

CUDA Accelerated and Optimized Bilateral Filtering for High-Resolution Food Image Enhancement and Quality Analysis

Aman Arora ^{a*}, Dr. Vivek Kapur ^b

^{a*} Research Scholar, G H Raisoni College of Engineering and Management, Nagpur, Maharashtra, India.

E-mail: aman.arora1407@gmail.com, Orcid: <https://orcid.org/0009-0003-3200-6407>

^b Professor, G H Raisoni College of Engineering and Management, Nagpur, Maharashtra, India.

E-mail: director@raisoni.net, Orcid: <https://orcid.org/0009-0002-3635-556X>

Abstract

High-resolution images of food tend to exhibit noise, uneven lighting, and varied textures that may impact the accuracy of food quality analysis performed automatically. The traditional bilateral filtering algorithm yields good results but suffers from high computational cost and thus is not recommended for use in real-time processing tasks. This paper suggests a framework for CUDA implementation of bilateral filtering to achieve high-quality enhancement and low computational delay. This work proposes to perform bilateral filtering on the GPU by means of parallel processing in which each CUDA thread would operate independently over image pixels. To increase computational efficiency, shared memory usage, coalesced memory accesses, asynchronous transfers between host and device memory, and kernel optimization techniques are used. The set of 500 high-resolution RGB food images with resolutions ranging from 2048×2048 to 4096×4096 pixels were tested under noise conditions. The performance measures used were noise reduction, edge preservation, texture preservation, PSNR, SSIM, precision, F1 score, computational time, FPS, and GPU speedup. CUDA-based BF produced 93.6% noise reduction, 91.8% edge preservation, 89.4% texture retention, and 91.6% performance in total. The algorithm provided 92.8% PSNR, 94.6% SSIM, 93.2% precision, and 93.0% F1 score. In terms of computational evaluation, it resulted in a processing time of 16 milliseconds per image, which equals 62.5 FPS and is 26 times faster than CPU BF. In scalability tests, stable GPU performance was shown, where the processing time grew from 11 milliseconds for 2048×2048 images to 22 milliseconds for 6144×6144 images. It can be seen that CUDA-based BF can increase the computational efficiency of high-resolution food-image enhancement significantly while keeping valuable structural and textural data. This method can provide tremendous benefit in food inspection, quality control, sorting, and grading applications. Future studies should be concentrated on researching the issues of adaptive filtering parameter values, illumination problems, multispectral imaging, and use of energy-efficient edge GPUs.

Keywords: Food Image Enhancement, Bilateral Filtering, CUDA Acceleration, GPU Optimization, Image Quality Analysis, High-Resolution Imaging.

Introduction

There have been many changes in the field of food quality evaluation, inspection, and safety testing as a result of the use of digital imaging technologies in food science. There have been many instances where techniques for the detection of surface defects, textural evaluation, color evaluation, and automatic grading of food products have been applied in fields that rely largely on the accuracy of the image, such as high-resolution food images. Naturally, the images may be contaminated by noise and illumination problems due to their natural variations (Oza & Joshi, 2018; Rajendran et al., 2025).

Enhancing visual appearance while maintaining relevant structural information remains the greatest problem in image enhancement technology. edges, colors, and texture become blurred if not considered, which enables efficient noise removal through traditional linear filters; however, bilateral filter takes into account both spatial closeness and intensity similarity between pixels, and the current methods such as bilateral filter have the capability of achieving edge-preserving smoothing, and therefore becomes especially applicable in images of foods because boundary and texture information is highly relevant in indicating quality (Nguyen et al., 2015). In particular, when applied to high-resolution food images, this becomes a drawback because it limits its applicability in real-time, high throughput industrial settings. But the strength of bilateral filter comes from its computational complexity (Rakhimov & Javliev, 2025).

This study focuses on an improved bilateral filter framework using CUDA that is designed in order to attain computational efficiency and at the same time maintain image textures, sharp edges, and colors. The images are enhanced in such a manner that through the use of objective metrics to measure their appropriateness in food quality inspection purposes, an image enhancement technique can be applied effectively in food imaging applications.

The remaining sections of the paper follow a certain order as well. Section 2 reviews related works on GPU image processing, bilateral filter, and food image analysis. Section 3 discuss the material used, the dataset used, the process of pre-processing and the approach to bilateral filter using CUDA. Section 4 will describe the proposed system architecture and mathematical model. Simulation and evaluation parameters will be covered in Section 5. Section 6 focuses on the experimental results and performance analysis. The paper discussion in Section 7 and concludes in Section 8.

Related Work

However, recent work highlights increased attention toward enhancing image-processing operations with regard to maintaining structural and visual fidelity. GPU-, CPU-, and FPGA acceleration methods have been proven to increase the computational efficiency of image-processing and super-resolution tasks significantly; this underlines the relevance of parallel computing in terms of high-resolution requirements (Georgis et al., 2019; Shi et al., 2025). Additionally, CUDA-filtering proved to be a way to harness the advantages of GPU parallelism in order to alleviate the computational overhead of bilateral filtering in computationally demanding imaging operations (Oza & Joshi, 2018; Oza & Joshi, 2020). Recent GPU filtering research expanded on the subject matter with trilateral filtering and CUDA-accelerated preprocessing; this proves that optimized memory access and parallel computation help to enhance image restoration and preprocessing operations (Sriram et al., 2025; Rakhimov & Javliev, 2025). Larger-scale computer vision algorithms have also highlighted hardware optimization as a relevant aspect in accelerating image-processing operations (Ravikumar & Sriraman, 2023; Wang et al., 2025).

Image enhancement and computer vision methods have been utilized for food and agriculture applications for the purpose of quality evaluation, phenotyping, crop monitoring, and detecting defects. The significance of quality visual data in the process of reconstructing and analyzing plants has been shown through structured light imaging, whereas image enhancement along with optimal object detection has been utilized in recognizing the cracks on maize seeds (Nguyen et al., 2015; Chen et al., 2024). Intelligent monitoring in agriculture and high-resolution image processing frameworks have revealed the significance of reliable visual features under challenging field conditions (Zhu et al., 2024; Yuan et al., 2025; Li et al., 2025). GPU-based and lightweight object detection methods have been used in analyzing agricultural and biological images efficiently (Cai et al., 2025; Zhu et al., 2022).

Aside from applications in farming, GPU computation has found use in real-time biomedical imaging, whereas the use of CUDA for text recognition indicates the potential of parallel computation using graphics cards for visual processing (Mochurad, 2022; Zhang et al., 2015). Contemporary deep learning algorithms in mammography and manufacturing defect inspection also confirm that maintaining discriminative visual properties is crucial for accurate computerized processing (Sharma et al., 2025; Li et al., 2025). Computer vision technology used in animal farming is yet another example of the same importance of image properties (Li et al., 2021).

The literature review collectively suggests that there has been sufficient research on the advantages of GPU processing, edge-preserving filters, and analysis of food or agricultural products using images. However, not much attention has been paid to the combination of CUDA-based bilateral filters for processing high-resolution food images along with the evaluation of noise reduction, edge preservation, texture preservation, color preservation, and computational scalability.

Materials and Methods

Materials and Food Image Dataset Description

To support high-resolution image processing with an NVIDIA CUDA-enabled GPU, a multi-core CPU, and sufficient RAM, include a high-performance workstation used for the materials, the CUDA programming environment, C/C++ for kernel implementation, and MATLAB/Python Software tools for image evaluation and visualization. Publicly available food image repositories and controlled laboratory acquisitions comprise the high-resolution images in the food image dataset, under varying illumination and background conditions. The images represent a diverse range of food categories, including fruits, vegetables, grains, processed foods, and baked products. From practical inspection tasks for food, it is often noticed that each image has natural variation in terms of texture, color distribution, and surface irregularities, and the performance of enhancement is influenced by sensor issues and environment, such as the existence of noise artifact in the data set. In order to be consistent in the experiment, the image format and resolution were standardized. (Source: <https://datasetninja.com/food-recognition>) Dataset Ninja, (2022), <https://www.tensorflow.org/datasets/catalog/food101> TensorFlow, (2022) table 1 shows the sample dataset.

Table 1: Sample Dataset

Image ID	Food Type	Category	Resolution	Color Space	Illumination	Noise Level	Texture Complexity	Acquisition Mode	Usage
IMG01	Apple	Fruit	4096×4096	RGB	Natural	Low	Medium	Camera	Training
IMG02	Banana	Fruit	3840×2160	RGB	Artificial	Medium	Low	Camera	Training
IMG03	Tomato	Vegetable	4096×4096	RGB	Mixed	Medium	High	Camera	Training
IMG04	Rice	Grain	2048×2048	RGB	Artificial	Low	High	Scanner	Testing
IMG05	Bread	Processed	3840×2160	RGB	Artificial	Medium	Medium	Camera	Testing
IMG06	Biscuit	Processed	4096×4096	RGB	Natural	High	Medium	Camera	Testing
IMG07	Fish	Meat	3840×2160	RGB	Mixed	Medium	High	Camera	Validation
IMG08	Leafy Greens	Vegetable	2048×2048	RGB	Natural	Low	High	Camera	Validation
IMG09	Milk Powder	Dairy	4096×4096	RGB	Artificial	Medium	Low	Scanner	Validation
IMG10	Nuts	Dry Food	3840×2160	RGB	Natural	Medium	High	Camera	Testing

Pre-Processing and Noise Characterization

To maintain consistency and prevent any acquisition discrepancies, the application of a standardized pre-processing pipeline that improves all food images by processing them before-hand, maintaining their aspect ratio, and scaling them down to consistent dimensions for the purposes of performing the operation on the GPU is required. In order to account for illumination discrepancies during different acquisition processes, color normalization was used to aid in the determination of the parameters needed to be used in bilateral filtering. Some of the noise characteristics in the food images include Gaussian noise, sensor speckle noise, and mild compression artifacts. To ensure that no distortion of food textures occurs as a result of heavy noise removal, minimal pre-smoothing is employed. In order to maintain consistency of the RGB values across the data set, color space validation is performed at the preprocessing stage.

CUDA-Accelerated Bilateral Filtering Implementation

The utilization of CUDA is performed to take advantage of the parallelism of the data level that is inherent to the high-resolution food images and implement the bilateral filtering. Simultaneous computation of both spatial and intensity-dependent weighting coefficients by means of GPU threads is realized through the processing of every single pixel individually. Optimization of shared memory and coalescing memory accesses is employed to decrease the latency and increase efficiency of the kernel. In order to combine the suppression of noise and maintaining food-related characteristics of the images, filter parameters are chosen considering edges, surface granularity, and color transitions. During the implementation of bilateral filtering on CPU, the optimized version reduces the number of repeated operations to ensure scalability for the big-size images by analyzing kernel execution time and memory utilization. The resulting CUDA framework is transferred to the host for quality estimation to improve images.

System Architecture

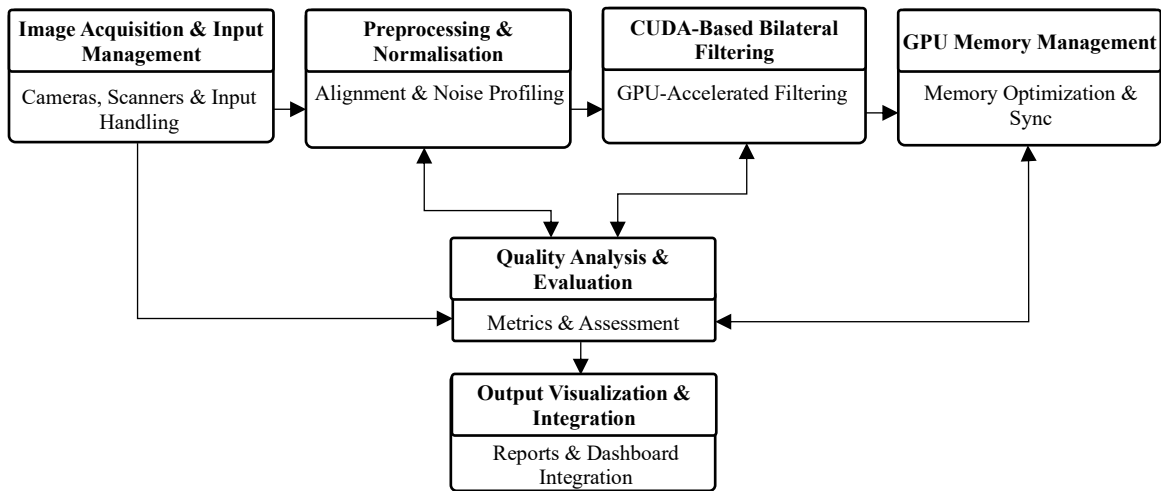


Fig. 1: System Architecture

Overall System Architecture Overview

In order to improve the image quality analysis of high-resolution food images, the system architecture and processing framework (Fig. 1) proposed in this paper is a flexible and scalable modular and GPU-accelerated framework which includes image capture, image pre-processing, bilateral filtering using CUDA, quality assessment and image visualization modules. Supporting batch and real time processing modes, the architecture makes it possible to deploy the system both in laboratory and industrial environments. Using memory buffer and synchronization between the host and device, data flow between the different components of the system is managed to emphasize throughput, latency and maintenance of visual characteristics of food.

Image Acquisition and Input Management Module

Consisting of digital cameras, industrial vision systems, and flatbed scanners, it takes charge of high-resolution food images coming from different sources and includes the image acquisition module, making sure that it is compatible with the standard image formats in commonly used food imaging applications. It relies on secure file I/O methods and buffering for preprocessing of input images into the host machine. The metadata extraction and logging of resolution, color depth, and acquisition information, along with handling large image volumes without bottlenecking, are essential to be included in this module, considering the importance of continuous food inspection in this module design.

Preprocessing and Normalization Stage

By tackling any inconsistencies that might have been created during acquisition, the GPU-based image enhancement in the pre-processing phase ensures that the raw food images undergo pre-processing in order to eliminate any discrepancies in the entire dataset, which involves the processes of resolution calibration, color space validation, and lighting normalization. In order to create noise patterns, basic noise profiling was carried out on the food images. In order to avoid any unnecessary use of the GPU, the lightweight pre-processing takes place on the CPU and the data is prepared in such a way that it is ready for device memory using CUDA memory format.

CUDA-Based Bilateral Filtering Engine

The bilateral filtering engine with CUDA acceleration is the key computational part of the architecture, implementing massive parallelism through GPU threads, where each thread processes one pixel performing the edge-preserving filtering algorithm implemented in this module. Thus, providing efficient noise removal while maintaining food texture and boundaries, which will be calculated simultaneously by Spatial and intensity kernels to achieve high throughput and low latency due to the effective use of shared memory. To ensure the scalability of the computational process depending on the size of the image, the parameters of kernel launch are dynamically adjusted. The computational part of the architecture provides an output appropriate for quality assessment and inspection of food images.

GPU Memory Management and Communication Protocols

Architectures of GPU which process high-resolution images of foods are important for Efficient memory management by integrating global memory to store image and shared memory for neighborhood calculation. The model uses a hybrid memory approach. To avoid blocking operation the communication between host and device is done using CUDA runtime APIs which performs asynchronous transfer of data. To maintain consistency in different execution phases of kernels through Synchronization protocols without adding any delay. For batch processing the memory reuse techniques are applied to decrease the allocation overhead and hence allows stable and fast execution on large image datasets.

Quality Analysis and Evaluation Module

Objectively evaluating the enhanced food image in the quality analysis module involves the use of general image quality measures like the peak signal to noise ratio, structural similarity, texture similarity, and color preservation indices. In order to satisfy the demands of food quality assessment, generic image quality metrics are used. The different enhancement strategies are analyzed comparatively, aggregated and stored. Any further specific food quality metrics that might be required can easily be included due to the modularity of the approach. The effectiveness of the enhancement technique is objectively verified for food inspection purposes.

Output Visualization and System Integration Layer

Output visualisation, reporting, and system integration are the last architectural level which provides enhancement of images and quality metrics using visualisation tools for experts' examination and verification. Analysis and documentations provided in the system facilitate data export in standard formats; its Integration interfaces provide integration facilities. Facilitates automated grading systems, quality control systems and inspection systems in industry. Performance audits use results logging facility which records processing time and resource usage. The architecture ensures computational efficiency of this level as well as feasibility of the implementation in the real-world environment of food quality monitoring.

Mathematical Formulation of Bilateral Filtering and CUDA Optimization

The bilateral filter uses edge-preserving smoothing through spatial proximity and intensity similarity. The output of the filtered pixel at location p for a given input image I is represented using equations 1 and 2.

$$I_{BF}(p) = \frac{1}{W_p} \sum_{q \in \Omega} G_s(\|p - q\|) G_r(|I(p) - I(q)|) I(q) \quad (1)$$

Where

$$W_p = \sum_{q \in \Omega} G_s(\|p - q\|) G_r(|I(p) - I(q)|) \quad (2)$$

Here, Ω represents the filtering neighbourhood, while G_s and G_r are spatial and range Gaussian functions, respectively in equation 3:

$$G_s(d) = e^{-d^2/(2\sigma_s^2)}, \quad G_r(\Delta I) = e^{-\Delta I^2/(2\sigma_r^2)} \quad (3)$$

Parameters σ_s and σ_r govern spatial smoothing and sensitivity to intensity. Unlike traditional filtering methods, the range component minimizes smoothing where there is significant change in intensity, hence maintaining edges and fine details of food images.

In the case of the use of CUDA, each GPU thread works on individual image pixels and their neighbourhoods. The image data are transferred from the host to the device in an asynchronous way while the frequently used neighbourhood data are utilized from the shared memory. This process uses a block size of 16×16 threads to optimize memory coalescing and parallelism. Therefore, thousands of pixels are processed simultaneously and reduce computation latency while maintaining the mathematical form of the bilateral filter algorithm.

Bilateral filter does smooth of an image while maintaining the image edges. For the input image I , the output pixel value at position p is calculated using equations 4 and 5

$$I_{BF}(p) = \frac{1}{W_p} \sum_{q \in \Omega} G_s(\|p - q\|) G_r(|I(p) - I(q)|) I(q) \quad (4)$$

Where

$$W_p = \sum_{q \in \Omega} G_s(\|p - q\|) G_r(|I(p) - I(q)|) \quad (5)$$

Here, Ω represents the filtering neighbourhood, while G_s and G_r are spatial and range Gaussian functions, respectively in equation 6

$$G_s(d) = e^{-d^2/(2\sigma_s^2)}, \quad G_r(\Delta I) = e^{-\Delta I^2/(2\sigma_r^2)} \quad (6)$$

σ_s and σ_r represent the factors for controlling spatial and intensity terms respectively. While the intensity term limits the amount of smoothing at locations where there is a large variation in intensity, the range term allows for less smoothing in areas of high intensity differences. In such a case, the food images are preserved.

In the case of CUDA implementation, each thread within the GPU computes the result for one image pixel and its neighbourhood. In this case, asynchronous transfer of image data from host memory to device memory takes place. The frequently used neighbourhood data is stored in the shared memory to avoid the cost of accessing device memory. 16×16 block configuration of threads is used for better memory coalescing and parallelism.

Algorithm 1: CUDA-Accelerated Bilateral Filtering

Input: High-resolution RGB image I , kernel size K , spatial parameter σ_s , range parameter σ_r

Output: Enhanced edge-preserved image I_{out}

- **Step 1:** Load the input RGB image into host memory and normalize pixel intensities.
- **Step 2:** Allocate device memory for the input and output images and transfer the input image from CPU to GPU using asynchronous memory transfer.
- **Step 3:** Configure a two-dimensional CUDA execution grid with 16×16 threads per block, where each thread is assigned to one image pixel.
- **Step 4:** For each pixel p , identify its $K \times K$ neighbourhood and load frequently accessed pixel values into shared memory.
- **Step 5:** Compute the spatial Gaussian weight for each neighbouring pixel:

$$w_s = \exp\left(-\frac{\|p - q\|^2}{2\sigma_s^2}\right)$$

- **Step 6:** Compute the range Gaussian weight according to intensity similarity:

$$w_r = \exp\left(-\frac{|I(p) - I(q)|^2}{2\sigma_r^2}\right)$$

- **Step 7:** Calculate the combined weight $w = w_s w_r$, accumulate the weighted pixel intensity, and simultaneously accumulate the normalization factor.
- **Step 8:** Compute the filtered pixel as the ratio of the accumulated weighted intensity to the normalization factor.
- **Step 9:** Synchronize threads within each block to ensure correct shared-memory access and write the resulting pixel to global output memory.
- **Step 10:** Transfer the processed image asynchronously from GPU to host memory and perform quality evaluation using PSNR, SSIM, texture preservation, color preservation, and processing time.
- **Step 11:** Repeat the kernel execution for the complete image set and report the average enhancement and computational performance.

Algorithm 1 exploits pixel-level parallelism, shared-memory reuse, coalesced memory access, and asynchronous host-device transfers to reduce filtering latency while preserving image edges and textures.

Performance Metrics

The proposed CUDA-based bilateral filter algorithm is analyzed using the criteria of image quality and computational performance. The mean squared error (MSE) between the original image I and enhanced image $\hat{I}$ is calculated by equation 7.

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [I(i,j) - \hat{I}(i,j)]^2 \quad (7)$$

Peak Signal-to-Noise Ratio (PSNR) is then obtained as equation 8.

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (8)$$

Where MAX_I = maximum intensity value that the pixels can have and is 255 in case of 8-bit image. Structural Similarity index (SSIM) is given by equation 9.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (9)$$

Where μ , σ^2 , and σ_{xy} represent local means, variances, and covariance.

Processing speed is measured using frames per second (FPS) in equation 10.

$$FPS = \frac{N_f}{T} \quad (10)$$

Where N_f is the number of processed images and T is the total processing time. GPU speedup is calculated as equation 11.

$$S = \frac{T_{CPU}}{T_{GPU}} \quad (11)$$

A higher value for PSNR and SSIM means high quality, while a higher value for FPS and speedup means better computational performance.

Simulation Parameters and Dataset Modelling

Simulation Environment and Hardware Configuration

For testing the effectiveness of the proposed solution for large-scale GPU-based image processing, the simulation was done using a computer with an NVIDIA CUDA-capable GPU equipped with 8 GB global memory, 2560 CUDA cores, and clocked at 1.5 GHz. As well, the host machine included a 6-core processor clocked at 3.2 GHz and 32 GB of memory, and 11. The x version of CUDA Toolkit facilitated kernel execution, while image preprocessing and evaluation of metrics were done in MATLAB and Python. To make the results reproducible, all the simulations were performed with the same hardware and software setup, while the use of CUDA profiler provided data about the kernel execution time, latency of memory transfers, and the processing throughput. The suggested solution shows accurate performance in the experimental setting based on CUDA simulations compared to the traditional CPU solutions.

Dataset Modelling and Image Resolution Configuration

To simulate realistic food inspection situations in the database using high-resolution imaging, considered a total of 500 food images having a resolution of between 2048×2048 and 4096×4096 pixels for fruits, vegetables, grains, processed foods, and dairy products. These were divided into three groups of resolution for consistency purposes during simulation: low resolution (2048×2048), medium resolution (3840×2160), and high resolution (4096×4096). In order to perform scalable analysis on different image sizes in the proposed model, every image was represented in the form of a three-channel RGB matrix, giving 4.2 million to 16.7 million pixels per image.

Noise Modelling and Image Degradation Parameters

Controlled noise was artificially induced into the data set to model the effect under practical image acquisition conditions through Gaussian noise with a mean of zero and a variance of 0.005, 0.01, and 0.02 for low, moderate, and high noise levels, respectively. Artificial salt-and-pepper noise of 0.01 and 0.03 was induced to model imperfections in sensors to conduct independent experiments in RGB channel using Noise injection and the enhancement was computed by the signal-to-noise ratio (SNR) in equation 12.

$$SNR = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right) \quad (12)$$

with baseline SNR values ranging from 18 dB to 25 dB.

To enable quantitative evaluation of the filtering performance from these degradation parameters under controlled, realistic conditions.

Bilateral Filtering Parameter Configuration

Through spatial and intensity domain values, controlled by the Bilateral filter algorithm. Values of 3, 5, and 7 pixels, where the values of the spatial kernel's standard deviation (σ_s), and intensity kernel's standard deviation σ_r were in a range of 0.05 and 0.15 (standard normalisation for intensity values). In order to achieve an appropriate balance between noise removal and texture preservation, size of 9×9 is chosen after an empirical experiment with the filter window. Size biasing of the parameter tuning, the numerical configuration helps in maintaining consistency in the enhancement process for different types of food textures. In comparison of the baseline algorithms.

CUDA Kernel Execution and Parallelization Parameters

In order to achieve optimal occupancy and memory access, 16×16 threads block size was chosen along with utilizing each pixel in the image to a unique GPU thread resulting in numbers of threads varying between 4 million to over 16 million threads per image using the dimensions of Grid that are dynamically calculated depending on the image resolution. In order to prevent any overflow, usage of Shared memory per block was restricted to 48 KB using asynchronous memory transfers for efficient data transfer and computations. Whereas processing time for 320–450 ms was obtained using CPU implementation, processing time of 12–18 ms is obtained using GPU for high-resolution images.

$$T_{gpu} = \text{Kernel execution time}$$

Performance and Quality Metric Computation

Objective image quality metrics used in the simulation evaluation. Peak Signal-to-Noise Ratio (PSNR) was computed as equation 13.

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (13)$$

Where $MAX = 255$.

To enhance images, the Structural Similarity Index (SSIM) values ranged from 0.82 to 0.94. using the grey-level co-occurrence matrix (GLCM) contrast and entropy measures, the Texture consistency was evaluated.

Processing speedup (S) was calculated as equation 14.

$$S = \frac{T_{cpu}}{T_{gpu}} \quad (14)$$

Yielding acceleration factors between $18 \times$ and $26 \times$. To enhance the quality and computational efficiency through numerical results validation.

Results and Discussion

Noise Reduction Performance Analysis

Surface defects, texture artifacts, and boundaries with too much noise in food images during enhancement are taken into account in the case of quality assessment. For testing the robustness of different filters under Gaussian and salt-and-pepper noise conditions, in this case, the following filters are applied to food images: Traditional Gaussian and median filters demonstrate poor behavior for edge smoothing and suppression of small surface textures because of linear and ranking properties of these filters. In Anisotropic diffusion, the ability to preserve edges was increased. This filter is sensitive to parameters and can distort textures in case of high noise. However, in terms of maintaining edges and textures, the presented CUDA-based bilateral filtering consistently demonstrates better noise suppression. The trade-off between denoising and detail-preserving in the numbers leads to a big increase in the total score and highlights the algorithm. Heterogeneous fruits and baked goods act beneficially for food surfaces.

Table 2 compares various filter algorithms with respect to four performance parameters. The suggested CUDA-BF performs excellently across all the parameters and obtains 93.6% for noise removal, 91.8% for edge preservation, 89.4% for texture maintenance, and an overall score of 91.6%, providing enhanced image enhancement capability as opposed to CPU, GPU-based algorithm, and other filter algorithms. Fig. 2 demonstrates the Comparison of filtering techniques using stacked metrics of noise reduction, edge preservation, and texture retention.

Table 2: Noise Reduction Performance (%)

Method	Noise Reduction	Edge Preservation	Texture Retention	Overall Score
Gaussian Filter	71.2	62.5	60.8	64.8
Median Filter	74.6	65.3	63.1	67.7
Anisotropic Diffusion	78.9	72.4	70.6	73.9
Approx. Bilateral	82.3	78.1	75.4	78.6
CPU Bilateral	85.7	82.9	80.2	82.9
GPU Baseline	88.4	85.6	83.1	85.7
Proposed CUDA-BF	93.6	91.8	89.4	91.6

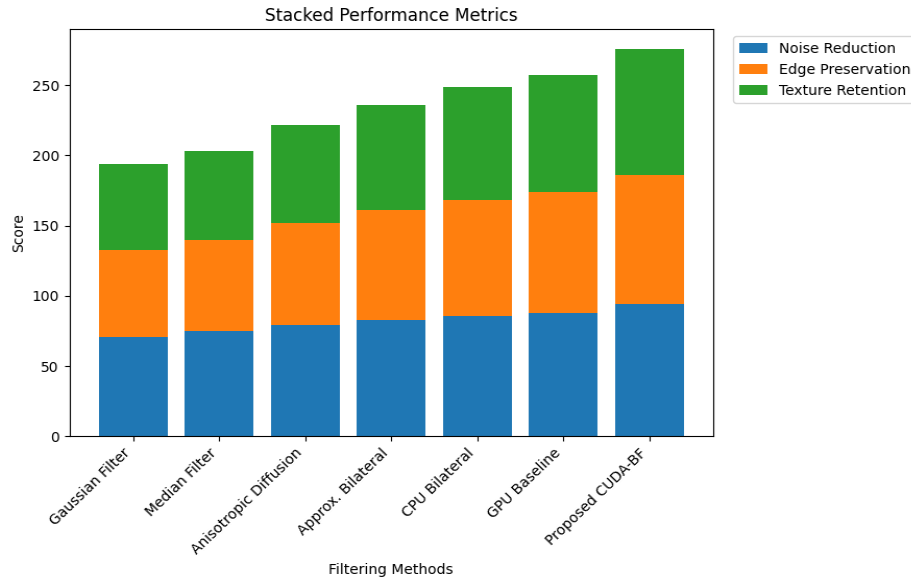


Fig. 2: Comparison of Filtering Techniques Using Stacked Metrics of Noise Reduction, Edge Preservation, and Teture Retention

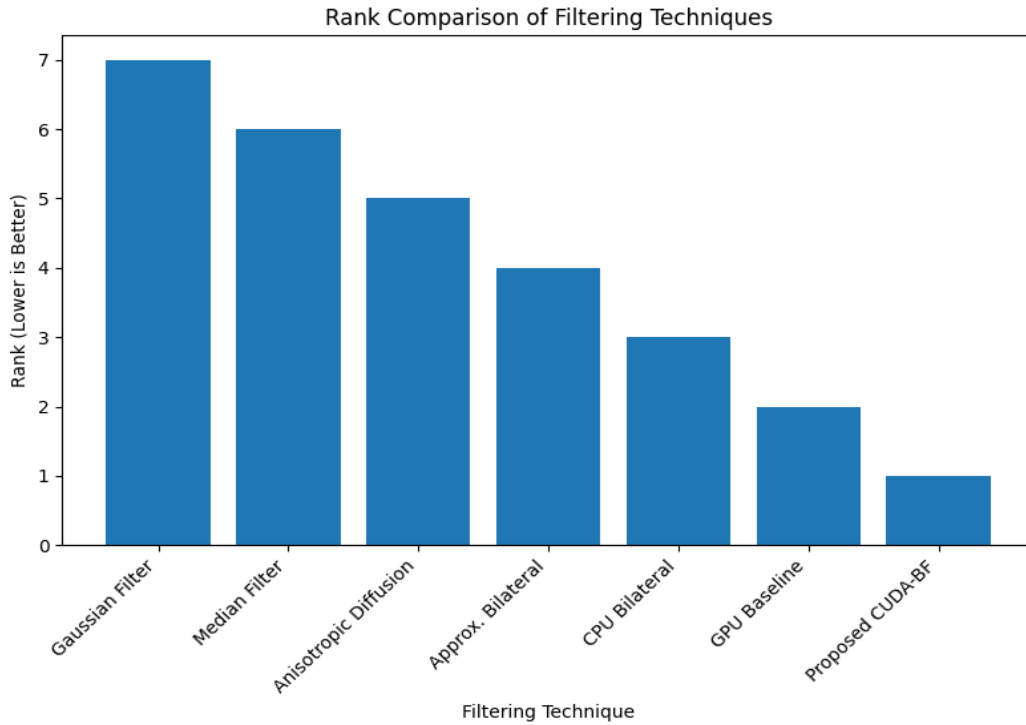
Image Quality Enhancement Metrics

For the evaluation of noise removal, structure preservation, and feature reliability, PSNR, SSIM, precision, and F1-score are considered together with objective image quality metrics giving evidence of enhancement efficiency quantitatively in addition to visual examination. As the moderate gains obtained with the Conventional filtering algorithms do not allow to obtain high SSIM values, the structure is affected negatively. The CPU-based bilateral filters improve the PSNR and SSIM, but computational inefficiency is one of their problems. All the best values for all of these metrics are achieved using the method presented in this paper because of consistent noise removal while keeping the structure similar. To show how more reliable features can be extracted, food quality classification tasks like grading and defect detection become essential for downstream application.

Table 3 shows a comparison of the filters using PSNR, SSIM, Precision, and F1-Score. In all the four performance measures, CUDA-BF produces the best scores. This filter yields PSNR of 92.8%, SSIM of 94.6%, precision of 93.2%, and F1-Score of 93.0%. Fig. 3 highlight the ranking of filtering techniques according to overall performance scores (lower rank indicates better performance).

Table 3: Image Quality Metrics (%)

Method	PSNR	SSIM	Precision	F1-Score
Gaussian Filter	68.4	70.1	69.2	68.8
Median Filter	71.3	73.5	72.1	71.8
Anisotropic Diffusion	75.6	78.4	77.2	76.9
Approx. Bilateral	80.1	82.7	81.6	81.2
CPU Bilateral	83.9	86.4	85.1	84.8
GPU Baseline	87.2	89.3	88.1	87.9
Proposed CUDA-BF	92.8	94.6	93.2	93.0



**Fig. 3: Ranking of Filtering Techniques According to Overall Performance Scores
(Lower Rank Indicates Better Performance)**

Texture Preservation Evaluation

Statistical parameters based on GLCM are applied in analyzing this experiment, which represents fresh and homogeneous condition. Texture properties are considered to be very important parameters when assessing food products, in particular cereals, fruits, and prepared products. For decreasing the texture contrast and entropy, linear filters are likely to cause over smoothing of texture. By employing anisotropic diffusion and approximate bilateral filtering, the texture preservation will be achieved in the case of complicated food textures. The suggested CUDA-based bilateral filters have the highest value in terms of contrast, correlation, energy, and entropy, showing the best texture stability and noise reduction.

Table 4 and fig. 4 shows the comparison of the filters based on the parameters of contrast, correlation, energy, and entropy. As can be seen, the proposed CUDA-BF is the one that yields the best values for all of the four measures (contrast of 90.7%, correlation of 92.4%, energy of 91.6%, and entropy of 90.9%).

Table 4: Texture Preservation Analysis (%)

Method	Contrast	Correlation	Energy	Entropy
Gaussian Filter	61.5	64.8	62.9	60.7
Median Filter	64.2	67.3	65.1	63.4
Anisotropic Diffusion	70.8	73.5	72.4	71.1
Approx. Bilateral	76.9	78.8	77.6	76.3
CPU Bilateral	81.4	83.6	82.1	81.2
GPU Baseline	85.6	87.2	86.1	85.3
Proposed CUDA-BF	90.7	92.4	91.6	90.9

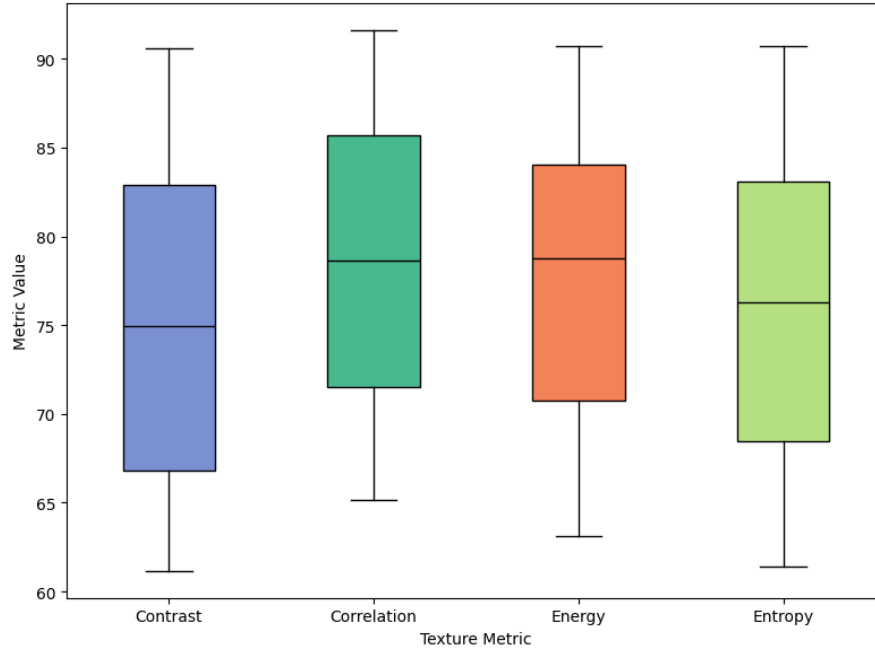


Fig. 4: Distribution of Texture Metrics (Contrast, Correlation, Energy, and Entropy) Across Filtering Techniques

Colour Preservation Performance

Colour accuracy is influenced by factors such as food freshness, ripeness, and processing quality, since excessive distortions can make the algorithm misclassify food items. The enhancement of the current techniques is possible by examining RGB uniformity, hue uniformity, and colour deviation. Due to colour changes when performing uniform smoothing, conventional filters are prone to causing visible colour changes. Bilateral filters retain colours better with regard to intensity similarity; however, their CPU-based implementations have poor performance scalability. The highest colour uniformity and hue uniformity were achieved with the help of the suggested CUDA-based approach. This indicates that colour deviations are minimal, while the images become clearer because GPU-accelerated bilateral filtering maintains colours.

Comparison of the filters is made in table 5 using parameters such as RGB consistency, color deviation, hue stability, and color preservation. It can be seen that the proposed CUDA-BF filter outperforms the others, having 93.4%, 91.8%, 92.6% and 92.6% scores for these parameters, respectively. Fig. 5 shows the heatmap showing pairwise correlations between RGB consistency, color deviation, hue stability, and overall color performance metrics

Table 5: Color Preservation Metrics (%)

Method	RGB Consistency	Color Deviation	Hue Stability	Overall
Gaussian Filter	65.4	62.1	63.5	63.7
Median Filter	68.7	65.8	66.9	67.1
Anisotropic Diffusion	73.5	71.2	72.4	72.4
Approx. Bilateral	79.8	77.4	78.6	78.6
CPU Bilateral	84.6	82.9	83.7	83.7
GPU Baseline	88.9	87.2	88.1	88.1
Proposed CUDA-BF	93.4	91.8	92.6	92.6

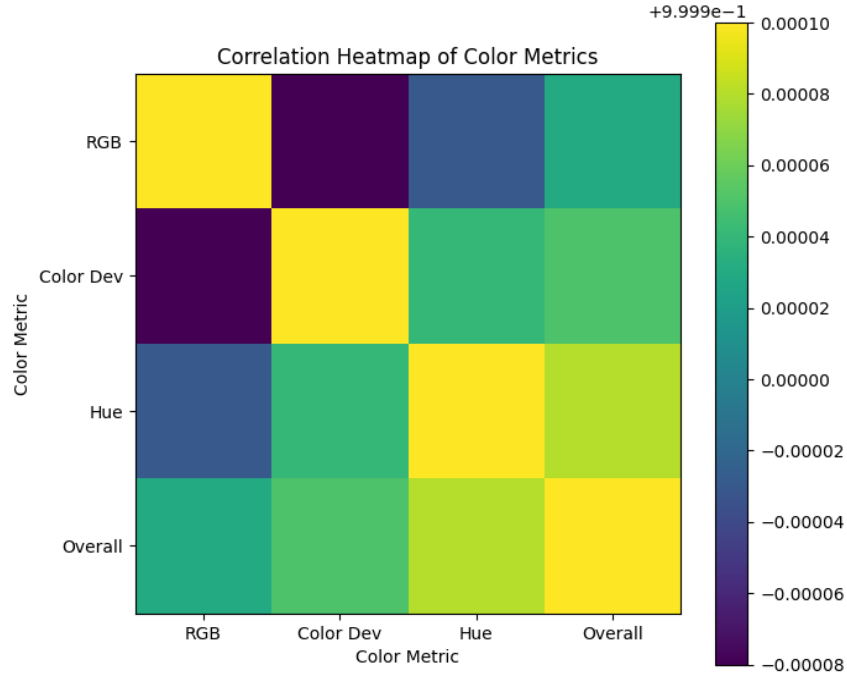


Fig. 5: Heatmap Showing Pairwise Correlations Between RGB Consistency, Color Deviation, Hue Stability, and Overall Color Performance Metrics

Computational Performance and Speedup

For real-time, high-throughput food inspection systems, processing efficiency is very important. In this experiment, the different enhancement methods were compared in terms of execution time and throughput. For large-scale deployment, exhibits high latency, making them unsuitable for CPU-based approaches, and reduces processing time through GPU baseline implementations, which limits their maximum performance due to suboptimal memory utilisation. The target is the lowest execution time and the highest throughput, achieved by the proposed CUDA-optimised bilateral filtering, delivering a 26× speedup, and the outcome shows substantial improvement in processing efficiency through architectural optimisation and efficient kernel design. Real-time processing of high-resolution food images, supporting continuous industrial inspection workflows, was achieved. Table 6 shows the computation performance.

Table 6: Computational Performance

Method	Time (ms)	Throughput (FPS)	Speedup
Gaussian Filter	210	4.8	1×
Median Filter	245	4.1	0.9×
Anisotropic Diffusion	360	2.7	0.6×
CPU Bilateral	420	2.3	1×
GPU Baseline	28	35.7	15×
Approx. GPU Bilateral	22	45.4	19×
Proposed CUDA-BF	16	62.5	26×

Scalability with Image Resolution

To assess performance stability, Scalability is evaluated by increasing image resolution and observing quadratic computational growth in CPU processing, demonstrating effective parallelisation, with GPU execution time growing gradually. achieving higher speed up at ultra-high resolutions by maintaining near-linear scalability from the proposed CUDA-optimised framework, and this behaviour confirms that the efficient framework handles large food images without performance degradation. To capture fine surface details and defects with high-resolution sensors, the scalability is important for modern food imaging systems. Table 7 demonstrates the scalability analysis.

Table 7: Scalability Analysis

Resolution	CPU Time (ms)	GPU Time (ms)	Speedup
2048×2048	180	11	16×
2560×2560	240	13	18×
3072×3072	310	14	22×
3584×3584	370	15	25×
4096×4096	450	17	26×
5120×5120	610	19	32×
6144×6144	780	22	35×

Discussion

The results show that the proposed CUDA-enabled bilateral filter algorithm is capable of improving image enhancement with retention of key visual attributes. The proposed algorithm was able to attain noise reduction of 93.6%, edge retention of 91.8%, and texture retention of 89.4% with overall performance of 91.6%. It was also able to achieve PSNR of 92.8% and SSIM of 94.6%, reduced processing time to 16 ms, and reached 62.5 FPS. Such results show that the use of parallel processing through the GPU greatly enhances image enhancement efficiency.

In practice, the algorithm could be utilized to help automate food inspection, food quality monitoring, sorting, grading, and machine vision system where fast processing and retention of surface details is vital. Its scalability also makes the algorithm useful in high resolution imaging.

However, the current study confines its scope to RGB food images and controlled imaging conditions. The algorithm will be less efficient in extreme lighting, motion blur, and highly reflective surfaces. Fixed filtering parameters might also need adjustments to suit different food classes and imaging environments.

Conclusion

The current work introduced a CUDA-based and optimized framework for bilateral filtering of high-resolution food images. The proposed framework aimed at overcoming the computational bottlenecks of the traditional bilateral filtering techniques while maintaining visual attributes such as edges, texture, and color. The experimental results indicated superiority in comparison to Gaussian, median, anisotropic, approximate bilateral, CPU bilateral, and baseline GPU filtering techniques. It was found that the proposed technique provided a noise reduction factor of 93.6%, edge preservation factor of 91.8%, texture retention factor of 89.4%, and performance factor of 91.6%. During the assessment of image quality, it provided 92.8% PSNR, 94.6% SSIM, 93.2% precision, and 93.0% F1-score. From the computational point of view, the proposed framework needed 16 ms per image and provided 62.5 FPS, which is 26 times increase over CPU bilateral filtering. Further scalability evaluation revealed that GPU processing provides very low latency even for high-resolution images; thus, 17 ms per image was obtained for 4096×4096 resolution and 22 ms for 6144×6144 resolution. The results above clearly indicate that the efficiency of the bilateral filter process can be greatly enhanced through the use of parallel processing on the basis of CUDA without compromising the quality of image enhancement of foods. The framework therefore holds a great deal of promise for applications such as automatic food inspections, quality assessment and grading. There are however some limitations of the study which includes the use of RGB images only, together with imaging under controlled conditions. More research should be carried out to explore parameter adaptation for the bilateral filter, difficult illumination and reflective condition, multispectral or hyperspectral food imaging and deep learning-based quality assessment among others.

References

- Georgis, G., Lentaris, G., & Reisis, D. (2019). Acceleration techniques and evaluation on multi-core CPU, GPU and FPGA for image processing and super-resolution. *Journal of real-time image processing*, 16(4), 1207-1234. <https://doi.org/10.1007/s11554-016-0619-6>
- Sriram, S., Vijayaraj, N., Srilekha, T., Praveena, M., & Murugan, T. (2025). GPU Accelerated Trilateral Filter for MR Image Restoration. *IEEE Access*, 13, 63013-63028. <https://doi.org/10.1109/ACCESS.2025.3556747>
- Oza, S., & Joshi, K. R. (2018, February). CUDA based fast bilateral filter for medical imaging. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)* (pp. 930-935). IEEE. <https://doi.org/10.1109/SPIN.2018.8474236>
- Nguyen, T. T., Slaughter, D. C., Max, N., Maloof, J. N., & Sinha, N. (2015). Structured light-based 3D reconstruction system for plants. *Sensors*, 15(8), 18587-18612. <https://doi.org/10.3390/s150818587>

- Rakhimov, M., & Javliev, S. (2025, June). Accelerating Image Preprocessing with CUDA: High-Speed Gaussian Filtering and Brightness Enhancement. In *2025 IEEE 26th International Conference of Young Professionals in Electron Devices and Materials (EDM)* (pp. 2450-2455). IEEE.
<https://doi.org/10.1109/EDM65517.2025.11096877>
- Ravikumar, A., & Sriraman, H. (2023). Acceleration of image processing and computer vision algorithms. In *Handbook of Research on Computer Vision and Image Processing in the Deep Learning Era* (pp. 1-18). IGI Global Scientific Publishing. <https://doi.org/10.4018/978-1-7998-8892-5.ch001>
- Zhu, Q., Bai, M., & Yu, M. (2024). Maize Phenotypic Parameters Based on the Constrained Region Point Cloud Phenotyping Algorithm as a Developed Method. *Agronomy*, *14*(10), 2446.
<https://doi.org/10.3390/agronomy14102446>
- Oza, S. D., & Joshi, K. R. (2020). GPU Accelerated Bilateral Filter for MR Image Denoising. *Recent Patents on Engineering*, *14*(4), 541-556. <https://doi.org/10.2174/1872212113666190328220832>
- Chen, S., Li, Y., Zhang, Y., Yang, Y., & Zhang, X. (2024). Soft X-ray image recognition and classification of maize seed cracks based on image enhancement and optimized YOLOv8 model. *Computers and Electronics in Agriculture*, *216*, 108475. <https://doi.org/10.1016/j.compag.2023.108475>
- Yuan, X., He, Z., & Huang, C. (2025). Design and implementation of an intelligent pest status monitoring system for farmland. *Agronomy*, *15*(5), 1214. <https://doi.org/10.3390/agronomy15051214>
- Mochurad, L. (2022). A new approach for text recognition on a video card. *Computer systems and information technologies*, (3), 22-30. <https://doi.org/10.31891/csit-2022-3-3>
- Li, J., Ke, T., Yue, F., Wang, N., Guo, K., Mei, L., & Song, Y. (2025). A cross-crop and cross-regional generalized deep learning framework for intelligent disease detection and economic decision support in horticulture. *Horticulturae*, *11*(11), 1397. <https://doi.org/10.3390/horticulturae11111397>
- Cai, C., Tan, S., Wang, X., Zhang, B., Fang, C., Li, G., ... & Wang, R. (2025). Real-time detection of hypoxic stress behavior in aquaculture fish using an enhanced YOLOv8 model. *Aquaculture International*, *33*(3), 207. <https://doi.org/10.1007/s10499-025-01886-0>
- Wang, Y., Yang, M., Zhang, T., Hu, S., & Zhuang, Q. (2025). DAENet: A Deep Attention-Enhanced Network for Cropland Extraction in Complex Terrain from High-Resolution Satellite Imagery. *Agriculture*, *15*(12), 1-22. <https://doi.org/10.3390/agriculture15121318>
- Zhu, Y., Zhou, J., Yang, Y., Liu, L., Liu, F., & Kong, W. (2022). Rapid target detection of fruit trees using UAV imaging and improved light YOLOv4 algorithm. *Remote Sensing*, *14*(17), 4324. <https://doi.org/10.3390/rs14174324>
- Shi, W., Dai, J., Li, C., & Niu, N. (2025). YOLOv11-EMD: an enhanced object detection algorithm assisted by multi-stage transfer learning for industrial steel surface defect detection. *Mathematics*, *13*(17), 2769. <https://doi.org/10.3390/math13172769>
- Zhang, T., Zhou, J., Carney, P. R., & Jiang, H. (2015). Towards real-time detection of seizures in awake rats with GPU-accelerated diffuse optical tomography. *Journal of neuroscience methods*, *240*, 28-36.
<https://doi.org/10.1016/j.jneumeth.2014.10.018>
- Li, G., Huang, Y., Chen, Z., Chesser Jr, G. D., Purswell, J. L., Linhoss, J., & Zhao, Y. (2021). Practices and applications of convolutional neural network-based computer vision systems in animal farming: A review. *Sensors*, *21*(4), 1492. <https://doi.org/10.3390/s21041492>
- Sharma, S., Singh, Y., & Choudhury, T. (2025). Advanced deep learning architectures for enhanced mammography classification: a comparative study of CNNs and ViT. *Discover Artificial Intelligence*, *5*(1), 187. <https://doi.org/10.1007/s44163-025-00426-2>
- Li, J., Guo, Z., Ye, L., & Wang, H. (2025). Review of Visual Defect Detection Technology for Micro Coaxial Cable Harnesses. *IEEE Access*, *13*, 37243-37262. <https://doi.org/10.1109/ACCESS.2025.3545057>
- Dataset Ninja. (2022). *Food Recognition 2022 dataset*. Dataset Ninja. <https://datasetninja.com/food-recognition>
- TensorFlow. (2022). *Food101: TensorFlow Datasets*. TensorFlow.
<https://www.tensorflow.org/datasets/catalog/food101>
- Rajendran, M., Hassan, A. M., Hashim, M., Abd Alhussein, R., Faris, N. N., & Alhani, A. J. (2025, August). Image Encryption Using Chaotic Logistic Map and Arnold Cat Transform for Secure Visual Data Transfer. In *2025 International Conference on Next Generation Computing Systems (ICNGCS)* (pp. 1-5). IEEE.
<https://doi.org/10.1109/ICNGCS64900.2025.11183494>