EVALUATING THE IMPACT OF IMAGE PREPROCESSING ON FOOD CLASSIFICATION: A COMPARISON OF LOW-LIGHT ENHANCEMENT, DEBLURRING, AND DOWNSCALING

Vincent Kao, Cheng-Yan Juang, Tien-Hao Chen

Department of Electrical and Computer Engineering
University of California, San Diego

ABSTRACT

This study discusses the impact of image preprocessing on a food type classification model. We use a standard computer vision model as our base, and apply a three kind preprocessing pipeline, including low-light enhancement, image deblurring, and downscaling. Each kind is evaluated with three representative algorithms. Using our own dataset, we compare classification accuracy in three scenarios: with unprocessed images on a pretrained model, with preprocessed images on the same pretrained model, and with unprocessed images on a model that is fine-tuned with new data we collected. Our results reveal which preprocessing strategies most benefit pretrained classifiers and whether spending effort on preprocessing or fine-tuning is more effective when adapting a model to new data.

Index Terms— Image processing, low-light enhancement, deblurring, downscaling

1. INTRODUCTION

Food image classification plays an important role in applications such as dietary evaluation to understand intake patterns and estimate nutrient levels. However, it faces several practical challenges. Real-world images are often affected by issues such as poor lighting, motion blur, and varying resolutions, which can degrade visual quality and obscure discriminative features essential for accurate classification. These degradations introduce inconsistencies in brightness, sharpness, and scale, which reduce the robustness of deep learning models that rely on stable visual representations.

To address these challenges, our project investigates the impact of image preprocessing techniques across three domains: low-light enhancement, image deblurring, and downscaling for model compatibility. These algorithms were applied as preprocessing stages to enhance input quality and consistency prior to classification. We evaluate their effects on the performance of a SigLIP model fine-tuned on the Food-101 dataset, aiming to determine how different enhancement strategies contribute to more robust and accurate food recognition.

2. RELATED WORKS

We reviewed related works in three significant image processing directions, which are low-light enhancement, image deblurring, and downscaling. Each direction represents an aspect of image quality degradation that can significantly affect the performance of the computer vision model. Through the analysis of representative research papers that applied algorithms within these domains, we summarized the key methodologies, strengths, and limitations of the algorithms used in each category.

2.1. Low-light Enhancement

Low-light conditions often degrade image quality and hinder the performance of classification models. Poor illumination reduces contrast and obscures important features, which makes it harder for models to extract meaningful patterns. To address this issue, various image enhancement techniques aim to improve visibility and contrast before classification. Some commonly used approaches are gamma transformation, adaptive histogram equalization (AHE), contrast-limited adaptive histogram equalization (CLAHE), and Retinex-based enhancement. Each of them offer different trade-offs between simplicity, adaptiveness, and perceptual quality.

2.1.1. Gamma transformation

Gamma transformation is a nonlinear intensity mapping that adjusts brightness according to a power-law function. By setting the gamma value less than 1, dark regions can be brightened while maintaining a smooth tone distribution. The method is simple, fast, and suitable for global brightness correction, but it does not handle uneven lighting well. Adaptive variants improve this by calculating gamma based on image statistics. For example, Huang et al. (2013)[1] proposed an adaptive gamma correction using the cumulative intensity distribution to brighten underexposed images without overexposing bright regions.

2.1.2. Adaptive Histogram Equalization (AHE)

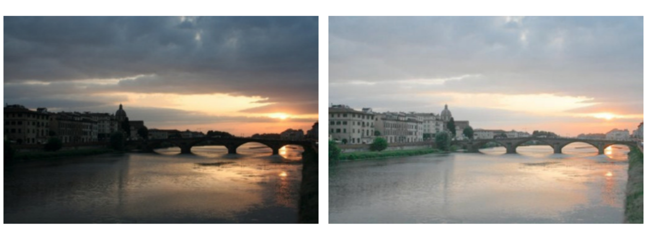
AHE enhances image contrast by computing several histograms corresponding to distinct regions of the image and using them to redistribute the intensity values of the image locally. Unlike global histogram equalization, which applies a single transformation to the entire image, AHE adapts to local intensity variations, making it highly effective for revealing details in both dark and bright regions simultaneously. This adaptability improves visibility in images with non-uniform illumination or low contrast. However, AHE tends to over-amplify noise in relatively homogeneous areas, as local contrast enhancement can exaggerate minor intensity fluctuations. It also involves a higher computational complexity compared to global methods, due to the need for processing per region.

2.1.3. Contrast-Limited Adaptive Histogram Equalization (CLAHE)

CLAHE enhances local contrast by dividing an image into small tiles and equalizing the histogram of each region, while limiting noise amplification. This approach reveals hidden details in shadowed areas and improves feature visibility for recognition systems. Compared with global methods, CLAHE produces better local contrast and edge clarity but requires careful parameter selection to avoid artifacts. A review of CLAHE-based methods in face recognition has shown its effectiveness in improving visual quality and recognition performance under poor lighting conditions[2].

2.1.4. Single-Scale Retinex (SSR)

Retinex-based methods, inspired by human vision, model an image as the product of illumination and reflectance. The Single-Scale Retinex (SSR) algorithm estimates illumination using a Gaussian filter and enhances the reflectance to recover detail from dark areas while preserving overall color balance. SSR handles non-uniform lighting more effectively than global methods, although it requires more intensive computation and is sensitive to parameter settings. Liu et al. (2021) presented a fast Retinex-based approach that improves visibility and color consistency in low-light images[3].



Before (left) and after (right) applying SSR

In summary, these three enhancement techniques represent strategies for improving image quality before classification. Together, they offer a comprehensive foundation for evaluating how classical image processing can improve classification performance under low-light conditions.

2.2. Image Deblurring

Image deblurring has been studied for decades and evolves from classical signal processing models to modern deep learning frameworks. In this report, we refer to the paper by Li [4], which categorizes existing image deblurring research into two main directions, which are traditional model-based methods and deep learning–based methods. To better understand the strengths and weaknesses of each category, a representative algorithm and the work chosen from each category are briefly discussed below.

2.2.1. Traditional Model-Based Method - Wiener Filter

Early studies focused on modeling the blurring process mathematically and restoring images through optimization or frequency-domain filtering. They are primarily designed for non-blind deblurring, assuming that the blur kernel is known. However, since they do not fully utilize the prior statistical information of natural images, their restorations often have inaccurate, oversmoothed edges and lose texture detail. A classic example is the Wiener Filter, which reconstructs images by minimizing the mean square error in the frequency domain.

It assumes that the degradation process can be represented as a convolution of the ideal image with a blur kernel, corrupted by additive noise. Mathematically, the Wiener filter reconstructs the image in the Fourier domain according to:

$$\hat{F}(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_n(u,v)}{S_f(u,v)}} G(u,v) \tag{1}$$

where $H(u,v)$ is the point spread function (PSF), $S_n(u,v)$ and $S_f(u,v)$ are the noise and image power spectra, and $G(u,v)$ is the degraded image.

By weighting the inverse of the PSF according to the noise-to-signal ratio, the Wiener filter effectively balances deblurring and noise suppression[5]. According to Bojarczak and Lukasik[5], the Wiener filter yields excellent results when the variance and type of noise are known.

In their experiments, when the filter's noise variance was adjusted correctly, the reconstructed image achieved a high improvement in Signal-to-Noise Ratio (ISNR = 1.55 dB). However, when the assumed noise variance was mismatched, the performance dropped drastically (ISNR = –0.13 dB), resulting in visible artifacts and excessive smoothing. This shows that Wiener filtering is highly sensitive to the accuracy of its parameters.

The main limitation highlighted in the paper is that real-world blurring cases rarely provides knowledge of the noise statistics or blur kernel. In real-world cases, when the blur kernel varies across the image or when the noise spectrum is unknown, Wiener filtering may amplify high-frequency noise or generate ringing artifacts. Nevertheless, its simplicity, strong theoretical foundation, and computational efficiency make it a valuable benchmark for evaluating more advanced restoration methods.

In conclusion, Wiener Filter performs best when the blur kernel and noise parameters are known. However, it is susceptible to noise and produces ringing artifacts when these parameters are inaccurate.

2.2.2. *Deep learning based methods - MSCNN*

With the rise of convolutional neural networks, image deblurring has transitioned to a data-driven approach. A representative model is the Multi-Scale Convolutional Neural Network (MSCNN), which restores images from rough to fine resolutions. Seungjun Nah, Tae Hyun Kim and Kyoung Mu Lee proposed the Deep Multi-Scale Convolutional Neural Network (MSCNN) [6], a pioneering end-to-end framework for blind deblurring in dynamic scenes.

Unlike traditional optimization-based methods that rely on explicit blur kernel estimation, MSCNN directly restores a sharp image from a single blurry input without any prior knowledge of motion or camera parameters. The network adopts a multi-scale coarse-to-fine architecture, which mimics the structure of classical deblurring pipelines.

Each scale predicts a progressively sharper image, and a multi-scale loss function ensures that intermediate outputs at lower resolutions contribute to the final refinement. This hierarchical design allows the model to handle spatially varying and non-uniform motion blur effectively. MSCNN eliminates the need for kernel estimation, avoids ringing artifacts, and generalizes well to complex motion and dynamic scenes.

However, the model requires large paired datasets, extensive GPU training, and does not generalize well to unseen blur types without retraining. Furthermore, the multi-scale design increases computational cost, making it less practical for real-time applications.

In conclusion, it handles large motion blur effectively but requires extensive training data and high computational resources.

2.3. Downscaling

Downscaling is an important preprocessing step before feeding images into a computer vision classification model, as most modern neural architectures, including the one adopted in this work, require inputs of size 224×224. In contrast, raw images captured by smartphones and cameras are much larger. There are multiple approaches to downscale an image, and they can be broadly categorized as traditional interpolation, optimization-based methods, and deep learning-based methods.

2.3.1. *Traditional Interpolation*

Commonly used traditional interpolation methods include bilinear, bicubic, and Lanczos, each relying on a fixed convolution kernel. The kernel defines the weights assigned to pixels in a local neighborhood when computing each output value. The primary distinction between these methods lies in how the weights are determined. Bicubic interpolation computes each output pixel from a 4×4 pixel region to produce smooth results, while Lanczos uses a windowed sinc function over a larger domain, and yields sharper details and better suppression of aliasing in general. Importantly, once the kernel weights are set, they remain uniform throughout the image. The primary drawback of these approaches is their inability to adapt to diverse image content, which often results in the loss of structural detail, edge blurring, and the appearance of artifacts such as aliasing or ringing, particularly in areas with complex textures or sharp transitions.

2.3.2. *Deep Learning Methods*

To address these limitations, deep learning methods have been introduced, such as CNN-based models like the Scale-arbitrary Invertible Image Downscaling Network (AIDN) [7], Learned Scale-arbitrary Image Downscaling for Non-learnable Upscaling (SAID) [8], and GAN-based models like KernelGAN [9]. These approaches can learn to synthesize content-aware, spatially varying kernels for each image patch or region and allows for superior retention of details and textures compared to fixed-kernel interpolation. However, their effectiveness relies on training with a large, diverse dataset of annotated images, and their adaptability is restricted without retraining when new data distributions are encountered. While deep learning-based downscaling excels in generalization and can be seamlessly integrated into modern vision pipelines, the high demands for data and computation are significant trade-offs.

2.3.3. *Optimization-Based Methods*

The advantage of Optimization over Deep Learning methods is that they do not need large-scale training data, while allowing content-adaptive refinement. Instead of globally fixed kernels or dataset-driven learning, these algorithms iteratively adjust the value of each pixel, typically starting from the output of a traditional interpolation method. Examples include Content-Adaptive Image Downscaling [10], Perceptually Based Downscaling of Images [11], L0-regularized optimization [12], and Rapid, Detail-Preserving Image Downscaling (DPID) [13]. The goal is to optimize pixel intensities or local filtering operations to maximize criteria such as edge sharpness, structural similarity (SSIM), or the sparsity of image gradients, thereby enhancing the perceptual quality of each input individually. The strength of optimization-based downscaling lies in its capacity for image-specific adaptation and the practical preservation of relevant details. However, this comes at the cost of increased computational load, as the underlying iterative procedures are resource-intensive, especially for high-resolution content.

3. METHOD

The goal of this project is to investigate the impact of various image preprocessing techniques on the performance of a classifier in the presence of low-quality images. Our classifier is a SigLIP model fine-tuned on the Food-101 dataset, and we have a small dataset of low-quality images for evaluation. The images will first be preprocessed by our algorithms before being fed to the classification model.

3.1. Dataset

The dataset used in this study was designed to reflect the most common image degradation issues encountered when using a smartphone for food recognition. We collect the food image set consisting of various dishes—such as ramen, curry rice, salad, etc., captured under diverse conditions. These images simulate typical degradation scenarios found in mobile photography, including motion Blur, caused by hand-held camera shake or rapid shooting, which results in blurred edges and the loss of fine texture, and Low-Light and Noise, occurring in poorly lit restaurants where high ISO introduces random sensor noise and long exposure further exacerbates motion blur. These variations collectively create a challenging yet realistic dataset for evaluating image enhancement and deblurring algorithms. The purpose of this study is to analyze how different types of image degradation affect the performance of a computer vision model, and how classical enhancement algorithms can mitigate these effects.

3.2. Image Processing Algorithms

The following three subsections will include the algorithms we propose to use to preprocess our images in each category.

3.2.1. Low-light enhancement

To improve brightness and contrast in images for better classification performance, we will apply and compare three image processing algorithms: Gamma Transformation, CLAHE, and SSR. These algorithms were selected for their distinct strengths: Gamma Transformation provides efficient global intensity correction; CLAHE enhances local contrast while minimizing noise amplification; and SSR models illumination and reflectance to restore visual balance under uneven lighting. By integrating these enhancement methods into the preprocessing pipeline, our objective is to improve the ability of the classification model to extract discriminative features from dark or underexposed images, increasing precision under low-light conditions. The choice of classical, training-free algorithms ensures interpretability, reproducibility, and computational efficiency, making this enhancement stage a practical and explainable component of the overall classification system.

3.2.2. Image Deblurring

While existing deblurring research primarily targets motion-induced blur through complex kernel estimation or deep-learning-based restoration, these approaches are unsuitable for lightweight, real-world applications, such as mobile image classification. In contrast, this project focuses on improving overall image quality—including edge sharpness, contrast, and noise reduction—through simple, interpretable, and computationally efficient filtering techniques. Thus, three representative algorithms were implemented: the Simple Sharpening Kernel, Unsharp Masking (USM), and a Simplified Wiener Filter. The sharpening kernel operates in the spatial domain using a fixed high-pass convolution mask to emphasize edges and fine details, effectively improving perceived sharpness and local contrast.

The USM method enhances detail visibility by subtracting a blurred version of the image from the original and adding the weighted difference back, expressed as

$$I_{\text{USM}} = I_{\text{original}} + k(I_{\text{original}} - I_{\text{blurred}}) \tag{2}$$

Where k controls the enhancement strength, this approach allows greater flexibility than the fixed kernel and is particularly effective for improving texture-rich regions. The Wiener Filter, initially designed for frequency-domain deblurring, is implemented here in a simplified form as an adaptive denoising tool. Because the actual blur kernel and the signal-to-noise ratio are unknown in our real-world dataset, the complete deconvolution formula introduced in the related works[5] cannot be applied. Instead, the adaptive filtering capability of the simplified Wiener algorithm is used to locally estimate and reduce image variance, suppressing random noise while preserving overall structure.

The Simple Sharpening Kernel and Unsharp Masking (USM) were applied to restore edge definition in images that are blurred or have low contrast. At the same time, the Simplified Wiener Filter focuses on suppressing sensor noise in high-ISO or low-light conditions. For each image, three enhanced versions were generated and evaluated using the pretrained model trained on a clean dataset. Classification accuracy serves as the primary evaluation metric, directly reflecting the model's recognition ability rather than perceptual image quality.

Through this comparison, the analysis aims to determine which type of degradation results in a greater performance loss for our classification model.

3.2.3. *Downscaling*

Initially, we intended to compare three downscaling methods: Lanczos interpolation, AIDN, and CAID. However, after trying them out, we decided to drop AIDN and CAID for some reasons. AIDN required more than 5 GB on a GPU or more than 50 GB on a CPU, which was beyond what we had. As for CAID, we couldn't find a usable version on GitHub, which made it impossible for us to implement. We also looked into an L0-based optimization method, but it was only available in MATLAB, and integrating it with Python was unstable and filled with errors.

In the end, we selected three downscaling methods for our experiments:

1. Lanczos: This method directly reduces the size of the image using Lanczos interpolation to 224*224 pixels.

2. Lanczos_SAID: Since SAID is a model trained for specific scaling factors (×2 and ×4), using it without adjustment can cause issues like checkerboard patterns. To avoid this, we first resized the image to 448*448 pixels with Lanczos, then applied SAID to downscale it by a factor of ×2, resulting in the size of 224*224 pixels.

3. DPID: We chose DPID as our third method. It is an optimization-based algorithm that supports arbitrary downscaling factors and is easy to integrate. DPID resizes the image directly to 224*224 pixels.

To ensure that our experiments tested only the effect of the downscaling method, we kept all other preprocessing steps the same as in the original pipeline and disabled the built-in resize and center-crop options in the processor, so that the only difference between our baseline and the experiments is the downscaling method used.

4. EVALUATIONS

4.1. Experimental Setup

This section describes the tools, dataset, parameters, and evaluation protocol used in our experiments. Our pipeline consists of four major stages designed to evaluate the impact of different preprocessing techniques on food classification performance under low-quality imaging conditions.

4.1.1. *Experimental Configuration and Evaluation Criteria*

All experiments were conducted in Python using PyTorch, torchvision, and standard image-processing libraries, including OpenCV, scikit-image, PIL, and tlib. The classification backbone was the SigLIP Base Patch16×384 model pretrained on ImageNet. The full dataset comprised 205 food images, combining self-collected smartphone images that exhibit real-world degradations, such as motion blur from mild handshakes, low-light noise due to high ISO settings in dim restaurants, and downscaled low-resolution content, with high-quality online images that were degraded through synthetic augmentation. Since real-world blur kernels and noise statistics were unknown, all experiments were conducted in a non-blind setting without access to ground-truth clean images, which motivated the use of classical enhancement algorithms rather than kernel-based deconvolution. The primary evaluation metrics were Top-1 accuracy and standard percentage accuracy, both of which directly measure how different preprocessing techniques influence recognition performance under degraded imaging conditions.

4.1.2. *Phase I: Baseline Evaluation*

To check the model's baseline performance, we first used our low-quality images as the direct input to the classification model.

4.1.3. *Phase II: Individual Evaluation*

For each preprocessing category, the algorithms were tested independently. Each algorithm were applied individually to the images, and the resulting images were fed into the classifier to measure classification accuracy. The algorithm of each category that achieves the highest accuracy is then selected as the best-performing method.

Once the best algorithm from each category was found, these selected methods were combined in a single preprocessing pipeline. The classifier's performance on low-quality images with combined preprocessing was then compared with the baseline performance without preprocessing. This step allowed us to evaluate the benefit of sequentially applying the best algorithms across different preprocessing categories.

Example:

- Low-light enhancement: L1, L2, L3 $\rightarrow$ L2 performs best
- Deblurring: D1, D2, D3 $\rightarrow$ D1 performs best
- Downscaling: S1, S2, S3 $\rightarrow$ S2 performs best

Thus, the combined preprocessing pipeline becomes L2 + D1 + S2, which was applied prior to feeding the images into the base classifier.

4.1.4. *Phase III: Combination Search*

After identifying the best-performing algorithms in Part II, we performed an exhaustive search of all possible algorithm combinations across the three preprocessing categories. We looped through every combination of algorithms on the images and measured the accuracy for each combination. To fully explore the possible combinations, we also included cases where the algorithm was not used for low-light enhancement and deblurr. For example, with three algorithms per category, plus the unused cases, this would result in $4 \times 4 \times 3 = 48$ combinations.

The purpose of this phase was to determine whether a combination of algorithms other than the best ones could provide superior classification performance. The best combination from this search would be compared to the baseline model and the Phase I results to identify the most effective preprocessing strategy.

4.1.5. *Phase IV: Model Fine-Tuning on Low-Quality Images*

In the last part, the SigLIP model was fine-tuned directly on the low-quality image dataset. The performance of the fine-tuned model would be compared with the preprocessing pipelines identified in Phases I and II, as well as the base model trained on standard-quality images. This part would allow us to determine the effectiveness of model learning versus image preprocessing to improve classification performance on low-quality images.

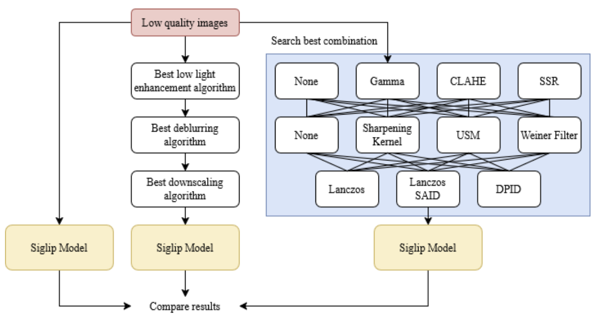


Fig. 1. Diagram of Part I to Part III, from left to right

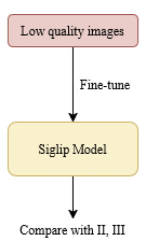


Fig. 2. Diagram of Part IV

4.2. Results and Comparison

After completing all experiments, the resulting accuracies were summarized in Table 1, which compared accuracies across preprocessing methods and experimental phases. The table presented the Top-1 accuracy achieved by each preprocessing algorithm across all four stages, which allowed us to directly observe how different enhancement strategies influence model performance under degraded imaging conditions.

First, among the low-light enhancement techniques, CLAHE achieved the accuracy by addressing contrast issues in local regions, resulting in a better detail enhancement. This enhancement technique helped the model classify images that were previously too dark to identify. On the other hand, gamma performed relatively well, while SSRetinex performed poorly. This was because all the images enhanced by SSRetinex were overexposed, which lowered the model's accuracy.

For image deblurring, the Simple Sharpening Kernel and Unsharp Masking both applied local edge enhancement with small spatial filters. Because the motion blur in our dataset spanned a much larger range than their kernel sizes, these methods could not meaningfully restore edges. As a result, they slightly reduced the accuracy compared to the baseline. Their similar results reflected the shared limitation that although they strengthen local contrast, they could not recover large-scale blur. The Simplified Wiener Filter performed the worst because it behaved as a noise-smoothing filter in the absence of known blur kernels. Instead of sharpening edges, it suppressed textures and high-frequency details that were essential for food classification and resulted in a larger drop in accuracy. Overall, all three deblurring methods underperformed the baseline, suggesting that classical spatial filtering is insufficient for real-world smartphone blur and noise, which are too complex for small-kernel or variance-based methods to correct.

In downscaling-related preprocessing, the Lanczos method gave results similar to the baseline. This makes sense because the Transformer used Bicubic resizing, and Bicubic and Lanczos work in similar ways. We also tested Lanczos_SAID and DPID. These would do better than Lanczos since they don't use a fixed kernel. Surprisingly, both did worse. Lanczos_SAID's results were about 2 percent lower than the baseline. This suggested that SAID, which had learned to create high-quality images, produced textures that were very close to those of Bicubic. In short, SAID's images looked natural and did not create many confusing artifacts. The small 2 percent gap could be due to SAID trying to reintroduce sharper edges, which were slightly different from the smoother Bicubic style the model was trained on. DPID did even worse, with a 7 percent drop from the baseline. DPID's method focused on sharpening edges and smoothing textures, but these changes were not what the model expected. While DPID images might appear better to people (clearer and sharper), for the model trained on softer, more natural images, this seems like noise, which lowers its performance.

In Part II, the best-performing individual algorithms from each preprocessing category were combined into a single pipeline: CLAHE (low-light enhancement) + SSK (deblurring) + Lanczos (downscaling). This combination resulted in an accuracy lower than that of the baseline model. This result suggests that, although each method may perform better within its own category, their sequential combination does not generate a cumulative benefit. Instead, the interaction between contrast enhancement, sharpening, and resampling may introduce additional artifacts that may reduce the classifier's ability to recognize food categories.

In Part III, we evaluated a more exhaustive set of preprocessing combinations. Each low-light method (three methods plus a no-enhancement option) was paired with each deblurring method (three methods plus a no-deblurring option), and each of these was further combined with one of the three downscaling algorithms, resulting in a total of 48 unique preprocessing configurations. Among the 48 possible preprocessing configurations, the best-performing combination was Gamma correction + Lanczos_SAID downscaling, leading to a result slightly higher than the baseline performance. This improvement, although modest, suggests that gamma correction can correct global illumination in a way that benefits classification, and Lanczos_SAID preserves high-frequency details more effectively than other downscaling methods. However, the gain is minimal, indicating that even the best classical preprocessing combination cannot significantly enhance recognition performance compared to using the original degraded images.

In Part IV, we fine-tuned the SigLIP model directly with the low-quality image dataset. It achieved the highest prediction accuracy by enabling the model to learn feature representations that are inherently robust to blur, noise, and illumination variations, as expected.

However, this improvement comes at the cost of higher computational cost and reduced generalization to unseen degradation types, as the model becomes specialized to the fine-tuned dataset. In contrast, classical enhancement techniques are model-agnostic and computationally lightweight, providing broader applicability despite their limited ability to recover fine details.

Phase	**Algorithm**	**Accuracy (%)**
Original Model		0.58
Low-light Enhancement	Gamma	0.58
	CLAHE	0.58
	SSRetinex	0.3
Deblurring	Sharpening Kernel	0.55
	Unsharp Masking	0.55
	Simplified Wiener	0.45
Downscaling	Lanczos	0.58
	Lanczos_SAID	0.55
	DPID	0.5
Part I		0.58
Part II		0.43
Part III		0.59
Part IV		0.91

Table 1. Accuracy comparison across preprocessing methods and experiment phases.

4.3. Discussion

Across all experiments, the results consistently show that classical preprocessing methods offer limited benefits for improving food classification under degraded imaging conditions. In Part I, the baseline model performed reasonably well despite the presence of blur and noise, establishing a strong reference point. In Part II, individual preprocessing algorithms rarely improved accuracy and, in many cases—especially in deblurring and downscaling—slightly reduced performance. Even the combined pipeline in Part II failed to outperform the baseline, suggesting that classical enhancement operations do not stack constructively. Part III expanded the search to 48 preprocessing combinations, yet only one configuration (Gamma + Lanczos_SAID) achieved a marginal improvement. In contrast, Part IV showed that fine-tuning the classifier directly on degraded images yields substantially higher accuracy.

The results were notably worse than our initial expectations had been. We identify several factors that likely contribute to the limited effectiveness of traditional preprocessing methods and explain why these approaches failed to improve classification accuracy.

4.3.1. 1. Mismatch in Training Distribution

One main reason the results did not meet our expectations is that the Siglip model was trained on natural, well-exposed, unenhanced images, compared to our low-quality datasets.

When we preprocess the images using our algorithms, such as gamma correction, CLAHE, sharpening kernels, and Wiener filters, although the preprocessed images may seem clearer to the human eye, the underlying pixel distributions, contrast, noise characteristics, and texture statistics still deviate significantly from the distributions observed during pretraining. Deep models were extremely sensitive to these shifts, so even "improved" images from a human perspective still produced poor results.

4.3.2. 2. Preprocessing Pipelines Compound Error

While each algorithm was tested individually, and achieved moderately good results, the combined pipeline did not aggregate those improvements. A possible reason is that one algorithm amplifies degradations in other aspects. For example, low-light enhancement and downscaling introduced distortions that deblurring cannot fix.

4.3.3. 3. Dataset Size

Since the dataset has only 205 images, small statistical changes in pixel distribution can cause disproportionately large swings in accuracy. This magnifies the negative effects of preprocessing and partially explains why the performance of some combinations drops significantly.

4.3.4. 4. Classical Enhancement Cannot Recover Lost Semantic Structure

Problems like blur, low light, or sensor noise can remove important details from your images. Older methods like CLAHE or sharpening may make the image look different, but they can't bring back lost information. They might boost contrast, but they don't restore the features the model needs.

4.3.5. 5. Over-Processing Can Be Detrimental

Another important observation from our experiments is that more preprocessing does not necessarily lead to better performance. In several cases, applying enhancement techniques to images that were not suffering from the corresponding degradation produced the opposite effect. For example, deblurring filters introduced noticeable high-frequency noise when applied to images that were already sharp. The noise was preserved or even amplified by subsequent steps such as downscaling, ultimately degrading the classifier's performance.

This phenomenon is reflected in the results of our pipeline search: The best-performing combination deliberately excludes deblurring, opting instead for the Identity operation. This suggests that avoiding unnecessary corrections is often more beneficial than aggressively improving image quality. When the original image is not severely degraded, additional processing can introduce distortions that push the data further from the model's pretrained distribution, leading to worse accuracy.

5. CONCLUSION

This study evaluated the effectiveness of traditional low-light enhancement, deblurring, and downscaling algorithms as preprocessing steps for food image classification under low-quality imaging conditions. Across individual algorithms and combined pipelines, preprocessing did not result in improvements and, in many cases, reduced accuracy to the baseline pretrained model. The results show that although these methods can improve the visual appearance of images, the resulting changes in pixel statistics and local textures are misaligned with the distributions on which vision models are trained. As a result, the enhancements frequently introduce distribution shifts that impair classification performance.

In contrast, fine-tuning the SigLIP model directly on the low-quality dataset produced a substantial performance increase, clearly outperforming all preprocessing strategies. This demonstrated that parameter adaptation is more effective than applying fixed image transformations when dealing with real-world degradations. Overall, the findings suggest that classical preprocessing offers limited benefits for modern deep classifiers, and that model-level adaptation is a more reliable approach for enhancing robustness to low-quality inputs, albeit at the expense of increased training costs.

6. FUTURE WORK

While our fine-tuning results worked exceptionally well for the model, this approach is unfortunately computationally expensive and model-agnostic. Therefore, we propose the following directions for future potential improvements.

First, we can try deep learning-based restoration models for image processing. Unlike classical algorithms, we can train modern low-light enhancement and deblurring networks to match the statistics that downstream classifiers expect.

Additionally, we could expand the dataset. The small size of the current dataset limits the generality of the evaluation. Expanding the low-quality image set would enable more reliable measurements and reduce sensitivity to slight variations in pixel statistics.

Apart from the quantity of samples, the quality of the samples is also worth improving. The current mix of real and synthetically degraded images may not fully capture natural noise, blur, and illumination patterns. Building a dataset with controlled real-world degradations may provide a more accurate sample.

7. LINK TO THE GITHUB PAGE

The complete codebase for this project can be found at:
https://github.com/vincent8264/ECE253-food-classification

8. REFERENCES

[1] Yi-Sheng Chiu, "Efficient contrast enhancement using adaptive gamma correction and cumulative intensity distribution," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, October 2011.

[2] P. Musa, "A review: Contrast-limited adaptive histogram equalization (clahe) methods to help the application of face recognition," in *Proceedings of the 2018 Third International Conference on Informatics and Computing (ICIC)*. IEEE, 2018, pp. 1–6.

[3] Mahdi Mirzapour, Ali Keshavarz Nasab, Amir Movafeghi, and Effat Yahaghi, "Retinex theory based automated contrast enhancement of gamma radiographic images of pipe welds," *Journal of Nondestructive Evaluation*, vol. 44, 06 2025.

[4] C. Li, "A survey on image deblurring," in *Proceedings of the International Conference on Image and Signal Processing*. IEEE, 2022, pp. 1–6.

[5] P. Bojarczak and Z. Lukasik, "Image deblurring – wiener filter versus tsvd approach," *Advances in Electrical and Electronic Engineering*, vol. 6, no. 2, pp. 85–91, 2007.

[6] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2016, pp. 3883–3891.

[7] Jinbo Xing, Wenbo Hu, and Tien-Tsin Wong, "Scale-arbitrary invertible image downscaling," *arXiv preprint arXiv:2201.12576*, 2022.

[8] Chengrui Huang, Wanjie Sun, and Zhenzhong Chen, "Learned scale-arbitrary image downscaling for non-learnable upscaling," *IEEE Signal Processing Letters*, 2023.

[9] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani, "Blind super-resolution kernel estimation using an internal-gan," *arXiv preprint arXiv:1909.06581*, 2019.

[10] Johannes Kopf, Ariel Shamir, and Pieter Peers, "Content-adaptive image downscaling," *ACM Transactions on Graphics*, vol. 32, no. 6, Nov. 2013.

[11] A. Cengiz Öztireli and Markus Gross, "Perceptually based downscaling of images," *ACM Transactions on Graphics*, vol. 34, no. 4, Aug. 2015.

[12] Junjie Liu, Shengfeng He, and Rynson W. H. Lau, "L0-regularized image downscaling," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1076–1085, Mar. 2018.

[13] Nicolas Weber, Michael Waechter, Sandra C. Amend, Stefan Guthe, and Michael Goesele, "Rapid, detail-preserving image downscaling," *ACM Trans. Graph.*, vol. 35, no. 6, Dec. 2016.