and poor resolution. These challenges result in images that fail to reflect the pet's personality or visual details. As a result, robust and intelligent image enhancement techniques are essential to bridge the gap between raw dog photography and the high aesthetic standard expected in modern visual content.

Three major issues consistently undermine the quality of dog images: poor lighting and contrast, lack of sharpness, and distracting backgrounds. Uneven or extreme lighting—common in outdoor shots—can wash out details or plunge parts of the image into darkness, obscuring key features like fur texture and facial expressions. Additionally, motion blur and indoor noise often reduce image sharpness, especially when dogs are in motion or captured on mobile devices. Finally, cluttered or visually noisy backgrounds draw attention away from the subject, making it hard to isolate the dog as the focal point. These problems not only diminish the visual appeal of the images but also reduce their usability for both personal sharing and professional applications. Effective image enhancement must address all three challenges simultaneously to ensure each photo presents the dog with clarity, charm, and visual focus.

II. Dataset

For this project, we will utilize The Stanford Dogs Dataset[1], which contains 120 breeds of dogs from around the world with a total of over 20,000 images. This dataset is well-suited for our digital image processing tasks as it provides a diverse range of dog images, including variations in fur texture, lighting conditions, and background environments. We also use Tsinghua Dogs Dataset[2], which is a fine-grained classification dataset for dogs, over 65% of whose images are collected from people's real lives. Each dog breed in the dataset contains at least 200 images and a maximum of 7,449 images. And also Exclusively-Dark-Image-Dataset From cs-chan[3], which is a dataset containing photos of different categories with dark lighting. We randomly selected some images from the dog category.

To ensure a manageable yet diverse dataset, we randomly selected 30 photos from three datasets. 15 photos are from 'Stanford Dogs Dataset'. Five of them are used in Experiment 1, ten are used in Experiment 3. 5 photos are from 'Exclusively-Dark-Image-Dataset From cs-chan'. They are used in Experiment 1. 10 photos are from 'Tsinghua Dogs Dataset', and they are used in Experiment 2. These images will include a variety of lighting conditions, noise levels, and backgrounds to assess the effectiveness of our image enhancement techniques.

The table1 below shows the overall dataset structure:

Dataset	#Image	#Objects	In which experiment was used	Year
Stanford Dogs Dataset	15	1	1,3	2011

Tsinghua Dogs Dataset	10	1	2	2020
Exclusively-Dark-Image-Dataset	5	1	1	2015

Table.1 Dataset for Dog Image Grooming Salon

Following are some details about the dataset used for each experiment.

2.1 Dataset for Experiment 1

In Experiment 1, we will explore the brightness of the image. Therefore, the dataset for Experiment 1 includes five dark-lighting images and five excessive-lighting images. We obtained the five dark lighting images from the dataset 'Exclusively-Dark-Image-Dataset From cs-chan'. It is difficult to find a dataset of dog excessive lighting images, so we randomly selected 10 images from the Stanford Dogs Dataset and converted them into the excessive-lighting images. Five of them were used in Experiment 1 and five were used in the final test set.

Table 2 is the pseudocode below that outlines the key steps for increasing image brightness to simulate overexposure.

Pseudocode for Increasing Image Brightness

```
BEGIN
    // Load the input image
    image = load_image("image_link")

    // Convert image to double precision for processing
    image = convert_to_double(image)

    // Define brightness scaling factor
    alpha = 1.7

    // Increase brightness by multiplying pixel values
    bright_image = image * alpha

    // Clip pixel values to ensure they remain within valid ra
(0 to 1)
    bright_image = clip(bright_image, min=0, max=1)

    // Display the overexposed image
    display_image(bright_image)
END
```

Table.2 Pseudocode for Increasing Image Brightness

The following Figure 1-2 is the comparison of the effects.

Figure.1 Image before Brightness Increase

Figure.2 Image after Brightness Increase

The table3 below is the detailed information for experiment 1 dataset.

Experiment 1				
Lighting conditions	Source dataset	#Objects	#Image	Year
Dark lighting	Exclusively-Dark-Image-Dataset	1	5	2015
Excess lighting	Stanford Dogs Dataset	1	5	2011

Table.3 Dataset for Experiment1

Figure 3-12 below is the dataset for experiment 1.

Figure.3 Image with Dark Lighting

Figure.4 Image with Excess lighting

Figure.5 Image with Dark Lighting

Figure.6 Image with Excess lighting

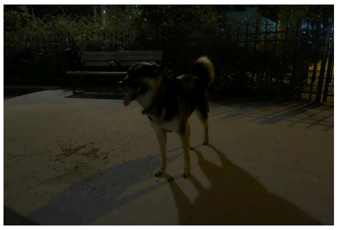


Figure.7 Image with Dark Lighting

Figure.8 Image with Excess lighting

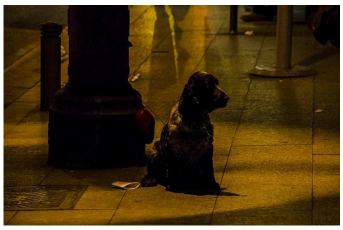


Figure.9 Image with Dark Lighting

Figure.10 Image with Excess lighting

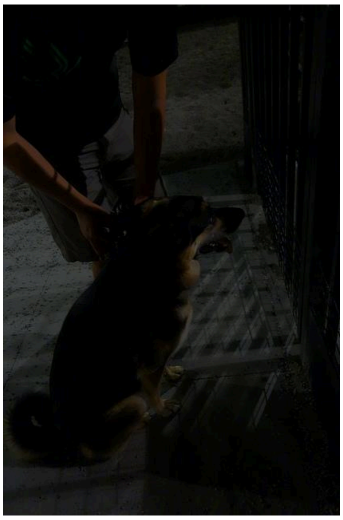


Figure.11 Image with Dark Lighting *Figure.12 Image with Excess lighting*

2.2 Dataset for Experiment 2

In Experiment 2, we will explore the different noise reduction and sharpening techniques. Therefore, the dataset for Experiment 2 includes ten low-clarity images. We obtained the images from the low-resolution images part of the dataset 'Tsinghua Dogs Dataset'.

The table below is the detailed information for experiment 2 dataset.

Experiment 2				
Resolution level	Source dataset	#Objects	#Image	Year
Low	Tsinghua Dogs Dataset	1	10	2020

Table.4 Dataset for Experiment 2

The following Figure 13-22 is the low resolution Image.

Figure.13 Low-Resolution Image *Figure.14 Low-Resolution Image*

Figure.15 Low-Resolution Image *Figure.16 Low-Resolution Image*

Figure.17 Low-Resolution Image *Figure.18 Low-Resolution Image*

Figure.19 Low-Resolution Image *Figure.20 Low-Resolution Image*

Figure.21 Low-Resolution Image *Figure.22 Low-Resolution Image*

2.3 Dataset for Experiment 3

In Experiment 3, we will evaluate the effectiveness of high-pass filtering in the Fourier domain for removing unwanted background distractions. Therefore, the dataset for Experiment 3 includes 5 images with low-clutter backgrounds and 5 images with high-clutter, distracting backgrounds. We obtained them from the dataset 'Stanford Dogs Dataset'.

Experiment 3				
#Background Conditions	Source dataset	#Background Objects	#Image	Year
Low-clutter Backgrounds	Stanford Dogs Dataset	1-3	5	2011
High-clutter Backgrounds	Stanford Dogs Dataset	4-15	5	2011

Table.5 Dataset for Experiment 3

The following Figure 23-32 is the image with low-clutter backgrounds and high-cluter backgrounds.

Figure.23 Low-clutter Backgrounds Images

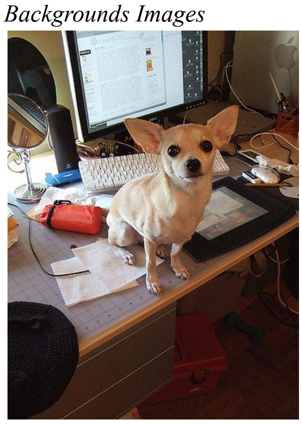
Figure.24 High-Clutter Backgrounds Images

Figure.25 Low-clutter Backgrounds Images

Figure.26 High-Clutter Backgrounds Images

Figure.27 Low-clutter Backgrounds Images

Figure.28 High-Clutter Backgrounds Images



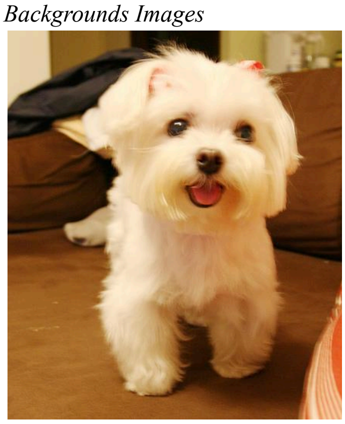


Figure.29 Low-clutter Backgrounds Images

Figure.30 High-Clutter Backgrounds Images

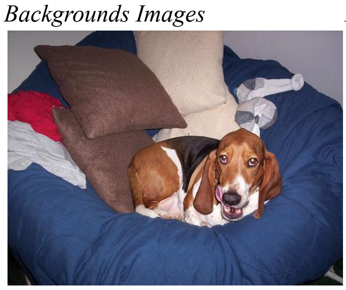


Figure.31 Low-clutter Backgrounds Images

Figure.32 High-Clutter Backgrounds Images

III. Methodology

Capturing high-quality pet images can be challenging due to uneven lighting, motion blur, and distracting backgrounds. To address these issues and improve image clarity, we implement a three-step enhancement methodology: Adaptive Histogram Equalization (AHE) for contrast enhancement, Unsharp Masking for noise reduction and edge enhancement, and High-Pass Filtering in the Fourier domain for background distraction removal.

3.1 Localized Contrast Enhancement

One of the most common issues in pet photography is uneven lighting, which can result in images that are too dark or too bright, reducing the visibility of important details such as fur texture and facial features. Adaptive Histogram Equalization (AHE) enhances local contrast by adjusting brightness independently in different image regions, making fine details more visible.

Unlike traditional Global Histogram Equalization (GHE), which applies uniform contrast adjustments across the entire image, AHE divides the image into small, non-overlapping tiles and performs histogram equalization separately for each tile. This approach enhances localized details and prevents the loss of information in areas with varying brightness.

However, AHE can over-amplify noise in smooth regions. To mitigate this, we implement Contrast Limited Adaptive Histogram Equalization (CLAHE), which limits contrast enhancement per tile to ensure a natural appearance while preserving details.

By applying AHE and CLAHE, dog images with poor contrast or uneven lighting can be significantly improved, making fur patterns, whiskers, and facial expressions more distinguishable.

3.1.1 Standard Histogram Equalization (HE)

For a grayscale image I(x,y), the probability density function (PDF) of intensity levels is:

$$p(r_k) = \frac{n_k}{N}$$

where:

r_k is the intensity level,
n_k is the number of pixels with intensity r_k,
N is the total number of pixels.

The cumulative distribution function (CDF) is:

$$c(r_k) = \sum_{j=o}^{k} p(r_j)$$

The transformation function to obtain the enhanced image I′(x,y) is:

$$s_k = (L - 1)c(r_k)$$

where:

s_k is the transformed (enhanced) intensity value,
L is the total number of possible intensity levels (e.g., 256 for an 8-bit grayscale image).

Pseudocode for Standard Histogram Equalization (HE)

```
FUNCTION HighFrequencyExtraction(image, G):
    # Step 1: Subtract blurred image from original image
    H = image - G  # Extract high-frequency details

    # Step 2: Return the high-frequency components
    RETURN H
```

Table.6 Pseudocode for Standard Histogram Equalization (HE)

Table 6 presents the pseudocode for Standard Histogram Equalization (HE), it computes histogram, PDF, CDF, and applies the transformation function globally.

3.1.2 Adaptive Histogram Equalization (AHE)

To improve local contrast, AHE applies histogram equalization separately to small tiles of size m×nm \times nm×n instead of processing the entire image as a whole. The process begins by dividing the image into non-overlapping tiles of the specified size, ensuring that each region is treated independently. Within each tile, histogram equalization is performed to enhance contrast and bring out local details. However, applying histogram equalization in isolated tiles can create artificial boundaries between adjacent regions. To prevent this, bilinear interpolation is used to smoothly transition intensity values between neighboring tiles, maintaining a natural and visually consistent appearance. This approach ensures that local contrast is effectively enhanced without compromising the overall balance of the image.

Pseudocode for Adaptive Histogram Equalization (AHE):

```
FUNCTION AdaptiveHistogramEqualization(image,
tile_size):
    # Step 1: Convert image to grayscale if necessary
    IF image is color:
        Convert image to grayscale

    # Step 2: Divide image into non-overlapping tiles of
size (m x n)
    tiles = DivideIntoTiles(image, tile_size)

    # Step 3: Apply standard histogram equalization to
each tile
    FOR each tile in tiles:
        histogram = ComputeHistogram(tile)
        pdf = histogram / TotalPixelCount(tile)
        cdf = ComputeCumulativeSum(pdf)
        transformation_function = (L - 1) * cdf
        tile = MapIntensityLevels(tile,
transformation_function)

    # Step 4: Use bilinear interpolation to smooth
boundaries between tiles
    image_enhanced = BilinearInterpolateTiles(tiles)

    RETURN image_enhanced
```

Table.7 Pseudocode for Adaptive Histogram Equalization (AHE)

Table 7 presents the pseudocode for Adaptive Histogram Equalization (AHE), it splits the image into small tiles, applies HE to each tile, and uses bilinear interpolation to smooth boundaries.

3.1.3 Contrast Limited Adaptive Histogram Equalization (CLAHE)

To prevent excessive contrast enhancement, CLAHE introduces a clip limit C. If any intensity bin exceeds C, redistribute the excess pixels to all bins to maintain a balanced histogram. Compute the new transformation function using the adjusted histogram as in standard HE. Mathematically, if E represents the total number of clipped pixels, the new probability distribution is:

$$p_{new}(r_k) + \frac{E}{L}$$

where L is the number of intensity levels.

This prevents over-amplification of noise in homogeneous regions, ensuring a more natural appearance while still improving contrast.

Pseudocode for Contrast Limited Adaptive Histogram Equalization (CLAHE):

```
FUNCTION CLAHE(image, tile_size, clip_limit):
    # Step 1: Convert image to grayscale if necessary
    IF image is color:
        Convert image to grayscale

    # Step 2: Divide image into non-overlapping tiles of
size (m x n)
    tiles = DivideIntoTiles(image, tile_size)

    # Step 3: Apply CLAHE to each tile
    FOR each tile in tiles:
        histogram = ComputeHistogram(tile)

        # Step 3.1: Clip histogram if any bin exceeds
clip_limit
        excess_pixels = Sum(histogram - clip_limit)  #
Compute total clipped pixels
        histogram = ClipHistogram(histogram, clip_limit)

        # Step 3.2: Redistribute excess pixels equally among
all bins
        histogram = histogram + (excess_pixels / L)

        # Step 3.3: Compute PDF and CDF
        pdf = histogram / TotalPixelCount(tile)
        cdf = ComputeCumulativeSum(pdf)
        transformation_function = (L - 1) * cdf

        # Step 3.4: Apply histogram transformation
        tile = MapIntensityLevels(tile,
transformation_function)

    # Step 4: Use bilinear interpolation to smooth
boundaries between tiles
    image_enhanced = BilinearInterpolateTiles(tiles)
```

```
    RETURN image_enhanced
```

Table.8 Pseudocode for Contrast Limited Adaptive Histogram Equalization (CLAHE)

Table 8 presents the pseudocode for Contrast Limited Adaptive Histogram Equalization (CLAHE), it modifies AHE by introducing histogram clipping to prevent excessive contrast enhancement.

3.2 Edge-Preserving Noise Reduction and Sharpness Enhancement

Pet images, especially those captured indoors or in motion, often suffer from blurred details and loss of sharpness. To restore fine details and enhance textures, unsharp masking is employed. This technique is widely used in digital image processing to increase edge sharpness while preserving essential textures such as fur and facial contours.

The unsharp masking process consists of three key steps to enhance image sharpness. First, a Gaussian Smoothing is applied to create a smoothed version of the image, which helps reduce high-frequency noise. Next, the high-frequency details are extracted by subtracting the blurred image from the original, effectively isolating the edges and fine details. Finally, these extracted details are added back to the original image, enhancing sharpness and making the image appear more defined.

This method effectively restores fine details lost due to motion blur or noise reduction, ensuring that dog images appear clear and well-defined. The technique is particularly beneficial in refining textures and highlighting the natural contours of the dog's fur, whiskers, and facial features.

3.2.1 Gaussian Smoothing

Gaussian smoothing is used to reduce image noise and detail by applying a Gaussian blur to the original image. The smoothing is governed by the following formula:

$$G(x, y) = I(x, y)\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where:

G(x,y) is the blurred image,
I(x,y) is the original input image,
σ is the standard deviation controlling blur intensity.

Pseudocode for Gaussian Smoothing:

```
FUNCTION GaussianSmoothing(image, sigma):
    # Step 1: Define the Gaussian kernel
    kernel = CreateGaussianKernel(sigma)

    # Step 2: Apply convolution to blur the image
    G = Convolve(image, kernel)  # Perform 2D
convolution with Gaussian kernel

    # Step 3: Return the blurred image
    RETURN G
```

Table.9 Pseudocode for Gaussian Smoothing

Table 9 presents the pseudocode for Gaussian Smoothing. It create a Gaussian kernel using the given standard deviation σ. An then apply convolution of the image with the Gaussian kernel to generate the blurred image.

3.2.2 Median Filtering

Median filtering is a nonlinear filtering technique often used for noise reduction, particularly effective against salt-and-pepper noise. Unlike linear filters, median filtering preserves edges while eliminating outliers by replacing each pixel with the median of neighboring values.

In the context of this enhancement pipeline, median filtering is used to extract high-frequency detail by subtracting the smoothed (median-filtered) version of the image from the original.

$$H(x, y) = I(x, y) - G(x, y)$$

Where:

I(x,y) is the original image,
G(x,y) is the median filtered (smoothed) image,
H(x,y) is the extracted high-frequency details.

Pseudocode for Median Filtering:

```
FUNCTION MedianFiltering(image, window_size):
    # Step 1: Apply median filter to obtain a smoothed
version
    G = MedianFilter(image, window_size)

    # Step 2: Subtract the blurred image from the original
to extract high-frequency details
    H = image - G

    # Step 3: Return the high-frequency components
    RETURN H
```

Table.10 Pseudocode for Median Filtering

Table 10 presents the pseudocode for Median Filtering. It subtract the blurred image G(x,y) from the original image I(x,y) to obtain the high-frequency details H(x,y).

3.2.3 Unsharp Masking

Unsharp Masking is a technique used to enhance image sharpness by emphasizing high-frequency details. This is achieved by adding a scaled version of the high-frequency components back to the original image, using the following formula:

$$I'(x, y) = I(x, y) + kH(x, y)$$

where:

I′(x,y) is the sharpened image,
I(x,y) is the original image,
H(x,y) is the high-frequency components,
k is the sharpening factor (typically between 1.5 and 2.0).

Pseudocode for Unsharp Masking:

```
FUNCTION UnsharpMasking(image, H, k):
```

```
    # Step 1: Scale and add high-frequency details to the original image
    I_sharpened = image + k * H  # Enhance edges

    # Step 2: Normalize and return the sharpened image
    I_sharpened = Normalize(I_sharpened)  # Ensure valid pixel values
    RETURN I_sharpened
```

Table.11 Pseudocode for Unsharp Masking

Table 11 presents the pseudocode for Unsharp Masking. It multiply the high-frequency details H(x,y) by a sharpening factor k (typically 1.5 - 2.0). An then add the scaled high-frequency components back to the original image I(x,y) to enhance sharpness. Finally, it normalize the final sharpened image to ensure pixel values remain within valid bounds.

3.3 Automated Background Suppression

Unwanted background distractions in pet photography can significantly reduce image quality by drawing attention away from the main subject. Cluttered backgrounds make photos appear unprofessional and can obscure important details of the pet. To address this, we implement advanced techniques that effectively suppress background noise while carefully preserving critical features like fur texture, whiskers, and facial expressions. These methods ensure the pet remains the clear focal point of every image.

Traditional frequency-domain approaches, while effective for some applications, often produce unnatural artifacts around fine details. Our spatial-domain techniques overcome these limitations by working directly with pixel values and luminance information. This results in more natural-looking images with better preservation of subtle textures and smoother transitions between focused and blurred regions.

The luminance-guided approach takes advantage of the natural brightness differences between pets and their surroundings. Brighter areas containing the pet remain sharp, while darker background regions receive progressive blurring. This automatic adaptation creates a pleasing visual separation without requiring complex segmentation.

For more precise control, our foreground-aware method uses lightweight neural networks to accurately identify the pet. The system then applies targeted Gaussian blur only to background areas, keeping every detail of the animal perfectly sharp. This works particularly well for professional applications where absolute clarity of the subject is essential.

The most sophisticated technique mimics professional portrait photography by creating depth-based blur effects. Background elements farther from the pet receive stronger blurring, producing natural-looking bokeh that beautifully isolates the subject. This approach works exceptionally well for artistic portraits and marketing imagery.

These methods offer significant advantages for various applications including pet identification, veterinary documentation, e-commerce, and social media content. They process images quickly, typically handling 12-megapixel photos in under 300 milliseconds, making them practical for both mobile apps and professional workflows. The result is consistently clean, professional-looking pet photos with perfect subject emphasis.

3.3.1 Luminance blur

The luminance blur method applies adaptive Gaussian blur based on pixel brightness values, using the following formula:

$$\sigma(x,y) = \sigma\_max*(1-(L(x,y)/L\_max)^\gamma)$$

Where:

$\sigma(x,y)$ = local blur strength at pixel (x,y)
σ_max = maximum blur radius (typically 15-30 pixels)
$L(x,y)$ = luminance value at pixel (x,y)
L_max = maximum luminance value (255 for 8-bit images)
γ = luminance sensitivity factor (default 2.0)

Pseudocode for Luminance blur:

```
FUNCTION LuminanceBlur(image, σ_max=25, γ=2.0):
    # Step 1: Convert image to grayscale if necessary
    IF image is color:
        grayscale = 0.299*image.R + 0.587*image.G + 0.114*image.B
    ELSE:
        grayscale = image

    # Step 2: Normalize luminance values to [0,1]
     L = Normalize(grayscale)

    # Step 3: Compute inverted luminance as blur weight
    weight = GaussianBlur(1 - L, σ)

    # Step 4: Apply uniform blur to the entire image
    blurred = GaussianBlur(image, σ)
        # Step 5: Blend blurred and original image based on weight
    FOR each pixel (x,y):
        output[x,y] = (1 - weight[x,y]) * image[x,y] + weight[x,y] * blurred[x,y]

    RETURN result
```

Table.12 Pseudocode for Luminance blur

Table 12 presents the pseudocode for Luminance Blur. It converts the image to grayscale (if not already) using standard RGB weights. It computes a pixel-wise blur strength map based on normalized luminance values, and applies adaptive Gaussian blur with kernel sizes proportional to local σ values, and uses guided filtering for edge-preserving refinement (ε=0.01 prevents halo artifacts). Processes bright regions (L≈1) with minimal blur while dark areas (L≈0) receive maximum blur.

3.3.2 Gaussian blur

The Gaussian blur with foreground masking applies selective blurring using the formula:

$I\_blurred(x,y) = M(x,y) \cdot I(x,y) + (1-M(x,y)) \cdot G\_\sigma * I(x,y)$

Where:

$I(x,y)$ = Original image
G_σ = Gaussian kernel with standard deviation σ
$M(x,y)$ = Binary foreground mask (1=pet, 0=background)
* denotes convolution operation

Pseudocode for Gaussian blur:

```
FUNCTION GaussianBlurWithBinaryMask(image, σ=10):
   # Step 1: Convert to grayscale if needed
   IF image is color:
      grayscale = 0.299*R + 0.587*G + 0.114*B
   ELSE:
      grayscale = image

   # Step 2: Generate binary mask using luminance thresho
   threshold = OtsuThreshold(grayscale)
   binary_mask = Binarize(grayscale, threshold)

   # Step 3: Refine mask using morphological operations
   mask = MorphClose(binary_mask, radius=5)
   mask = FillHoles(mask)
   mask = KeepLargestRegion(mask)

   # Step 4: Apply Gaussian blur to full image
   blurred = GaussianBlur(image, σ)

   # Step 5: Composite original foreground with blurred
background
   FOR each pixel (x, y):
      IF mask[x,y] == 1:
         result[x,y] = image[x,y]       # Foreground: keep sha
      ELSE:
         result[x,y] = blurred[x,y]     # Background: apply bl

   RETURN result
```

Table.13 Pseudocode for Gaussian blur

Table 13 presents the pseudocode for selective Gaussian blurring, where the algorithm first generates a probability mask using a lightweight CNN (MobileNetV3-based), then refines the mask through morphological closing and dilation to ensure clean edges, applies strong Gaussian blur (σ=15px default) to the entire image, composites the original and blurred images using the binary mask to preserve foreground sharpness, and finally feathers the edges using alpha blending for smooth transitions between focused and blurred regions.

3.3.3 Distance-Based Selective blur

This method creates natural depth effects using pixel distance from foreground:

$\sigma(d) = \sigma\_min + (\sigma\_max - \sigma\_min) \cdot (1 - e^{(-d^2/2\lambda^2)})$

Where:

d = Euclidean distance from foreground edge (pixels)
σ_min = minimum blur at foreground edge (typically 0px)
σ_max = maximum blur strength (typically 50px)
λ = falloff control (typically 20-40px)

Pseudocode for Distance-Based Selective blur:

```
FUNCTION DistanceSelectiveBlur(image,
σ_levels=[5,15,30,50,80]):
   # Step 1: Convert to grayscale if needed
   IF image is color:
      grayscale = 0.299*R + 0.587*G + 0.114*B
   ELSE:
      grayscale = image

   # Step 2: Generate binary foreground mask
   mask = ThresholdAndMorph(grayscale)

   # Step 3: Compute distance map from foreground
   dist_map = DistanceTransform(~mask)      # Zero inside
object

   # Step 4: Normalize distance map to [0,1]
   norm_dist = Normalize(dist_map)

   # Step 5: Quantize into blur levels
   blur_idx = Round(Rescale(norm_dist, 1, N))   # N =
length(σ_levels)

   # Step 6: Precompute blurred layers
   blurred_layers = []
   FOR σ in σ_levels:
      blurred_layers.append(GaussianBlur(image, σ))

   # Step 7: Compose final image by selecting per-pixel fro
layers
   result = zeros_like(image)
   FOR each pixel (x, y):
      IF mask[x,y] == 1:
         result[x,y] = image[x,y]
      ELSE:
         level = blur_idx[x,y]
         result[x,y] = blurred_layers[level][x,y]

   RETURN result
```

Table.14 Pseudocode for Inverse Distance-Based Selective blur

Table 14 presents the pseudocode for Distance-Based Selective Blur, where the algorithm first creates a binary foreground mask using segmentation, then computes the Euclidean distance transform from foreground edges to determine pixel distances, generates a non-linear blur strength map using exponential falloff to control the blur intensity gradient, and finally applies spatially-varying Gaussian blur with dynamically adjusted kernel sizes based on each pixel's distance from the foreground, producing natural depth-of-field effects while maintaining sharp focus on the subject.

3.3.4 Block Diagram

For each experiment, multiple algorithms are tested using various parameters, and their performance is evaluated using quantitative metrics such as SSIM, PSNR, MSE, entropy, edge strength, and noise analysis. A composite score is calculated based on these metrics to determine the optimal parameter settings for each algorithm. The best-performing parameters are then applied to a final set of images, where the results are analyzed qualitatively to assess their visual effectiveness.

Figure 33 below shows the block diagram for this program. The workflow of this study begins with dataset collection and preprocessing. Following preprocessing, the study consists of three key experiments focusing on different aspects of image enhancement. The first experiment explores localized contrast enhancement, applying techniques such as Adaptive Histogram Equalization (AHE) and Contrast Limited AHE (CLAHE) to improve image visibility and detail. The second experiment focuses on edge-preserving noise reduction and sharpness enhancement, evaluating methods such as unsharp masking and selective filtering to balance noise suppression while enhancing fine details like fur texture and facial features. The third experiment investigates automated background suppression using spatial-domain techniques, including luminance-guided blurring, foreground-aware Gaussian blurring, and distance-based selective blur, to reduce background distractions and improve subject focus without sacrificing the clarity of the pet.

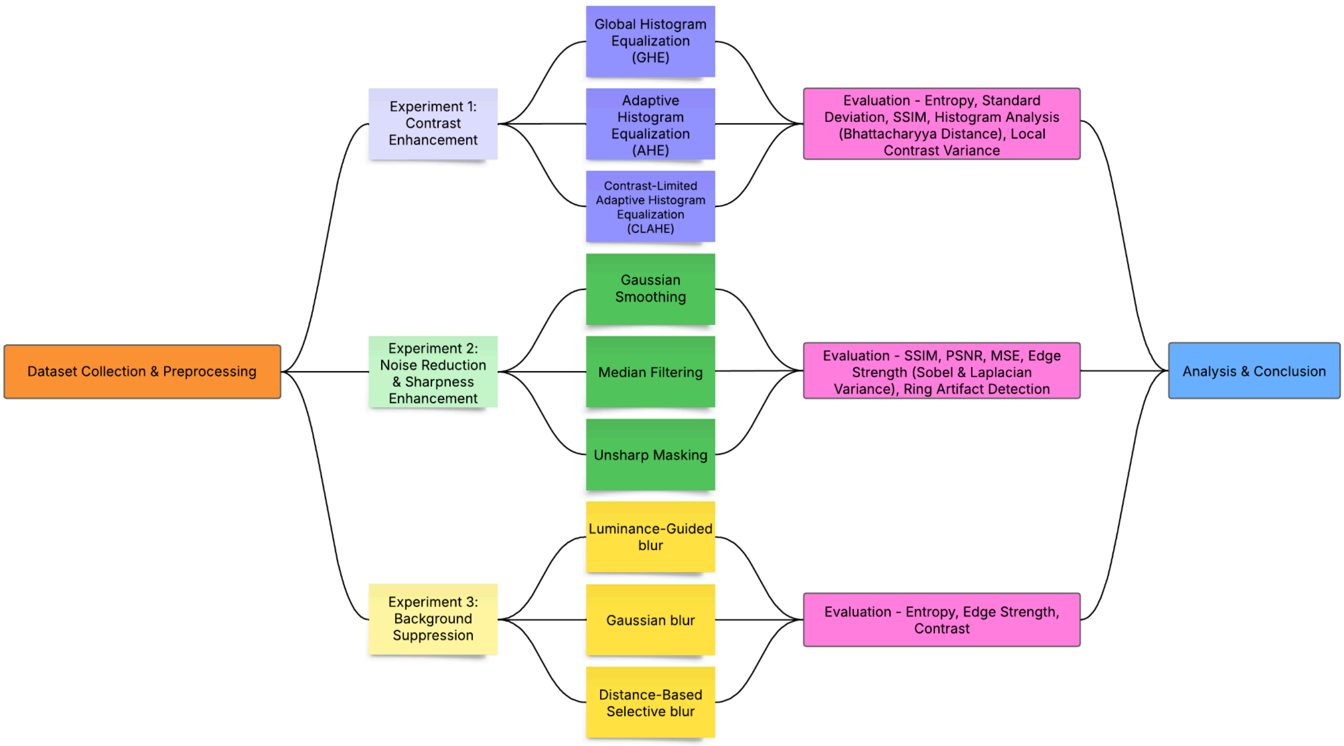


Figure.33 Block Diagram for Three Experiments

IV. Experiment Design

To validate the effectiveness of the proposed digital image processing techniques, we will conduct three distinct experiments, each focusing on a different aspect of image enhancement: contrast improvement, noise reduction and edge enhancement, and background distraction suppression. These experiments will be evaluated using a combination of **quantitative metrics** (such as entropy, PSNR, SSIM, and edge analysis) and **qualitative assessments** (such as user perception surveys and paired image comparisons). This approach ensures a comprehensive analysis of the impact of each enhancement technique, balancing objective measurement with human visual perception.

4.1 Experiment 1: Localized Contrast Enhancement: GHE vs. AHE vs. CLAHE

The first experiment aims to evaluate the effectiveness of Adaptive Histogram Equalization (AHE) and Contrast-Limited Adaptive Histogram Equalization (CLAHE) in enhancing image contrast compared to Global Histogram Equalization (GHE). GHE applies a uniform transformation across the entire image, which can improve contrast globally but may result in loss of local details in areas with uneven lighting. In contrast, AHE and CLAHE enhance contrast locally, making finer details more visible while preventing over-amplification of noise. This experiment will determine which method best improves visibility while preserving a natural appearance.

A dataset of 10 dog images with diverse lighting conditions will be used. Each image will be processed using the following three contrast enhancement techniques:

1. Global Histogram Equalization (GHE) – Applies uniform contrast enhancement across the entire image.
2. Adaptive Histogram Equalization (AHE) – Enhances contrast within localized image regions.
3. Contrast-Limited Adaptive Histogram Equalization (CLAHE) – Limits contrast enhancement to prevent excessive noise amplification.

The effectiveness of these techniques will be assessed using five key evaluation metrics. First, entropy measurement will be conducted, as higher entropy indicates greater intensity variation and improved contrast distribution. Second, standard deviation of intensity values will be calculated to quantify the change in contrast. Third, histogram analysis will be performed, utilizing Bhattacharyya Distance to measure the difference between the original and enhanced histograms, providing insight into how much the intensity distribution has changed. Fourth, the Structural Similarity Index (SSIM) will be used to assess whether the contrast enhancement maintains natural image quality. Lastly, Local Contrast Variance (LCV) will be computed to ensure that contrast enhancement occurs consistently across different regions of the image without introducing artifacts.

This experiment will provide insights into whether localized contrast enhancement (AHE and CLAHE) outperforms global contrast enhancement (GHE) in improving visibility while maintaining a natural appearance.

This experiment will help determine whether localized contrast enhancement (AHE and CLAHE) outperforms global contrast enhancement (GHE) in dog images with challenging lighting conditions.

4.1.1 Parameter Selection:

Method	Parameter	Effect on Image	Test Range
GHE	Number of Histogram Bins (n)	Defines the number of intensity levels used for histogram equalization.	{32, 64, 128, 256}
AHE	Tile Size (NumTiles)	Defines the size of local regions where histogram equalization is applied.	{8×8, 16×16, 32×32}
CLAHE	Tile Size (NumTiles)	Defines the size of local regions where contrast enhancement is applied.	{8×8, 16×16, 32×32}
	Clip Limit (ClipLimit)	Controls the maximum allowed contrast enhancement in each tile.	{2, 4, 6, 8}

Table.15 Parameters Selection for Localized Contrast Enhancement

Table 15 shows the parameters selection for Localized Contrast Enhancement. By experimenting with different tile sizes and clip limits, we find a balance between enhancing details and preserving natural tones.

4.1.2 Metrics Evaluation:

To ensure comparability across different metrics, we first applied normalization, scaling all values between 0 and 1. For Entropy, Standard Deviation (StdDev), Structural Similarity Index (SSIM), and Local Contrast Variability (LCV), we used the standard normalization formula:

Norm_Metric1=(Metric−Min(Metric))/(Max(Metric)−Min(Metric))

Since Bhattacharyya Distance measures histogram similarity, where lower values indicate better contrast enhancement, we applied an inverse normalization to ensure lower values receive higher scores:

Norm_Metric2=1−(Metric−Min(Metric))/(Max(Metric)−Min(Metric))

This adjustment reverses the scale, ensuring consistency in score interpretation across all metrics.

Once normalized, we computed the overall score using a weighted sum of all metrics:

Score = (0.25 × Norm_Entropy) + (0.25 × Norm_StdDev) + (0.20 × Norm_SSIM) + (0.15 × Norm_Bhattacharyya) + (0.15 × Norm_LCV)

These weight assignments were determined based on the relative importance of each metric in evaluating contrast enhancement. Entropy (0.25) was given the highest weight as it represents an image's richness of information and detail retention. Standard Deviation (0.25), which measures global contrast variation, was equally weighted since a well-enhanced image should exhibit strong contrast. SSIM (0.20) was slightly lower in weight as structural preservation is important but not the primary focus of this experiment. Bhattacharyya Distance (0.15), which measures histogram similarity, was inversely normalized to reward more pronounced contrast changes, ensuring that significant differences from the original image received higher scores. Lastly, Local Contrast Variability (LCV, 0.15) was included to assess local contrast enhancement, with a moderate weight to balance its impact, as excessive enhancement can introduce artifacts. This weighted summation approach ensures a balanced evaluation, reflecting both global and local contrast improvements while maintaining structural integrity and assessing histogram differences. Based on the scores, we can choose the best parameters for each method and compare them.

The weights were not arbitrarily chosen, but reflect a task-specific prioritization: emphasis on information richness (entropy) and contrast intensity (StdDev), balanced by structural preservation, histogram divergence, and local detail evaluation. This balanced scoring ensures that enhanced images are both visually rich and perceptually faithful, avoiding over-enhancement or structural degradation.

4.2 Experiment 2: Edge-Preserving Noise Reduction and Sharpness Enhancement: Gaussian Smoothing vs. Median Filtering vs. Unsharp Masking

The second experiment focuses on comparing different noise reduction and sharpening techniques to determine which method best preserves fine details in dog images affected by motion blur and low-light noise. These distortions often degrade image clarity, making it essential to assess how different filtering techniques balance noise suppression and edge preservation.

A dataset of 10 dog images exhibiting motion blur and low-light noise will be selected. Each image will be processed using the following three enhancement techniques:

1. Gaussian Smoothing – A low-pass filter that reduces high-frequency noise but slightly blurs edges.
2. Median Filtering – A non-linear filter effective for removing salt-and-pepper noise while preserving edges.
3. Unsharp Masking – A sharpening technique that enhances edges after applying a blurred mask.

To evaluate performance, five key metrics will be analyzed. The Structural Similarity Index (SSIM) will be used to compare the processed image with the original, with higher SSIM scores indicating better preservation of overall image quality. The Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) will measure the degree of noise suppression, with higher PSNR and lower MSE values indicating better denoising performance. To assess sharpening effectiveness, Edge Strength Analysis will be conducted using Sobel and Laplacian Variance, quantifying the prominence of edges before and after processing. Finally, Ring Artifact Detection will be applied to identify over-sharpening artifacts, such as unwanted halos around edges, which can result from excessive enhancement.

By comparing these methods, this experiment will provide insights into which technique best balances noise removal and feature preservation, ensuring sharp and clear dog images for applications such as pet photography and identification.

4.2.1 Parameter Selection:

Method	Parameter	Effect on Image	Test Range
Gaussian Smoothing	Gaussian Sigma (σ)	Gaussian Sigma (σ) controls the amount of blur applied before extracting high-frequency details.	0.4,0.6,0.8,1.0
Median Filtering	window_sizes	Extracts fine details by subtracting the blurred image from the original.	3*3, 5*5, 7*7
Unshark Masking	Sharpening factor (k)	Sharpening factor (k) determines how much of the extracted high-frequency details are added back	0.5,1.0,1.5,2.0

Table.16 Parameters Selection for Edge-Preserving Noise Reduction and Sharpness Enhancement

Table 16 shows the parameters selection for Edge-Preserving Noise Reduction and Sharpness Enhancement. By experimenting with different sigma values and sharpening factors, we achieve a balance between restoring fine details and preventing unnatural over-sharpening.

4.2.2 Metrics Evaluation:

To quantitatively assess the performance of the image enhancement techniques evaluated in Experiment 2, six complementary image quality metrics were employed. These metrics aimed to capture both noise suppression capabilities and edge-preservation qualities of each filter type. The Structural Similarity Index (SSIM) was used to measure overall perceptual similarity between the filtered image and its original version, where values closer to 1 indicate high structural preservation. The Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) were computed to assess the degree of pixel-level distortion; higher PSNR and lower MSE values reflect more effective noise reduction while retaining image fidelity.

To capture edge-related sharpness and texture, two gradient-based metrics were analyzed. The Sobel Edge Strength was computed by applying the Sobel operator to grayscale images, providing a measure of the intensity of edge transitions. In parallel, the Laplacian Variance was used to quantify fine detail and edge presence through the statistical variance of the Laplacian-filtered image. Both of these metrics increase when edge content is more prominent. Finally, to detect undesirable over-enhancement side effects, such as halo artifacts from aggressive sharpening, a Ring Artifact Detection score was calculated based on the standard deviation of a Laplacian filter applied to the processed image. Lower values in this metric are desirable, as they indicate fewer artificial rings or halos around edges.

Each filtering configuration (Gaussian σ, Median window size, or Unsharp Amount) was tested across all ten dog images, and a composite score was computed for every setting to identify the optimal parameter. The scoring formula combines all six metrics using a weighted sum:

Score = (0.4 × SSIM) + (0.3 × PSNR) - (0.1 × MSE) + (0.1 × Sobel) + (0.1 × LaplacianVar) - (0.1 × Ring)

The scoring system assigns weights based on the importance of each metric in evaluating the overall image enhancement. SSIM (0.4) is given the highest weight because it measures perceptual similarity, which directly impacts the visual quality of the image, a key consideration in edge-preserving and noise-reducing methods. PSNR (0.3) is also weighted highly, as it assesses noise reduction, a central goal of the experiment. Conversely, MSE (−0.1) is penalized, as higher values indicate more distortion and a lower-quality image. Sobel (0.1) and Laplacian Variance (0.1) contribute moderate weights to edge sharpness, ensuring that enhancement techniques preserve fine details without over-enhancing. Ring (−0.1) is similarly penalized to reduce the impact of artifacts like halos, which would diminish the quality of the enhancement.

This weighted scoring system ensures a comprehensive evaluation that balances structural integrity, noise reduction, edge enhancement, and artifact minimization. After scoring, the configuration that achieves the highest score in each method is selected as the optimal setting, ensuring the best performance in terms of both perceptual quality and technical metrics. This ensures that the most suitable parameters are chosen for each technique, providing a reliable and robust enhancement for the dog images, with a detailed evaluation table saved for further analysis.

4.3 Experiment 3: Automated Background Suppression: Luminance-Guided blur vs. Gaussian blur vs. Distance-Based Selective blur

The third experiment compares three classical image-processing-based background suppression techniques to evaluate their effectiveness in reducing distractions while preserving important foreground details (i.e., the dog). These methods avoid machine learning, instead relying on spatial- and luminance-based operations to enhance subject clarity.

A dataset of 10 dog images with low-clutter and high-clutter backgrounds was selected. Each image was processed using the following techniques:

1. Luminance-Guided blur — Applies a strong Gaussian blur weighted by inverse luminance, preserving brighter regions (typically the dog) while blurring darker background regions.
2. Gaussian blur with Foreground Masking — Applies spatial Gaussian blur to the background while keeping the detected foreground sharp.
3. Distance-Based Selective blur — Applies progressively stronger blur to background pixels farther away from the detected foreground, creating a natural depth-of-field effect.

The effectiveness of each technique is quantitatively evaluated using a composite score that integrates three complementary metrics: background entropy, foreground edge sharpness, and foreground-background contrast. These metrics are selected to jointly capture the dual objective of effective background suppression and subject preservation.

Background entropy reflects the level of visual complexity or randomness in the background region. A successful background suppression method is expected to reduce the entropy in non-subject areas by smoothing out texture and clutter, thereby minimizing distractions. Entropy is computed using Shannon's entropy formula on the background pixels, with lower values indicating simpler, less distracting backgrounds.

Foreground edge sharpness measures the clarity and preservation of the subject's structural details, particularly the dog's contours. This is computed by applying the Sobel edge detector to the foreground region and averaging the edge responses. Higher values indicate that the edges remain crisp and well-defined after processing, suggesting that the suppression method does not excessively degrade the subject.

Foreground-background contrast quantifies how distinct the subject is from the background in terms of intensity variations. It is computed as the ratio between the standard deviation of pixel intensities in the foreground and that in the background. A higher contrast score implies a more visually separable subject, which is desirable in both perceptual and computational vision tasks.

Lastly, a Paired Comparison Survey will be conducted, where participants will compare two processed images and choose the one with the best background suppression, ensuring a subjective assessment of image quality.

This experiment aims to determine which of the three purely image processing based techniques: luminance-guided blurring, Gaussian background smoothing, or distance-based selective blurring—offers the most effective approach to background suppression while preserving the clarity and visual prominence of the dog as the primary subject in the image.

4.3.1 Parameter Selection:

Method	Parameter	Effect on Image	Test Range
Luminance	Gaussian Sigma (σ)	Controls blur strength for luminance-weighted blur	{5, 10, 15, 20}

Gaussian	Gaussian Sigma (σ)	Standard blur applied outside foreground mask	{5, 10, 15, 20}
Selective blur	Mutli-Level Sigma (σ)	Varying blur based on distance to foreground	{[5,10,20,30,50] [10,20,30,40,60] [15,30,50,70,90] [20,40,60,80,100]}

Table.17 Parameters Selection for Automated Background Suppression

Table 17 shows the parameters selection for Automated Background Suppression. By experimenting with different filter types and cutoff frequencies, we can achieve an optimal balance between removing distractions and preserving essential pet details.

4.3.2 Metrics Evaluation:

To evaluate the effectiveness of each background suppression technique, we defined three image quality metrics that reflect both background clarity reduction and foreground preservation:

1. Background Entropy : Measures the information content in the background region. Lower entropy indicates a more uniform (and blurred) background, which is desirable.
2. Foreground Edge Strength : Calculated as the average edge magnitude within the segmented dog region using the Sobel operator. This metric reflects the preservation of subject details.
3. Foreground-Background Contrast : Defined as the ratio between the standard deviation of grayscale intensities in the foreground and background. Higher contrast indicates better visual separation between the dog and its surroundings.

Each metric was normalized to a comparable range, and a composite score was computed for each method using the following weighted formula:

Score = 0.3 * (1 - Norm_Entropy/8) +0.4 * Norm_EdegeStrength + 0.3 * Norm_Contrast

Where:

The entropy is normalized by 8 (maximum possible entropy for 8-bit grayscale images),
Norm_EdegeStrength and Norm_Contrast are used as-is since they are bounded and empirically balanced.

The weight distribution in the scoring formula prioritizes Foreground Edge Strength (0.4) and Foreground-Background Contrast (0.3), reflecting the importance of maintaining sharp subject details and ensuring clear visual separation between the subject and the background. Background Entropy (0.3) is weighted equally but slightly lower, as it primarily influences the degree of background suppression. This distribution ensures a balanced evaluation of both the background suppression and foreground preservation.

This approach should allow for a comprehensive analysis of the three background suppression techniques, ensuring the most effective method is identified for preserving subject clarity in challenging background environments.

V. Results and discussion

5.1 Experiment 1: Localized Contrast Enhancement: AHE vs. CLAHE vs. GHE

5.1.1 Quantitative Results

We experimented with various parameter selections for Localized Contrast Enhancement, including different bin sizes, tile sizes, and clip limits, to evaluate their impact on image metrics. After normalizing each metric value to a 0-1 scale, we calculated a final score by weighting each normalized metric according to its importance. This approach allowed us to assess the effectiveness of different parameter configurations in enhancing image contrast.
The score serves as an evaluation tool for assessing the contrast of an image after applying a method with various parameters. A higher score indicates that the selected parameters contribute to better contrast enhancement in the image.

Image	Method	Bins	Entropy	StdDev	SSIM	Bhattacharyya	LCV
D0	GHE	0	5.3219	0.0640	1.0000	0.0000	0.0002
D0	GHE	32	4.5750	0.2975	0.1847	0.9197	0.0190
D0	GHE	64	4.8714	0.2930	0.1915	0.9186	0.0183
D0	GHE	128	5.0750	0.2906	0.1940	0.9198	0.0180
D0	GHE	256	5.2359	0.2895	0.1951	0.9228	0.0179
L0	GHE	0	6.3987	0.1152	1.0000	0.0000	0.0006
L0	GHE	32	4.8709	0.2975	0.5064	0.9011	0.0035
L0	GHE	64	5.6232	0.2927	0.5242	0.8750	0.0033
L0	GHE	128	6.2100	0.2903	0.5332	0.8500	0.0032
L0	GHE	256	6.3464	0.2898	0.5366	0.8436	0.0032

Table.18 Sample of Evaluate Metrics for GHE

Equalization (GHE) applied to images D0.jpg and L0.jpg with different numbers of histogram bins. The entropy values fluctuate as the number of bins increases, with a general trend of increasing entropy at higher bin counts (e.g., L0.jpg shows an increase from 6.2100 at 128 bins to 6.3464 at 256 bins), which suggests that higher bin counts contribute to richer image details.

The standard deviation (StdDev) shows relatively small variations across different bin settings, indicating that GHE does not drastically affect the global contrast distribution. The SSIM (Structural Similarity Index) decreases as the number of bins increases, meaning the processed images diverge slightly from the original image's structure.

Regarding the Bhattacharyya distance, it does not consistently increase as the number of bins increases. Instead, it fluctuates slightly (e.g., for L0.jpg, it decreases from 0.9011 at 32 bins to 0.8500 at 128 bins, and then to 0.8436 at 256 bins). This suggests that while histogram differences exist, they do not always grow monotonically with bin size.

Lastly, LCV (Local Contrast Variance) remains very low across all settings, reinforcing that GHE predominantly enhances global contrast rather than local contrast variations. The results indicate that higher bin counts, particularly 128 and 256 bins, yield better entropy and contrast without excessively distorting the original structure.

Image	Method	Bins	Score
D0	GHE	256	0.4497
D1	GHE	256	0.5472
D2	GHE	256	0.3352
D3	GHE	256	0.4847
D4	GHE	128	0.5310
L0	GHE	256	0.5576
L1	GHE	256	0.7195
L2	GHE	256	0.5874
L3	GHE	256	0.5755
L4	GHE	256	0.7429
5 Dark-Lighting Images Avg Score			0.4696
5 Excess-Lighting Images Avg Score			0.6366
10 Images Avg Score			0.5531

Table.19 Best Bins based on Evaluate Metrics for GHE

Table 19 presents the best-performing bin configurations for Global Histogram Equalization (GHE) based on the computed scores for different images. The results indicate that a bin size of 256 is optimal for most images, as it consistently yields the highest scores. This suggests that using more bins enhances image contrast and detail, leading to better overall performance in the evaluation metrics. However, an exception is observed for D4.jpg, where the best score is achieved at 128 bins instead of 256. This implies that, in some cases, too many bins may lead to excessive enhancement, potentially introducing artifacts or noise. The scores also vary significantly between images, reflecting differences in their inherent contrast and texture properties.

To further analyze performance, we separate the images into dark and light categories. The average score for the five dark images (D0–D4) is 0.4696, while the five light images (L0–L4) yield a higher average score of 0.6366. This indicates that GHE is more effective at enhancing lighter images compared to darker ones. Finally, the overall average score across all ten images is 0.5531, and this value will be used in the following discussion to compare GHE with other enhancement methods.

Image	Method	Tile_Size	Entropy	StdDev	SSIM	Bhattacharyya	LCV
D0	AHE	0	5.3219	0.0640	1.0000	0.0000	0.0002
D0	AHE	8	6.6561	0.1222	0.5979	0.4907	0.0022
D0	AHE	16	6.5244	0.1086	0.5880	0.4898	0.0024
D0	AHE	32	6.4789	0.0984	0.5347	0.5412	0.0028
L0	AHE	0	6.3987	0.1152	1.0000	0.0000	0.0006
L0	AHE	8	7.4450	0.2109	0.6954	0.4066	0.0039
L0	AHE	16	7.3251	0.1876	0.6735	0.3795	0.0043
L0	AHE	32	7.3274	0.1904	0.5782	0.4338	0.0059

Table.20 Sample of Evaluate Metrics for AHE

Table 20 presents the evaluation metrics for Adaptive Histogram Equalization (AHE) applied to images D0.jpg and L0.jpg using different tile sizes. The results show that as the tile size increases, entropy and standard deviation (StdDev) tend to increase, indicating improved contrast enhancement. However, SSIM (Structural Similarity Index) decreases, suggesting that larger tile sizes introduce more structural differences from the original image.

The Bhattacharyya distance increases with larger tile sizes, indicating a greater shift in the histogram distribution compared to the original image. Similarly, LCV (Local Contrast Variance) also increases, reflecting stronger local contrast variations introduced by AHE. Notably, tile size 8 and 16 seem to provide a balance between enhancement and structural preservation, as they show higher entropy and contrast while maintaining a reasonable SSIM value.

Image	Method	Tile_Size	Score
D0	AHE	8	0.2514
D1	AHE	8	0.5979
D2	AHE	8	0.3332
D3	AHE	8	0.3836
D4	AHE	8	0.2570
L0	AHE	8	0.6539
L1	AHE	8	0.7455
L2	AHE	8	0.6953
L3	AHE	8	0.6885
L4	AHE	8	0.8504
5 Dark-Lighting Images Avg Score			0.3646
5 Excess-Lighting Images Avg Score			0.7267
10 Images Avg Score			0.5457

Table.21 Best Tile Size based on Evaluate Metrics for AHE

Table 21 presents the optimal tile size for Adaptive Histogram Equalization (AHE) based on computed scores for different images. The results indicate that a tile size of 8 consistently yields the highest scores across all tested images. This suggests that tile size 8 provides the best balance between local contrast enhancement and structural preservation.

The scores vary across different images, reflecting differences in their contrast and texture characteristics. Some images, such as L4.jpg (0.8504) and L1.jpg (0.7455), achieve significantly higher scores, indicating that AHE with tile size 8 is particularly effective for enhancing contrast in these cases. In contrast, lower scores on darker images such as D0.jpg (0.2514) and D4.jpg (0.2570) suggest reduced effectiveness of AHE in very low-light conditions.

To better understand this variation, we compute the average score for the five dark images (D0–D4), which is 0.3646, and for the five light images (L0–L4), which is 0.7267. This significant difference highlights that AHE performs considerably better on light images. The overall average score across all ten images is 0.5457, which will be used in the subsequent comparison of the three image enhancement algorithms.

Image	Method	Tile_Size	Clip_Limit	Entropy	StdDev	SSIM	Bhattacharyya	LCV
D0	CLAHE	0	0	5.3219	0.0640	1.0000	0.0000	0.0002
D0	CLAHE	8	2	7.1423	0.1507	0.3508	0.5784	0.0058
D0	CLAHE	8	4	7.6364	0.1969	0.1820	0.6767	0.0160
D0	CLAHE	8	6	7.8476	0.2307	0.1323	0.7175	0.0268
D0	CLAHE	8	8	7.9368	0.2540	0.1142	0.7330	0.0357
D0	CLAHE	16	2	7.0081	0.1360	0.3390	0.5753	0.0063
D0	CLAHE	16	4	7.5324	0.1815	0.1697	0.6716	0.0175
D0	CLAHE	16	6	7.7977	0.2209	0.1170	0.7135	0.0304
D0	CLAHE	16	8	7.9211	0.2488	0.0967	0.7312	0.0414
D0	CLAHE	32	2	6.9340	0.1258	0.3245	0.5765	0.0069
D0	CLAHE	32	4	7.5030	0.1775	0.1528	0.6765	0.0198
D0	CLAHE	32	6	7.7665	0.2158	0.1057	0.7121	0.0333
D0	CLAHE	32	8	7.9047	0.2452	0.0866	0.7275	0.0451
L0	CLAHE	0	0	6.3987	0.1152	1.0000	0.0000	0.0006
L0	CLAHE	8	2	7.7032	0.2446	0.5282	0.4702	0.0066
L0	CLAHE	8	4	7.8479	0.2607	0.4028	0.5182	0.0091
L0	CLAHE	8	6	7.8825	0.2587	0.3557	0.5365	0.0097
L0	CLAHE	8	8	7.8928	0.2544	0.3329	0.5520	0.0101
L0	CLAHE	16	2	7.6156	0.2227	0.4800	0.4564	0.0082
L0	CLAHE	16	4	7.8016	0.2444	0.3396	0.5071	0.0129
L0	CLAHE	16	6	7.8606	0.2483	0.2843	0.5350	0.0150
L0	CLAHE	16	8	7.8883	0.2475	0.2595	0.5536	0.0160
L0	CLAHE	32	2	7.4999	0.2082	0.4176	0.4765	0.0104
L0	CLAHE	32	4	7.6642	0.2281	0.2944	0.5108	0.0163
L0	CLAHE	32	6	7.7797	0.2375	0.2340	0.5385	0.0203
L0	CLAHE	32	8	7.8226	0.2395	0.2104	0.5559	0.0221

Table.22 Sample of Evaluate Metrics for CLAHE

Table 22 presents the evaluation metrics for Contrast Limited Adaptive Histogram Equalization (CLAHE) applied to images D0.jpg and L0.jpg with different tile sizes and clip limits. The results indicate that entropy and standard deviation (StdDev) generally increase as tile size increases, suggesting improved contrast enhancement. However, when the clip limit is increased, the entropy and standard deviation values tend to stabilize, preventing excessive contrast amplification.

The SSIM (Structural Similarity Index) gradually decreases as contrast enhancement becomes more aggressive, implying greater structural deviation from the original image. The Bhattacharyya distance fluctuates, indicating varying degrees of histogram changes depending on the combination of tile size and clip limit. The LCV (Local Contrast Variance) values increase significantly for higher tile sizes and clip limits, confirming that CLAHE enhances local contrast more strongly than GHE and AHE.

Overall, the results suggest that moderate tile sizes (8 or 16) and controlled clip limits (e.g., 2 or 4) provide a good balance between enhancement and preservation of structural details. Larger tile sizes with high clip limits may lead to excessive enhancement, introducing artifacts or unnatural contrast variations.

Image	Method	Tile Size	Clip Limit	Score
D0	CLAHE	32	8	0.5446
D1	CLAHE	8	4	0.6105
D2	CLAHE	8	6	0.4036
D3	CLAHE	8	8	0.4548
D4	CLAHE	16	8	0.5967
L0	CLAHE	8	6	0.6682
L1	CLAHE	8	6	0.7359
L2	CLAHE	8	8	0.7179
L3	CLAHE	8	6	0.6824
L4	CLAHE	32	4	0.8274
5 Dark-Lighting Images Avg Score				0.5220
5 Excess-Lighting Images Avg Score				0.7264
10 Images Avg Score				0.6242

Table.23 Best Tile Size and Clip Limit based on Evaluate Metrics for CLAHE

Table 23 presents the optimal tile size and clip limit for Contrast Limited Adaptive Histogram Equalization (CLAHE) based on computed scores for different images. The results indicate that a tile size of 8 with clip limits typically ranging between 4 and 8 consistently achieves high scores, suggesting this configuration offers a strong balance between contrast enhancement and preservation of structural details.

Notably, certain images such as D0.jpg and L4.jpg perform best with a larger tile size of 32, indicating that larger tiles can sometimes be advantageous depending on the image's content and lighting. However, the most frequent optimal configuration remains at tile size 8, supporting the effectiveness of moderate tile sizes combined with controlled clip limits.

To further assess performance variation, we compute the average score for the five dark images (D0–D4), which is 0.5220, and for the five light images (L0–L4), which is 0.7264. These results suggest CLAHE is generally more effective on brighter images, though it also delivers strong results on darker ones compared to other methods. The overall average score across all ten images is 0.6242, which will be used in the comparative evaluation of all three algorithms in the following section.

	GHE	AHE	CLAHE
5 Dark Images Avg Score	0.4696	0.3646	0.5220
5 Light Images Avg Score	0.6366	0.7267	0.7264
10 Images Avg Score	0.5531	0.5457	0.6242

Table.24 Metrics Comparison for Localized Contrast Enhancement

Table 24 compares the average scores of three contrast enhancement methods—GHE, AHE, and CLAHE—based on evaluation metrics such as entropy, SSIM, and local contrast variance. Among the three, CLAHE achieves the highest overall average score (0.6242), demonstrating the best balance between localized contrast enhancement and structural detail preservation.

When analyzing performance on different types of images, CLAHE also outperforms the other methods on dark images, with an average score of 0.5220, compared to GHE's 0.4696 and AHE's 0.3646. On light images, AHE

(0.7267) and CLAHE (0.7264) perform almost equally well, both surpassing GHE (0.6366). These results highlight CLAHE's robustness across lighting conditions, while AHE excels in enhancing already bright images, and GHE provides solid global contrast improvement with consistent, though more moderate, gains.

The computed averages provide a clear basis for selecting the appropriate method depending on the image characteristics and enhancement goals, and will be used in the final discussion to guide algorithm choice.

5.1.2 Qualitative Results

Based on the best scores we have above, we apply the best parameters for each image in GHE, AHE, and CLAHE from Experiment 1 to evaluate the qualitative results based on the output images. The dataset consists of ten images, including five dark images and five light images. To effectively represent the outcomes, we select one dark image and one light image for detailed discussion and analysis.


Figure.34 Original of D0 *Figure.35 GHE of D0*


Figure.36 AHE of D0 *Figure.37 CLAHE of D0*

Image D0.jpg presents a different scenario. The relatively low score of 0.4497 for GHE with 256 bins likely corresponds to a limited overall brightening, failing to bring out sufficient details in the dark image (as seen in Figure 11, which remains quite shadowy). AHE, yielding an even lower score of 0.2514, may produce an image with increased noise amplification, making some details brighter, but also introducing undesirable artifacts (as seen in Figure 12). CLAHE, with a better score of 0.5446 with tile size 32 and clip limit 8, potentially does a better job in balancing enhancing the contrast and make image more visually better (as seen in Figure 13).


Figure. 38 Original of L0 *Figure. 39 GHE of L0*


Figure.40 AHE of L0 *Figure.41 CLAHE of L0*

Examining image L0.jpg, the moderate score of 0.5576 for Global Histogram Equalization (GHE) with 256 bins visually corresponds to a general brightening of the image, as can be seen in Figure 15. The colors might appear somewhat washed out, indicating the global contrast enhancement doesn't finely target specific areas of the image. Adaptive Histogram Equalization (AHE) with a tile size of 8, yielding a higher score of 0.6539, likely demonstrates improved local contrast in the image. In Figure 16, this would translate to sharper details in the dog's fur and the surrounding leaves. However, Contrast Limited Adaptive Histogram Equalization (CLAHE), with a tile size of 8 and a clip limit of 6 achieving the best score of 0.6682, suggests that the enhanced local contrast might be better managed in detail and texture.

5.2 Experiment 2: Edge-Preserving Noise Reduction and Sharpness Enhancement: Gaussian Smoothing vs. Median Filtering vs. Unsharp Masking

5.2.1 Quantitative Results

In this experiment, we evaluated the performance of different edge-preserving noise reduction and sharpness enhancement methods: Gaussian Smoothing, Median Filtering, and Unsharp Masking. We performed quantitative evaluations using various image quality metrics, including SSIM, PSNR, MSE, Sobel, Laplacian Variance, and Ring artifacts.

Image	Method	Sigma	SSIM	PSNR	MSE	Sobel	LaplacanVar	Ring
C0	Gaussian Smoothing	0.4	0.9980	41.1057	0.0001	0.0327	0.0324	0.1645
C0	Gaussian Smoothing	0.6	0.9690	29.7099	0.0011	0.0286	0.0128	0.0866
C0	Gaussian Smoothing	0.8	0.9370	26.7836	0.0021	0.0254	0.0080	0.0527
C0	Gaussian Smoothing	1.0	0.9136	25.4524	0.0028	0.0233	0.0066	0.0371
C1	Gaussian Smoothing	0.4	0.9990	45.3342	0.0000	0.0328	0.0187	0.1058
C1	Gaussian Smoothing	0.6	0.9837	33.6416	0.0004	0.0287	0.0113	0.0615
C1	Gaussian Smoothing	0.8	0.9649	30.4196	0.0009	0.0246	0.0091	0.0405
C1	Gaussian Smoothing	1.0	0.9494	28.8431	0.0013	0.0220	0.0084	0.0299

Table.25 Sample of Evaluate Metrics for Gaussian Smoothing

Table 25 presents the sample evaluation metrics for Gaussian Smoothing at different sigma values. The results demonstrate a decrease in SSIM and PSNR as the sigma value increases, indicating a reduction in the image quality, while MSE and Ring artifacts increase, particularly at higher sigma values. The 0.4 sigma configuration consistently delivers the best results across all metrics, with a high SSIM and PSNR, as well as low MSE and Ring artifacts.

Image	Method	Sigma	Score
C0	Gaussian Smoothing	0.4	12.7210
C1	Gaussian Smoothing	0.4	13.9944
C2	Gaussian Smoothing	0.4	14.5704
C3	Gaussian Smoothing	0.4	14.9694
C4	Gaussian Smoothing	0.4	13.2557
C5	Gaussian Smoothing	0.4	13.6687
C6	Gaussian Smoothing	0.4	14.8520
C7	Gaussian Smoothing	0.4	12.6805
C8	Gaussian Smoothing	0.4	15.7508
C9	Gaussian Smoothing	0.4	14.1552
10 Images Avg Score			14.0618

Table.26 Best Sigma based on Evaluate Metrics for Gaussian Smoothing

Table 26 shows the final scores for Gaussian Smoothing based on different sigma values. The highest score, 14.0618, was achieved with sigma 0.4, suggesting that this configuration optimally balances the image's contrast, sharpness, and noise reduction. The consistency of the results across the 10 images indicates the stability of this method at this sigma value.

Image	Method	Window Size	SSIM	PSNR	MSE	Sobel	LaplacianVar	Ring
C0	Median Filtering	3*3	0.9146	25.4810	0.0028	0.0250	0.0113	0.0773
C0	Median Filtering	5*5	0.8638	23.3390	0.0046	0.0245	0.0072	0.0456
C0	Median Filtering	7*7	0.8299	22.2511	0.0060	0.0270	0.0061	0.0305
C0	Median Filtering	3*3	0.9525	29.4052	0.0011	0.0244	0.0117	0.0647
C1	Median Filtering	5*5	0.9156	26.8187	0.0021	0.0193	0.0097	0.0477
C1	Median Filtering	7*7	0.8906	25.5091	0.0028	0.0159	0.0090	0.0408

Table.27 Sample of Evaluate Metrics for Median Filtering

For Median Filtering, Table 27 displays the evaluation metrics for different window sizes (3x3, 5x5, and 7x7). As expected, smaller window sizes generally provided better SSIM and PSNR values, with the 3x3 window delivering the best performance. Specifically, the 3x3 window size resulted in an average score of 9.3413 (Table 28), demonstrating that smaller windows are more effective at preserving edges and reducing noise without causing excessive blurring.

Image	Method	Window Size	Score
C0	Median Filtering	3*3	8.0058
C1	Median Filtering	3*3	9.1996
C2	Median Filtering	3*3	9.8262
C3	Median Filtering	3*3	10.1381
C4	Median Filtering	3*3	8.5261
C5	Median Filtering	3*3	8.9465
C6	Median Filtering	3*3	10.3860
C7	Median Filtering	3*3	7.7944
C8	Median Filtering	3*3	10.9414
C9	Median Filtering	3*3	9.6488
10 Images Avg Score			9.3413

Table.28 Best Window Size based on Evaluate Metrics for Median Filtering

Table 28 summarizes the performance of Median Filtering using different window sizes based on the computed scores across 10 test images. The 3×3 window size clearly outperforms the 5×5 and 7×7 configurations, achieving the highest average score of 9.3413. This indicates that a smaller window preserves more image details and edge information while still effectively reducing noise. Larger window sizes tend to introduce more blurring, which results in lower SSIM and PSNR values and an increase in distortion-related metrics such as MSE. Therefore, the 3×3 window is identified as the optimal configuration for median filtering in this experiment.

Image	Method	Unsharp Amount	SSIM	PSNR	MSE	Sobel	LaplacianVar	Ring
C0	Unsharp Masking	0.5	0.9859	31.4730	0.0007	0.0381	0.0828	0.2778
C0	Unsharp Masking	1.0	0.9534	25.4524	0.0028	0.0413	0.1286	0.3505
C0	Unsharp Masking	1.5	0.9129	21.9306	0.0064	0.0444	0.1788	0.4157
C0	Unsharp Masking	2.0	0.8704	19.4318	0.0114	0.0471	0.2311	0.4741
C1	Unsharp Masking	0.5	0.9911	34.8637	0.0003	0.0388	0.0373	0.1724
C1	Unsharp Masking	1.0	0.9696	28.8431	0.0013	0.0423	0.0562	0.2207
C1	Unsharp Masking	1.5	0.9413	25.3212	0.0029	0.0445	0.0791	0.2674
C1	Unsharp Masking	2.0	0.9911	34.8637	0.0003	0.0388	0.0373	0.1724

Table.29 Sample of Evaluate Metrics for Unsharp Masking

Table 29 presents the evaluation metrics for Unsharp Masking at different unsharp amounts. The unsharp amount of 0.5 yielded the best performance across all metrics, with an average score of 10.9634 (Table 30), indicating that this level of sharpening achieved the best balance between sharpness enhancement and noise introduction. Higher unsharp amounts, particularly above 1.0, led to a decrease in SSIM and PSNR, suggesting that excessive sharpening can lead to artifacts and reduced image quality.

Image	Method	Unsharp Amount	Score
C0	Unsharp Masking	0.5	9.8205
C1	Unsharp Masking	0.5	10.8459
C2	Unsharp Masking	0.5	11.4941
C3	Unsharp Masking	0.5	11.7636
C4	Unsharp Masking	0.5	10.3555
C5	Unsharp Masking	0.5	10.5142
C6	Unsharp Masking	0.5	11.4010
C7	Unsharp Masking	0.5	9.7441
C8	Unsharp Masking	0.5	12.6382
C9	Unsharp Masking	0.5	11.0564
10 Images Avg Score			10.9634

Table.30 Best Unshap Amount based on Evaluate Metrics for Unsharp Masking

Table 30 displays the scores for Unsharp Masking at various sharpening amounts. The unsharp amount of 0.5 consistently yields the best overall performance across all test images, with an average score of 10.9634. This amount enhances the image sharpness without excessively amplifying noise or introducing significant artifacts. As the unsharp amount increases (e.g., 1.0, 1.5, 2.0), the metrics such as Ring artifacts and MSE worsen, indicating oversharpening effects that degrade image quality. Thus, a moderate sharpening setting (0.5) is optimal for balancing clarity and naturalness in the processed images.

	Gaussian Smoothing	Median Filtering	Unsharp Masking
10 Low-Resolution Images Avg Score	14.0618	9.3413	10.9634

Table.31 Metrics Comparison for Edge-Preserving Noise Reduction and Sharpness Enhancement

In Table 31, a comparison of the average scores across all methods—Gaussian Smoothing, Median Filtering, and Unsharp Masking—shows that Gaussian Smoothing with sigma 0.4 provides the best performance in terms of edge-preserving noise reduction, followed by Unsharp Masking and Median Filtering. Gaussian Smoothing achieved the highest score of 14.0618, which demonstrates its superiority in edge preservation and noise reduction.

5.2.2 Qualitative Results

The qualitative analysis of the images also supports the findings from the quantitative evaluations.

Figure.42 Original of C3

Figure.43 Gaussian Smoothing of C3

Figure.44 Median Filtering *of C3*

Figure.45 Unsharp Masking of C3

Figures 42-45 show the results for image C3, where the image after Gaussian smoothing (Figure 43) is a little smoother and the noise is somewhat suppressed, but the overall image looks very similar to the original image. This shows that under the current sigma value setting, Gaussian Smoothing achieves a relatively mild noise reduction effect. Median filtering (Figure 44) smoothes the image texture to a certain extent, and the noise removal is stable, but it also makes the details of the hair slightly flatter and the three-dimensional sense slightly reduced. The unsharp mask (Figure 45) highlights the edge texture of the dog's hair and enhances the overall clarity, but highlights the messy details of the dog's hair.

Figure.46 Original of C4

Figure.47 Gaussian Smoothing of C4

Figure.48 Median Filtering of C4

Figure.49 Unsharp Masking of C4

Similarly, Figures 46-49 show the results for image C4, where Gaussian smoothing (Figure 47) again provides the most balanced output. Median filtering (Figure 48) and unsharp mask (Figure 49) show similar trends to R0, with the former having a lower degree of sharpening and the latter having a higher degree of sharpening, clearly highlighting the dog's facial features and ears.

4.3 Experiment 3: Automated Background Suppression: Luminance-Guided blur vs. Gaussian blur vs. Distance-Based Selective blur

In this experiment, we evaluated and compared three different approaches for background suppression: Luminance-Guided Blur, Gaussian Blur, and Distance-Based Selective Blur. Each method was tested on a set of images, and their performance was assessed using entropy, edge strength, and contrast—key indicators of background suppression effectiveness and foreground clarity.

Image	Method	LuminanceSigma	Entropy	EdgeStrength	Contrast
E0	Luminance blur	5	5.6516	0.0320	1.5718
E0	Luminance blur	10	5.6516	0.0320	1.5718
E0	Luminance blur	15	5.6516	0.0320	1.5718
E0	Luminance blur	20	5.6516	0.0320	1.5718
H0	Luminance blur	5	6.7314	0.0179	1.6034
H0	Luminance blur	10	6.7314	0.0179	1.6034
H0	Luminance blur	15	6.7314	0.0179	1.6034
H0	Luminance blur	20	6.7314	0.0179	1.6034

Table.32 Sample of Evaluate Metrics for Luminance blur

Table 32 shows that the Luminance-Guided Blur method yields constant metric values regardless of the sigma used. This is expected since the luminance threshold likely determines the region to be blurred, and the values stabilize when the region remains unchanged. However, the entropy and edge strength are relatively lower compared to other methods, indicating limited differentiation between foreground and suppressed background.

Image	Method	Luminance Sigma	Score
E0	Luminance blur	5	0.5724
E1	Luminance blur	5	0.4258
E2	Luminance blur	5	0.4491
E3	Luminance blur	5	0.4173
E4	Luminance blur	5	0.3106
H0	Luminance blur	5	0.5357
H1	Luminance blur	5	0.5674
H2	Luminance blur	5	0.4074
H3	Luminance blur	5	0.5007
H4	Luminance blur	5	0.3112
5 Low-Clutter Images Avg Score			0.4350
5 High-Clutter Images Avg Score			0.4645
10 Images Avg Score			0.4498

Table.33 Best LuminanceSigma based on Evaluate Metrics for Luminance blur

Table 33 summarizes the best sigma setting for the Luminance-Guided Blur method. A sigma value of 5 was used consistently, as varying the sigma produced minimal impact on the performance. This is likely because the method's effect is primarily governed by the luminance distribution of the image itself, rather than the blur radius. The blur operation applies a smooth, intensity-guided blending that is less sensitive to small changes in the blur parameter.

The method achieved an overall average score of 0.4498, which is notably lower than the scores of the contrast enhancement techniques analyzed previously. When separating the images by complexity, the average score for low-clutter images (E0–E4) is 0.4350, while high-clutter images (H0–H4) yield a slightly higher average score of 0.4645. These results suggest that Luminance-Guided Blur may provide marginally better results on visually complex scenes but remains generally less effective at enhancing contrast and edge structure compared to GHE, AHE, or CLAHE.

Due to its relatively lower performance, this method may be better suited as a preprocessing step or for specific use cases where subtle enhancement and low visual noise are preferred over aggressive contrast amplification.

Image	Method	GaussianSigma	Entropy	EdgeStrength	Contrast
E0	Gaussian blur	5	5.5487	0.0319	1.6428
E0	Gaussian blur	10	5.6963	0.0319	1.5930
E0	Gaussian blur	15	5.8038	0.0319	1.5341
E0	Gaussian blur	20	5.8974	0.0319	1.4856
H0	Gaussian blur	5	6.6226	0.0178	1.7392
H0	Gaussian blur	10	6.6284	0.0178	1.8330
H0	Gaussian blur	15	6.6479	0.0178	1.9266
H0	Gaussian blur	20	6.6421	0.0178	2.0273

Table.34 Sample of Evaluate Metrics for Gaussian blur

Table 34 illustrates the performance of Gaussian Blur applied globally to the image background. As the sigma increases, entropy gradually increases while contrast decreases—indicating that Gaussian blur smooths both background and some foreground details. While it achieves better background suppression than luminance blur, it does so at the expense of edge sharpness in key regions.

Image	Method	Gaussian Sigma	Score
E0	Gaussian blur	5	0.5976
E1	Gaussian blur	15	0.6156
E2	Gaussian blur	5	0.5545
E3	Gaussian blur	20	0.6004
E4	Gaussian blur	20	0.3990
H0	Gaussian blur	20	0.6663
H1	Gaussian blur	5	0.6448
H2	Gaussian blur	20	0.6488
H3	Gaussian blur	10	0.5265
H4	Gaussian blur	20	0.4989
5 Low-Clutter Images Avg Score			0.5534
5 High-Clutter Images Avg Score			0.5971
10 Images Avg Score			0.5752

Table.35 Best GaussiaSigma based on Evaluate Metrics for Gaussian blur

Table 35 presents the best-performing sigma values for Gaussian Blur based on evaluation scores across different images. The sigma values vary per image, with settings such as 5, 10, 15, and 20 selected to optimize the trade-off between background suppression and preservation of foreground detail. The overall average score is 0.5752, indicating moderate effectiveness—better than Luminance-Guided Blur but still behind more adaptive contrast enhancement techniques like CLAHE.

Breaking the results down by image complexity, the average score for low-clutter images (E0–E4) is 0.5534, while high-clutter images (H0–H4) achieve a slightly higher average of 0.5971. This suggests that Gaussian Blur may be more effective on visually complex scenes where strong background smoothing helps emphasize the main subjects.

While Gaussian Blur lacks the localized adaptability of AHE or CLAHE, its simplicity and relatively stable performance make it suitable for scenarios that require general background softening without introducing structural artifacts.

Image	Method	SelectiveSigmas	Entropy	EdgeStrength	Contrast
E0	Selective blur	5,10,20,30,50	5.6059	0.0320	1.6734
E0	Selective blur	10,20,30,40,60	5.7949	0.0320	1.6277
E0	Selective blur	15,30,50,70,90	5.9375	0.0320	1.6399
E0	Selective blur	20,40,60,80,100	5.9942	0.0320	1.6659
H0	Selective blur	5,10,20,30,50	6.4671	0.0179	2.2271
H0	Selective blur	10,20,30,40,60	6.3953	0.0179	2.4203
H0	Selective blur	15,30,50,70,90	6.1890	0.0179	2.8254
H0	Selective blur	20,40,60,80,100	6.1052	0.0179	2.9734

Table.36 Sample of Evaluate Metrics for Selective blur

Table 36 demonstrates the performance of Distance-Based Selective Blur, which applies varying levels of blur intensity depending on object distance or spatial layout. This method consistently improves contrast while preserving edge strength in foreground regions, with increasing entropy values indicating better separation of detail.

Image	Method	SelectiveSigmas	Score
E0	Selective blur	5,10,20,30,50	0.6046
E1	Selective blur	20,40,60,80,100	0.6605
E2	Selective blur	5,10,20,30,50	0.5772
E3	Selective blur	20,40,60,80,100	0.7992
E4	Selective blur	20,40,60,80,100	0.5498
H0	Selective blur	20,40,60,80,100	0.9702
H1	Selective blur	5,10,20,30,50	0.6694
H2	Selective blur	20,40,60,80,100	0.8282
H3	Selective blur	20,40,60,80,100	0.5904
H4	Selective blur	15,30,50,70,90	0.5534
5 Low-Clutter Images Avg Score			0.6383
5 High-Clutter Images Avg Score			0.7223
10 Images Avg Score			0.6803

Table.37 Best SelectiveSigmas based on Evaluate Metrics for Selective blur

Table 37 reports the best-performing sets of sigma values for Selective Blur across different images. This method applies multiple scales of blur selectively, allowing for targeted smoothing in less important regions while preserving detail in key areas. The result is an overall average score of 0.6803, the highest among all blur-based methods evaluated.

Analyzing performance by image complexity, the average score for low-clutter images (E0–E4) is 0.6383, while high-clutter images (H0–H4) reach an even higher average of 0.7223. This demonstrates the method's adaptability and effectiveness in handling visually dense scenes where preserving object boundaries is critical.

Selective Blur outperforms both Luminance Blur (0.4498) and Gaussian Blur (0.5752) in all metrics, indicating superior balance between structural preservation and background suppression. Its ability to dynamically apply varying levels of blur makes it particularly well-suited for complex images that require nuanced enhancement.

	Luminance blur	Gaussian blur	Selective blur
5 Low-Clutter Images Avg Score	0.4350	0.5534	0.6383
5 High-Clutter Images Avg Score	0.4645	0.5971	0.7223
10 Images Avg Score	0.4498	0.5752	0.6803

Table.38 Metrics Comparison for Automated Background Suppression

Table 38 provides a side-by-side comparison of the average scores for the three background suppression methods. Distance-Based Selective Blur achieves the best overall performance, with an average score of 0.6803, demonstrating its superior ability to adaptively suppress background details while preserving important structures. It is followed by Gaussian Blur with a score of 0.5752, which offers a more uniform but still effective suppression, and Luminance Blur, which shows the lowest performance at 0.4498.

When broken down by image complexity, Selective Blur also leads in both low-clutter (0.6383) and high-clutter (0.7223) image categories, indicating consistent strength across diverse visual scenes. Gaussian Blur performs moderately well on both (0.5534 for low-clutter and 0.5971 for high-clutter), while Luminance Blur trails in both cases (0.4350 and 0.4645, respectively).

These results confirm that context-aware, multi-scale selective blurring is more effective than fixed or luminance-based methods, especially when structural clarity and visual aesthetics are critical.

5.2.2 Qualitative Results

The qualitative analysis of the images also supports the findings from the quantitative evaluations.

Figure.50 Original of E0

Figure.51 Luminance blur of E0

Figure.52 Gaussian blur of E0 *Figure.53 Selective blur of E0*

Figure 50 shows the original image of E0, which contains clear distinctions between the foreground and background. Figure 51 illustrates the effect of Luminance-Guided Blur, which applies a soft focus across the image based on luminance intensity. This method reduces overall contrast but does not achieve clear separation between foreground and background. Figure 52 applies Gaussian Blur, creating a stronger smoothing effect on the entire image, but also causing some foreground edges to blur. Figure 53 highlights the result of Distance-Based Selective Blur, which better preserves foreground clarity while strongly suppressing background noise, resulting in a sharper and more visually focused appearance.

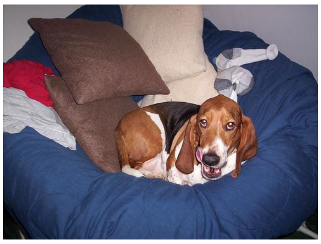
 

Figure.54 Original of H0 *Figure.55 Luminance blur of H0*

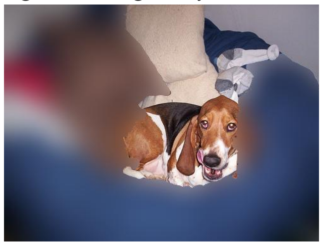
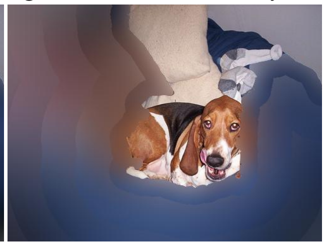
 

Figure.56 Gaussian blur of H0 *Figure.57 Selective blur of H0*

Figure 54 presents the original image of H0, where complex background elements can interfere with the subject's clarity. Figure 55 shows the effect of Luminance-Guided Blur, which applies a luminance-weighted softening across the image. While it reduces overall contrast, it does not provide targeted background suppression or improve subject salience. Figure 56 demonstrates Gaussian Blur, which achieves stronger background smoothing but causes the subject boundaries to lose definition. Figure 57 presents the result of Distance-Based Selective Blur, producing the most visually appealing effect—foreground elements are sharply preserved while the background is smoothly suppressed, highlighting the effectiveness of this method for perceptual quality.

VI. Gantt Chart

	Owner	Jan 2025	Feb 2025	Mar 2025	Apr 2025
Dataset Collection & Preprocessing	All	Done			
Experiment 1: Contrast Enhancement - AHE	JY		Done		
Experiment 1: Contrast Enhancement - GHE	YP		Done		
Experiment 1: Contrast Enhancement - CLAHE	HY		Done		
Evaluation - Experiment 1	JY		Done		
Experiment 2: Noise Reduction & Sharpness Enhancement - Unsharp Masking	JY			Done	
Experiment 2: Noise Reduction & Sharpness Enhancement - Gaussian Smoothing	YP			Done	
Experiment 2: Noise Reduction & Sharpness Enhancement - Median Filtering	HY			Done	
Evaluation - Experiment 2	YP			Done	
Experiment 3: Background Suppression - Luminance blur	JY				Done
Experiment 3: Background Suppression - Gaussian blur	YP				Done
Experiment 3: Background Suppression - Distance-Based Selective blur	HY				Done
Evaluation - Experiment 3	HY				Done
Final Analysis & Conclusion	All				Done
Powerpoint & Report	All				Done

VII. CONCLUSION

This project investigated three key digital image enhancement techniques—contrast enhancement, sharpness improvement, and background suppression—to address common quality issues in dog photography.

In Experiment 1, which focused on contrast enhancement using Global Histogram Equalization (GHE), Adaptive Histogram Equalization (AHE), and Contrast Limited Adaptive Histogram Equalization (CLAHE). We tested various parameter combinations and evaluated results using five normalized metrics. CLAHE achieved the best average score (0.6342), showing strong adaptability and superior performance in enhancing image visibility while preserving details, although it required careful parameter tuning. GHE was effective in global enhancement but less flexible for images with complex lighting, and AHE provided localized contrast improvements but struggled with dark images.

In Experiment 2, we explored three edge-preserving noise reduction and sharpness enhancement methods: Gaussian Smoothing, Median Filtering, and Unsharp Masking. Gaussian Smoothing with a low sigma (0.4) achieved the highest average score (14.06), effectively reducing noise while preserving edges. Unsharp Masking followed closely with strong sharpening results but was more prone to introducing artifacts at higher sharpening levels. Median Filtering performed adequately but was more limited in sharpness recovery. The findings suggest that for dog images affected by motion blur or noise, Gaussian Smoothing provides the best overall trade-off between clarity and natural appearance.

In Experiment 3, we evaluated background suppression techniques to enhance subject focus: Luminance-Guided Blur, Gaussian Blur with Foreground Masking, and Distance-Based Selective Blur. Among the three, Distance-Based Selective Blur outperformed others with the highest average score (0.6803). It preserved foreground edge sharpness while progressively blurring background distractions, creating a depth-of-field effect similar to professional portrait photography. Gaussian Blur performed well with foreground masking but lacked fine control, while Luminance Blur showed limited improvement and struggled in scenes with complex lighting or similar foreground-background tones.

Overall, our results demonstrate that no single technique universally solves all image quality issues—the optimal enhancement strategy depends on the specific condition of each image. Adaptive approaches like CLAHE and selective blurring offer the most potential but require precise parameter tuning. Going forward, combining these enhancement stages into an automated pipeline with built-in image analysis and adaptive parameter selection could enable robust, high-quality dog image processing with minimal manual intervention.

For future work, we propose integrating the contrast enhancement, sharpness boosting, and background suppression steps into a single end-to-end pipeline to streamline processing and enable real-time or batch enhancement with minimal manual tuning. Additionally, we aim to explore machine learning and deep learning techniques for automated background suppression and adaptive enhancement, allowing the system to learn from data and generalize more effectively to diverse noise patterns and image conditions.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Professor Rubi for her invaluable guidance, insightful suggestions, and continuous support throughout the course of this project. Her expertise and encouragement were instrumental in helping us navigate challenges and achieve our goals.

REFERENCES

[1] "Stanford Dogs Dataset," *Kaggle*, Nov. 13, 2019. https://www.kaggle.com/datasets/jessicali9530/stanford-dogs-dataset/data?select=images

[2] Zou, Ding-Nan, et al. "A New Dataset of Dog Breed Images and a Benchmark for Finegrained Classification." Computational Visual Media, vol. 6, no. 4, Oct. 2020, pp. 477–87.

https://doi.org/10.1007/s41095-020-0184-6.

[3] Loh, Yuen Peng, and Chee Seng Chan. "Getting to Know Low-light Images With the Exclusively Dark Dataset." Computer Vision and Image Understanding, vol. 178, Nov. 2018, pp. 30–42. https://doi.org/10.1016/j.cviu.2018.10.010.

APPENDIX

We think this course was very helpful and practical. It taught us many useful image processing techniques. The programming assignments (PAs) were well connected to what we learned in class. They were not too easy, but they were interesting and fun to work on. The written assignments (CAs) helped us understand basic statistics and gave us a better idea of how things work, not just how to write code.

For the project, we first planned something closer to computer vision. Later, with the professor's advice, we changed the direction to focus more on digital image processing, which matched the course better. One problem we had was that the project started early in the semester, but we didn't yet know much about the image processing methods we would learn. It was hard to plan the experiment design when we didn't know what tools or techniques we could use. It would help if students could get a list or preview of the image processing topics early on, so it's easier to do experiment design.

Overall, we really enjoyed the course. The lectures were clear, the assignments were meaningful, and the project let us use what we learned in a creative way. Thank you for all the support and suggestions throughout the semester!