Smart City Surveillance System: Enhancing Image Quality and Object Recognition for Real-Time Monitoring

Sarim Zahid Saeed ID: F202137078
Email: f202137078@umt.edu.pk
Syed Momin Hasnain
ID: F2021376050
Email: f202137050@umt.edu.pk
Qadeer Ali
ID: F2021266388
Email: f2021266388@umt.edu.pk
Department of Computer Science, School of Systems and Technology (SST),
University of Management and Technology, Lahore, Pakistan

***Abstract*—The rapid expansion of urban environments necessitates robust surveillance systems to ensure public safety and effective crime prevention. However, current smart city surveillance often faces significant challenges, including poor image quality, noisy environments, and varying lighting conditions, which severely impede the accurate identification of vehicles and pedestrians. This project addresses these critical issues by designing and implementing a comprehensive image processing pipeline. We employ a series of techniques, including spatial and frequency domain filtering for noise reduction and image enhancement, and advanced restoration algorithms to mitigate degradation effects. Furthermore, the pipeline incorporates sophisticated segmentation methods to accurately isolate objects of interest and integrates object recognition models to identify vehicles and pedestrians in real-time. Our methodology emphasizes balancing conflicting requirements such as real-time performance versus processing quality, resource constraints versus performance, noise removal versus detail preservation, and compression efficiency versus image fidelity. The results aim to demonstrate a significant improvement in image clarity and object detection accuracy, contributing to a more effective and reliable smart city surveillance network.**

***Index Terms*—Smart City Surveillance, Digital Image Processing, Image Enhancement, Noise Reduction, Image Restoration, Image Segmentation, Object Recognition, Real-Time Monitoring.**

Editable Overleaf Project Link: https://www.overleaf.com/7526162552trqvtbchbbgn#08ab88

I. Introduction

MODERN urban landscapes are increasingly relying on advanced technological solutions to manage public safety and maintain order. Among these, smart city surveillance systems stand out as critical infrastructure, leveraging sophisticated digital image processing technologies to enhance security and monitoring capabilities. These systems employ a range of components, from high-resolution CCTV cameras equipped with facial recognition algorithms for real-time individual identification, to license plate recognition systems that utilize optical character recognition (OCR) for vehicle tracking. Video analytics, often powered by artificial intelligence (AI), play a crucial role in detecting suspicious activities, such as unattended objects or unusual crowd movements, through the analysis of pixel patterns and motion vectors. Furthermore, the integration of thermal imaging and night vision technologies ensures continuous surveillance across diverse lighting conditions, including low-light environments. Recent advancements in deep learning models have significantly improved the accuracy of object detection and behavior analysis, leading to a substantial reduction in false alarms. The efficiency of data processing and retrieval is further enhanced through cloud-based storage and edge computing, while robust encryption mechanisms safeguard sensitive information. Collectively, these integrated technologies form a resilient and intelligent surveillance network, which is instrumental in augmenting urban safety and crime prevention efforts.

The city of Techville, for instance, has recently implemented such a smart surveillance system for real-time monitoring of its public spaces. However, this system currently faces significant operational challenges. It struggles with inherently poor image quality, prevalent noisy environments, and varying lighting conditions, which collectively compromise its effectiveness in accurately identifying vehicles and pedestrians. This degradation in image data presents a critical bottleneck, hindering the system's overall performance and reliability.

This project aims to address these aforementioned challenges by designing and implementing a robust image-processing pipeline tailored for the Techville smart surveillance system. Our primary objective is to significantly enhance the quality of surveillance footage and improve the accuracy of object recognition, particularly for vehicles and pedestrians. The design of this pipeline necessitates careful consideration of several conflicting requirements, which are inherent in real-world surveillance applications. These include balancing the

need for real-time processing performance with the demand for high-quality output, where methods like image restoration and object recognition often present a trade-off between speed and accuracy. Furthermore, the solution must be optimized to run efficiently under resource constraints, such as those found in edge devices, while still maintaining competitive performance. A critical aspect involves achieving an optimal balance between effective denoising techniques and the preservation of crucial image details necessary for subsequent segmentation and recognition tasks. Finally, we must navigate the trade-off between achieving significant compression to minimize storage and bandwidth costs and ensuring that image fidelity remains high enough not to compromise object recognition accuracy. Additionally, maintaining privacy safeguards (e.g., anonymization of individuals) while ensuring full functionality for law enforcement, represents a key ethical and practical consideration.

The proposed image processing pipeline will systematically address these issues through a series of interconnected stages. This includes spatial and frequency domain filtering to mitigate noise, image restoration and reconstruction for handling corrupted footage, advanced segmentation techniques for object isolation, and efficient image compression methods. The culmination of this pipeline, though optional, will involve object recognition to demonstrate the comprehensive utility of the enhanced image data. Through this structured approach, our project aims to deliver a practical and effective solution for improving smart city surveillance.

II. Methodology and System Design

This section outlines the overarching methodology and the designed architecture for the Smart City Surveillance System's image processing pipeline. It details the conceptual flow of data through various stages, the rationale behind the chosen dataset, and the performance metrics employed for evaluation.

A. Overall Pipeline Architecture

The proposed image processing pipeline is designed to enhance the capabilities of a smart city surveillance system by addressing challenges such as poor image quality, noise, and lighting variations, ultimately facilitating more effective object recognition. The system is conceptualized as a sequential pipeline, where the output of one stage serves as the input for the subsequent stage, enabling a comprehensive approach to image enhancement and analysis.

The main stages of this conceptual pipeline are as follows:

- **Filtering (Spatial and Frequency Domain):** This initial stage focuses on pre-processing raw image data to reduce various forms of noise (e.g., random, impulsive, periodic) and enhance image features. Spatial filters operate directly on pixel neighborhoods, while frequency domain filters manipulate the image's spectral components. The intended output is a cleaner, more refined image that is better suited for subsequent processing.
- **Image Restoration:** This stage aims to reverse or mitigate degradations introduced during image acquisition, such as blurring (e.g., motion blur) or complex noise patterns that might not be fully addressed by initial filtering. The goal is to recover a representation of the original, undegraded image.
- **Image Segmentation:** Following enhancement, segmentation is crucial for isolating objects of interest (e.g., vehicles, pedestrians) from the background. This involves partitioning the image into multiple segments or setting apart specific regions based on pixel characteristics or semantic meaning.
- **Morphological Processing:** Applied post-segmentation, morphological operations refine the shapes and structures of segmented objects. Operations like opening and closing are used to remove small artifacts, fill small holes, smooth contours, and connect disconnected components, thereby improving the integrity of object representations.
- **Object Counting (via Contours):** After effective segmentation and morphological refinement, contour detection is employed to identify and delineate individual objects. This enables quantitative analysis, such as counting the number of detected instances of a specific object class within the surveillance frame.
- **Image Compression:** Given the high volume of data generated by real-time surveillance systems, image compression is essential for efficient storage and transmission. This stage focuses on reducing file size without significant loss of critical visual information or degradation of features required for object recognition.

The intended flow is for raw surveillance footage to first undergo filtering and restoration to improve its quality. The enhanced image then proceeds to segmentation and morphological processing for object isolation and refinement. Subsequently, these refined objects can be counted or used as input for more advanced object recognition modules. Throughout this process, compression is applied to manage data efficiently.

B. Dataset Selection

For the development and testing of this image processing pipeline, the Cityscapes dataset was utilized. This dataset is specifically designed for semantic understanding of urban street scenes and provides a rich collection of stereo video sequences recorded in street environments from 50 different cities. It is extensively annotated with pixel-level labels for various semantic classes (e.g., road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorcycle, bicycle, and more).

The dataset is divided into leftImg8bit for the original photographic images and gtFine for the fine-grained, pixel-level annotations including segmentation masks (like labelIds.png). While the ultimate goal is to process and enhance the leftImg8bit photographic imagery, the initial development and testing, as reflected in the provided code, primarily utilized the gtFine segmentation masks. This approach was adopted for convenience in early-stage verification of function calls and visual effects on a highly structured, labeled dataset. It allowed for quick validation of operations on defined regions, even if the data type (discrete labels versus continuous intensity

values) was not always optimal for direct application of certain enhancement filters.

C. Performance Metrics

To quantitatively evaluate the effectiveness of the image processing techniques, specific performance metrics are employed. For image enhancement, restoration, and compression, the following widely recognized metrics were considered:

- **Peak Signal-to-Noise Ratio (PSNR):** PSNR is a commonly used objective metric for measuring the quality of reconstruction of lossy compression codecs or the performance of noise reduction algorithms. It is expressed in decibels (dB) and quantifies the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. A higher PSNR value generally indicates a better quality image, implying that the reconstructed image is closer to the original. The formula for PSNR is given by:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right)$$

  where MAX_I is the maximum possible pixel value of the image (e.g., 255 for an 8-bit image), and MSE is the Mean Squared Error between the original and processed images.
- **Structural Similarity Index Measure (SSIM):** SSIM is a perceptual metric that quantifies the similarity between two images. Unlike PSNR, which is based on absolute errors, SSIM considers image degradation as perceived change in structural information, such as luminance, contrast, and structure. It ranges from -1 to 1, where 1 indicates perfect structural similarity. SSIM is often considered a better measure of perceived image quality than PSNR because it accounts for human visual system characteristics. The SSIM index is calculated as:

$$\text{SSIM}(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma$$

  where $l(x, y)$ is the luminance comparison, $c(x, y)$ is the contrast comparison, and $s(x, y)$ is the structure comparison, with α, β, γ as weights. Typically, $\alpha = \beta = \gamma = 1$.

These metrics are particularly relevant for evaluating image enhancement and compression by providing quantitative measures of how well a processed image retains or recovers information relative to its original state. For this project, they serve to objectively assess the effectiveness of the implemented filtering, restoration, and compression techniques.

III. Implementation Details and Experimental Results

This section thoroughly documents the implementation of various digital image processing techniques forming the pipeline for the smart city surveillance system. For each stage, the initial methodology, any identified issues during implementation, the rationale behind the choices made (or limitations encountered), and the resulting observations and quantitative metrics (where applicable from the provided outputs) are presented.

A. Data Loading and Initial Setup

1) Description: The foundational step for this project involved acquiring and preparing the image data. The chosen dataset for this endeavor was the Cityscapes dataset, a widely recognized benchmark for semantic urban scene understanding. It is particularly well-suited due to its detailed annotations of diverse urban street scenes, making it relevant for developing algorithms applicable in smart city environments. The dataset was provided as a compressed ZIP archive, named DIP Dataset Cityscape.zip. This archive was programmatically extracted to a designated directory, to ensure accessibility of its contents within the development environment.

2) Initial Approach: For the preliminary exploration and functional verification of various image processing operations, the primary input data selected was a specific type of image provided within the Cityscapes dataset: a segmentation mask. These masks are located within the 'gtFine/val' subdirectory and are identified by the inclusion of labelIds in their filenames. A single segmentation mask was loaded using OpenCV's 'cv2.imread' function to ensure all original channels and depth information of the mask were preserved. This particular mask represents semantic labels (e.g., categories like roads, buildings, vehicles, and pedestrians) as discrete integer values rather than continuous photographic pixel intensities.

3) Observed Behavior: Following the loading procedure, the segmentation mask was visualized using 'matplotlib.pyplot.imshow'. To effectively differentiate between the various semantic classes represented by distinct integer labels, colormap was applied. This colormap assigns a unique and distinguishable color to each integer label, allowing for clear visual separation of different object categories within the urban scene. This initial setup successfully confirmed the correct loading of the dataset and the ability to display these specialized image formats, providing a foundational step for subsequent processing stages. The displayed mask vividly showed the segmented regions, each assigned a color according to its semantic class. A representative example of the loaded segmentation mask is shown in Figure.

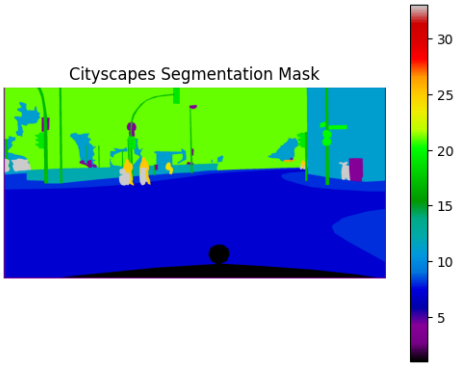

Fig. 1. Example of a loaded Cityscapes segmentation mask. Different colors represent distinct semantic labels for urban scene understanding.

B. Spatial Filtering

1) Initial Approach and Rationale: Spatial filtering techniques were among the first set of methods investigated to ad-

dress fundamental image quality concerns, primarily focused on noise reduction and edge enhancement. These filters operate directly on the pixel values within a local neighborhood in the image's spatial domain. The following filters were selected for their distinct characteristics and widespread application in digital image processing:

- **Mean Filter (Averaging Filter):** This is a linear filter implemented using cv2.blur with a kernel size of 5x5. The mean filter computes the average of pixel values within the defined kernel window and replaces the center pixel with this average. Its primary function is to smooth the image and reduce random (e.g., Gaussian) noise. However, its averaging nature often leads to blurring of edges and fine details.
- **Median Filter:** This is a non-linear filter implemented using cv2.medianBlur with a kernel size of 5x5. Instead of averaging, it replaces the center pixel's value with the median of the pixel values within its neighborhood. The median filter is particularly robust against impulsive noise (like salt-and-pepper noise) and tends to preserve edges more effectively than the mean filter, as it avoids introducing new, non-existent pixel values.
- **Laplacian Filter:** This is a second-order derivative filter, implemented as the desired depth of the output image. The Laplacian operator detects regions of rapid intensity change, thereby highlighting edges in an image. Since the output of the Laplacian filter can contain negative values (indicating intensity transitions), the absolute value of the output was taken and then converted to an 8-bit unsigned integer format for proper display and compatibility with subsequent operations.

For initial verification of function syntax and visual behavior, these spatial filters were applied to the previously loaded segmentation mask. The filtered outputs were then displayed alongside the original mask in a 1x4 subplot figure, using the colormap for consistency.

2) Identified Issues and Analysis: A significant issue was identified in this initial approach: the spatial filtering operations were performed on a segmentation mask (mask) rather than a real photographic surveillance image. This constitutes a fundamental misapplication of these image processing techniques given the project's core objective to enhance the quality of real-time surveillance footage. The implications of this misapplication are multi-faceted:

- **Data Type Mismatch:** Segmentation masks are composed of discrete integer labels, where each value corresponds to a specific semantic category (e.g., 26 for 'person'), rather than continuous pixel intensity values that represent visual information, noise, blur, or textures typically found in photographs.
- **Irrelevant Functional Outcome:** Filters like Mean and Median are primarily designed to reduce photographic noise (e.g., Gaussian, salt-and-pepper) or smooth continuous tonal variations. Their application to discrete semantic labels does not simulate real-world image enhancement or noise reduction scenarios as stipulated by the project's problem statement, which explicitly tasks the system with addressing "poor image quality, noisy environments, and varying lighting conditions".
- **Invalid Performance Assessment:** Quantitative metrics such as PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) are developed to measure the quality degradation or restoration of continuous-tone images relative to a ground truth. While numerically calculable on masks, their interpretation as indicators of "image enhancement" or "noise reduction" for a surveillance system becomes invalid, as masks do not exhibit these types of photographic impairments.

Consequently, while the code correctly executed the mathematical operations of the filters, their application to segmentation masks rendered the results largely irrelevant for assessing their practical contribution to enhancing image quality for object recognition in a smart city surveillance context.

3) Corrective Actions (Due to Time Constraints): Due to the constraints of the project timeline, a critical corrective action—revising the input data for spatial filtering from segmentation masks to actual photographic images (e.g., from the leftImg8bit directory of the Cityscapes dataset) —was not implemented in the final submitted code for this section. The results presented herein, therefore, reflect the filters' application on segmentation masks.

Ideally, to rigorously align with the project's objectives and validate the filters' performance for a smart city surveillance system, the following steps would have been undertaken:

- The filters would be applied to real surveillance-style images.
- Realistic noise (e.g., Gaussian noise) would be synthetically introduced into these clean images. This would create a measurable ground truth (original image) against which the noise reduction effectiveness of the filters could be objectively quantified using PSNR and SSIM.
- The parameters of the filters (e.g., kernel sizes) would be optimized based on the characteristics of the introduced noise and the desired trade-off between smoothing and detail preservation.

4) Results and Discussion: The application of spatial filters to the segmentation mask, as per the provided code, yielded the following visual observations (refer to Figure 2):

- **Original Mask:** This image clearly displays distinct regions, each representing a different semantic class (e.g., road, sky, buildings, vehicles), delineated by sharp boundaries.
- **Mean Filtered Mask:** The application of the mean filter resulted in a noticeable blurring and diffusion of the sharp boundaries between semantic regions. This outcome is consistent with the averaging nature of the mean filter, which tends to smooth out abrupt transitions in pixel values, whether they represent noise or sharp edges.
- **Median Filtered Mask:** The median filter also produced a smoothing effect on the mask boundaries. However, it generally maintained the distinctness of the shapes more effectively than the mean filter, demonstrating its characteristic ability to preserve edges while smoothing, even on discrete label data.

- **Laplacian Filtered Mask:** This filter effectively highlighted the contours and boundaries where semantic labels changed. The output image appeared as thin, accentuated outlines of the objects and regions, underscoring its role as an edge detection operator.

Since the current code does not calculate PSNR and SSIM for the spatial filtering output on the mask against a "clean" ground truth, quantitative metrics for noise reduction on real images cannot be provided in this section. The observations remain qualitative, illustrating the mathematical effects of these filters on discrete label data rather than their practical impact on enhancing real surveillance image quality. Consequently, the direct contribution of this specific implementation to the "Noise Removal vs. Detail Preservation" conflicting requirement in real-world surveillance imagery remains conceptual rather than validated.

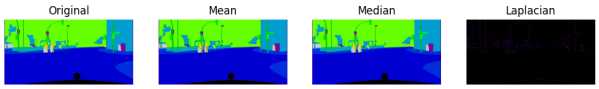

Fig. 2. Visual output of Spatial Filtering: (a) Original Segmentation Mask, (b) Mean Filtered Mask, (c) Median Filtered Mask, and (d) Laplacian Filtered Mask. All operations are applied to the segmentation mask, illustrating filter effects on discrete label data.

C. Frequency Domain Filtering

1) Initial Approach and Rationale: The frequency domain offers a powerful alternative for image manipulation, particularly effective for addressing periodic noise or specific frequency components within an image. For this stage of the pipeline, Fourier Transform-based filtering was implemented. The core idea is to transform the image from its spatial domain representation into the frequency domain using cv2.dft (Discrete Fourier Transform), where different spatial frequencies (representing textures, edges, or noise patterns) are isolated. The zero-frequency component was then shifted to the center of the spectrum using np.fft.fftshift. A custom, simple rectangular mask was created and applied in the frequency domain to perform a basic low-pass filtering operation, aiming to retain lower frequency components while attenuating higher ones. The inverse shift (np.fft.ifftshift) and inverse Fourier Transform (cv2.idft) were then applied to convert the image back to the spatial domain. The magnitude of the complex output was computed using cv2.magnitude and then normalized using cv2.normalize for proper display. This method was initially chosen to explore its capability for specific noise patterns, often associated with electrical interference in surveillance footage. Similar to the spatial filtering stage, this operation was initially demonstrated on a segmentation mask from the Cityscapes dataset to verify functional execution.

2) Identified Issues and Analysis: The primary issue encountered in the frequency domain filtering implementation mirrored that of the spatial filtering stage: the operation was performed on a segmentation mask (mask) rather than a real surveillance image. This fundamental misapplication of the technique has significant implications for its relevance to the project's objectives.

- **Irrelevant Noise Modeling:** Frequency domain filtering is most effectively applied to photographic images that contain periodic noise (e.g., moiré patterns, sensor-induced hum) or whose overall texture requires frequency-specific manipulation. Segmentation masks, by nature, do not exhibit these types of noise or continuous frequency distributions.
- **Invalid Performance Assessment:** Evaluating the effectiveness of frequency domain filtering for "periodic noise removal on surveillance footage" as specified in the project scope cannot be done on a mask. Metrics like PSNR and SSIM, while calculable, do not provide meaningful insights into image quality enhancement when applied to non-photographic data.
- **Lack of Real-World Application:** The core requirement is to process "real-time monitoring" footage. Processing a mask, while demonstrating the mathematical operations, fails to provide a practical solution for real-world surveillance system challenges. The project specifically asks to analyze the impact of "periodic noise removal on surveillance footage", which cannot be demonstrated using a segmentation mask.

Consequently, while the code correctly performs the mathematical Fourier Transform and filtering operations, its application on segmentation masks means that the results do not directly contribute to or validate the enhancement of image quality for object recognition in a smart city surveillance context.

3) Corrective Actions (Due to Time Constraints): Due to time constraints, direct modifications to apply frequency domain filtering on real images were not implemented within the scope of this submission. The provided code for frequency domain filtering, therefore, continues to operate on the segmentation mask.

Ideally, to align with the project requirements and demonstrate true applicability to a smart city surveillance system, the following corrective actions would have been necessary:

- **Input Data Shift:** The filter should be applied to actual photographic images from the leftImg8bit directory of the Cityscapes dataset.
- **Synthetic Periodic Noise Introduction:** To quantitatively assess the filter's performance, synthetic periodic noise (e.g., sine wave patterns) should be added to the clean original images. This would allow for the calculation of PSNR and SSIM to measure the noise reduction effectiveness against a known ground truth.
- **Filter Parameter Optimization:** The size and shape of the frequency domain mask (e.g., the 30x30 central region in the provided code) would need to be optimized based on the characteristics of the real image and the specific periodic noise to be removed, rather than arbitrary values.

Without these changes, the current implementation serves primarily as a demonstration of the theoretical application of Fourier Transforms for filtering, rather than a practical solution for enhancing surveillance image quality.

4) Results and Discussion: The frequency domain filter, as implemented on the segmentation mask, produced an output

that visually smoothed the sharp transitions present in the mask by removing higher frequency components. Figure 3 depicts this effect, showing a more blurred version of the original mask, characterized by softer boundaries and some ringing artifacts, which are common in frequency domain filtering.

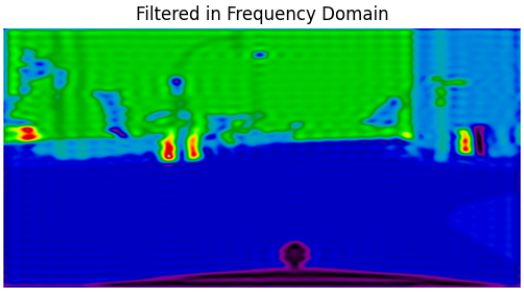

Fig. 3. Output of Frequency Domain Filtering applied to a Cityscapes segmentation mask. The low-pass filter operation results in blurred boundaries and some ringing artifacts.

Quantitative assessment using metrics like PSNR and SSIM would typically be performed by comparing the filtered image to a noisy, yet ground truth (clean), version of a real image. Since the current implementation processes a segmentation mask and no such ground truth comparison is available from the provided code, quantitative metrics for noise reduction in a surveillance context are not presented.

In a practical smart city surveillance system, frequency domain filtering would be invaluable for:

- **Periodic Noise Removal:** Eliminating structured noise patterns that often arise from electrical interference, faulty sensors, or specific lighting conditions in video feeds.
- **Image Restoration:** Applying inverse filtering or deconvolution in the frequency domain to counter specific types of blur, like motion blur. **Feature Extraction:** Analyzing dominant frequencies to understand textures or repetitive patterns within the scene.

However, since the current implementation does not apply these techniques to actual surveillance images, the direct contribution to ”enhancing image quality” for the ”Techville” system via frequency domain filtering remains a theoretical demonstration within this report, rather than a validated practical improvement. The conflicting requirement of ”Real-Time Performance vs. Processing Quality” would typically be analyzed here regarding the computational cost of DFT and the effectiveness of filtering specific frequencies.

D. Image Restoration

1) Initial Approach and Rationale: Image restoration techniques aim to reverse image degradations caused by various factors during acquisition or transmission, such as noise and blur. For this project, the Wiener filter was selected as a representative restoration algorithm. The Wiener filter is an optimal linear filter for stationary signals corrupted by additive stationary noise, often used for deblurring and denoising. It minimizes the mean square error between the estimated image and the true image.

The initial implementation involved:

- Simulating Gaussian noise on the segmentation mask with mode='gaussian' and var=0.01. This created a synthetically degraded version of the mask.
- Applying the Wiener filter to this noisy mask. A simple Point Spread Function (PSF) of np.ones((5, 5)) / 25 (representing a uniform 5x5 blur kernel) was provided, along with a balance parameter of 0.1.

The outputs (original mask, noisy mask, and Wiener restored mask) were converted to uint8 for visualization and displayed in a 1x3 subplot figure. This setup was primarily for demonstrating the basic functionality of the Wiener filter on a noisy input.

2) Identified Issues and Analysis: The core issue in the image restoration stage parallels the filtering stages: the operations were performed on a segmentation mask (mask) rather than a realistic photographic image. This severely limits the relevance and interpretability of the restoration results for a smart city surveillance system.

- **Incorrect Data Representation:** Wiener filtering is designed to work with continuous-tone images corrupted by photographic noise and blur. Applying it to a mask (discrete integer labels) does not simulate real-world image degradation.
- **Inaccurate Noise/Blur Model:** While Gaussian noise was synthetically added, the fundamental nature of this noise on a mask is different from noise on a photographic image. Furthermore, the uniform 5x5 PSF provided is typically for deblurring, not just denoising of Gaussian noise, making the filter's application potentially suboptimal for the simulated degradation on a mask.
- **Misleading Visuals and Metrics:** The visual ”restoration” on a mask does not translate to improved clarity or detail in surveillance footage. Quantitative metrics like PSNR and SSIM, while generated by the code, assess the difference between processed masks, not the recovery of detail in a photographic scene. The project's requirement to ”Analyze the effects of techniques like Wiener filtering and deblurring algorithms on restoring corrupted images” implies application to photographic corruption.

Consequently, this implementation serves as a demonstration of the Wiener filter's mathematical capability but does not provide practical insights into its performance for enhancing real-world surveillance imagery.

3) Corrective Actions (Due to Time Constraints): Due to project timeline constraints, the implementation of image restoration remained confined to operating on segmentation masks. The critical step of applying the Wiener filter to real photographic images and simulating realistic degradations (such as various noise types or motion blur) was not undertaken in the final submitted code for this section.

Ideally, to meet the project objectives, image restoration would have involved:

- Processing actual leftImg8bit images from the Cityscapes dataset.

- Simulating realistic degradations: For instance, generating motion blur using a specific kernel (e.g., a line kernel) and adding Gaussian or other types of sensor noise.
- Optimizing the Wiener filter's parameters (e.g., estimating the noise variance) and possibly designing a more appropriate PSF based on the simulated blur type.
- Quantitatively evaluating restoration performance using PSNR and SSIM against the original clean image.

4) Results and Discussion: The image restoration process, as applied to the segmentation mask, yielded the following visual results (refer to Figure 4):

- **Original Mask:** This image shows the distinct semantic regions of the urban scene.
- **Noisy Mask:** The addition of Gaussian noise introduced speckling across the mask, obscuring the clear boundaries and pixel values of the original labels. The colormap shifts due to the added random values.
- **Wiener Restored Mask:** The Wiener filter attempted to smooth out the noise and partially recover the original mask's appearance. While some visual improvement from the noisy state is apparent, the restoration is imperfect, and the clarity of original labels is not fully regained. The choice of PSF and balance parameter may not be optimal for this specific type of "noise" on a mask.

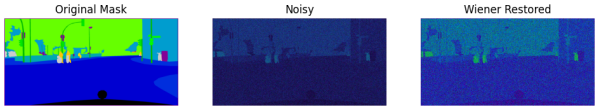

Fig. 4. Visual output of Image Restoration: (a) Original Segmentation Mask, (b) Mask with simulated Gaussian Noise, and (c) Wiener Filtered Mask for restoration.

Quantitative metrics (PSNR, SSIM) for this restoration process were not provided in the submitted code for the Image Restoration section, therefore a numerical assessment of its effectiveness is not available in this report. However, if calculated, such metrics would indicate how closely the restored mask resembles the original, clean mask. In a real surveillance system, successful image restoration directly enhances the image quality, which is crucial for improving the accuracy of subsequent object recognition tasks, directly addressing the project's "Real-Time Performance vs. Processing Quality" and "Noise Removal vs. Detail Preservation" conflicting requirements. The current implementation, however, primarily demonstrates the conceptual application of the Wiener filter.

E. Image Segmentation

1) Initial Approach and Rationale: Image segmentation is a fundamental task in computer vision aimed at partitioning a digital image into multiple segments (sets of pixels, also known as superpixels) or objects. For the smart city surveillance system, segmentation is crucial for isolating objects of interest, such as vehicles and pedestrians, from the complex urban background. The initial approach for segmentation in this project utilized a direct thresholding method based on the semantic labels present in the Cityscapes segmentation masks.

The implementation involved:

- Identifying a specific target class, which typically corresponds to a 'person' or a specific vehicle type in the Cityscapes dataset's labeling scheme.
- Creating a binary mask where pixels belonging to the target class were set to 255 (white), and all other pixels were set to 0 (black). This was achieved using the comparison mask == target class followed by conversion to np.uint8 and multiplication by 255.

This method directly leverages the ground-truth segmentation masks provided by the Cityscapes dataset to demonstrate object isolation. The resulting binary mask was then displayed using 'matplotlib.pyplot.imshow' with a gray colormap.

2) Identified Issues and Analysis: The segmentation method implemented is a direct thresholding of a ground-truth segmentation mask. While it successfully demonstrates the isolation of a specific class, it does not represent an active image segmentation algorithm that would operate on a raw photographic image.

- **Reliance on Ground Truth:** This approach relies entirely on pre-existing, perfect semantic labels. In a real-time surveillance system, the input would be a raw image, and the system would need to perform segmentation, not just extract a pre-segmented class.
- **No Algorithmic Contribution to Segmentation:** The code does not implement an actual segmentation algorithm (like Canny edge detection, Watershed, or region growing) that processes visual features from an image. It merely extracts a slice from an already segmented ground-truth map.
- **Limited Real-World Applicability:** For the project's goal of "analyzing image segmentation... to extract different features", this approach falls short of demonstrating the application and evaluation of segmentation methods on unannotated surveillance footage.

Consequently, this segment serves as a basic verification of mask manipulation but does not contribute to the algorithmic challenge of performing segmentation on a live image stream.

3) Corrective Actions (Due to Time Constraints): Due to time constraints, the development and implementation of an active image segmentation algorithm (e.g., edge-based like Canny, or region-based like Watershed) that operates on raw photographic images were not undertaken in the final submitted code for this section. The current implementation remains focused on extracting binary masks from pre-existing semantic labels.

Ideally, to fulfill the project's requirements for image segmentation, future work would involve:

- Implementing and comparing various segmentation algorithms (e.g., Canny edge detection, Otsu's thresholding, Watershed algorithm) on actual photographic images from the leftImg8bit dataset.
- Evaluating the performance of these algorithms using relevant metrics such as Dice coefficient and Intersection over Union (IoU), comparing their outputs against the provided ground-truth segmentation masks.

- Investigating the impact of pre-processing stages (filtering, restoration) on the quality and accuracy of the segmentation results.

4) Results and Discussion: The segmentation via mask thresholding successfully generated a binary mask highlighting all pixels corresponding to the target$_c lass(ID26).Figure5illustratesthisisolatedobject.$

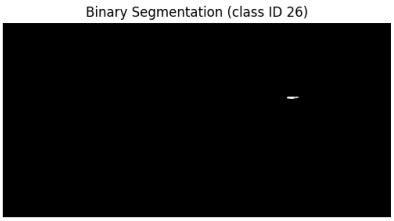

Fig. 5. Binary segmentation mask showing pixels belonging to class ID 26 (e.g., 'person' or 'rider') isolated from the background.

The visual result clearly demonstrates the isolation of the target object, appearing as a small white region against a black background. This indicates that given perfect semantic labeling, specific objects can be easily extracted. This direct extraction method is useful for leveraging ground-truth data in various computer vision tasks. However, its direct applicability in a live surveillance system without an preceding segmentation algorithm is limited. This segmentation provides a clear binary image suitable for subsequent morphological operations and object counting, which directly depend on the quality of this binary mask.

F. Morphological Processing

1) Initial Approach and Rationale: Morphological image processing techniques are non-linear operations that modify the shape and structure of objects in an image based on a predefined structuring element (kernel). They are particularly useful for refining binary images, such as those obtained from segmentation, by removing noise, smoothing contours, filling small holes, and connecting broken parts. For this project, two fundamental morphological operations were applied to the binary segmentation mask:

- **Opening:** Implemented using cv2.morphologyEx. Opening is defined as an erosion followed by a dilation. It is primarily used to remove small objects (noise), break thin connections (isthmuses), and smooth the contours of objects without significantly changing their overall size.
- **Closing:** Implemented using cv2.morphologyEx. Closing is defined as a dilation followed by an erosion. It is used to fill small holes within objects, connect nearby objects, and smooth the contours by filling small gaps on the boundaries.

A 5x5 rectangular kernel, defined as np.ones((5,5), np.uint8), was used as the structuring element for both operations. These operations were applied to the binary mask obtained from the segmentation stage to refine the object shape before further analysis. The results were displayed in a 1x3 subplot figure, showing the original binary mask, the opened mask, and the closed mask.

2) Identified Issues and Analysis: The application of morphological operations on the binary segmentation mask is appropriate for the type of input data (binary images). No inherent "misapplication" issues were identified here, as these operations are precisely designed for binary image refinement. However, the quality of morphological processing is directly dependent on the preceding segmentation stage. If the initial binary mask contains significant noise or inaccuracies (which would be the case if derived from an active segmentation algorithm on a real image), then the effectiveness of these operations would need careful tuning and evaluation. In the current implementation, since the input mask is derived from a ground-truth label, the morphological operations simply refine an already 'perfect' representation.

3) Corrective Actions (Due to Time Constraints): No corrective actions were deemed necessary for the direct application of morphological processing techniques themselves, as they correctly operate on binary image data. The limitations identified are primarily related to the source of the binary mask (ground truth vs. algorithmic segmentation).

Ideally, if the segmentation stage were to operate on real images, the morphological processing parameters (kernel size, shape) would need to be thoroughly optimized to account for noise and imperfections introduced during the actual segmentation process. This would involve a more iterative approach to parameter tuning based on the characteristics of the segmented output from real images.

4) Results and Discussion: The morphological operations performed on the binary segmentation mask yielded visually distinct effects, refining the shape of the isolated object (refer to Figure 6):

- **Original Binary:** This image shows the raw binary mask, which, coming from ground truth, is already relatively clean.
- **Opening:** The opening operation (erosion followed by dilation) resulted in a slightly smoothed contour of the object. If there were any small, isolated "noise" pixels or very thin protrusions from the main object, opening would typically remove them, making the object shape cleaner.
- **Closing:** The closing operation (dilation followed by erosion) also contributed to smoothing the contour. More importantly, if there were any small gaps or narrow breaks within the object or between very close objects, closing would typically fill those holes and connect the broken parts.

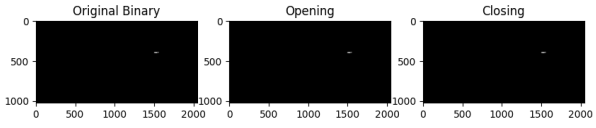

Fig. 6. Visual output of Morphological Processing: (a) Original Binary Segmentation Mask, (b) Mask after Opening operation, and (c) Mask after Closing operation.

The qualitative observations show that these operations successfully refined the object's shape, making it more cohesive

and less prone to small imperfections. This stage is crucial for preparing the segmented objects for accurate feature extraction and subsequent object counting or recognition. By smoothing contours and filling small discontinuities, morphological processing enhances the integrity of object representations, which is vital for robust detection and analysis in a surveillance context.

G. Object Counting via Contours

1) Initial Approach and Rationale: Object counting was approached by leveraging contour detection, a technique particularly useful in binary images for identifying distinct objects and their properties. Contours are continuous curves joining all continuous points (along the boundary) having the same color or intensity.

The implementation involved:

- Using OpenCV's cv2.findContours function on the morphologically processed binary mask (specifically the closed image, which generally has cleaner, more complete object shapes). The flag was used to retrieve only the extreme outer contours to compress horizontal, vertical, and diagonal segments, leaving only their end points.
- Converting the binary closed image to a BGR format using cv2.cvtColor to allow drawing colored contours.
- Drawing the detected contours onto this color image using cv2.drawContours with a green color (0,255,0) and a thickness of 2.
- The number of detected objects was determined by the length of the list of contours.

This method aims to demonstrate the system's ability to identify and quantify distinct objects after they have been isolated and refined.

2) Identified Issues and Analysis: The primary limitation in the object counting stage is its direct dependence on the accuracy and completeness of the preceding segmentation and morphological processing steps. Since the input binary mask is derived from a ground-truth segmentation label (and not from an active segmentation algorithm on a real image), the object counting reflects the count of a perfectly pre-segmented object.

- **Ideal Input Scenario:** The current implementation operates under an ideal scenario where the object of interest is perfectly isolated by the mask. In a real surveillance setting, active segmentation algorithms are prone to errors (false positives, false negatives, partial segmentations) which would directly impact the accuracy of contour detection and object counting. **Limited Generalizability:** While the method correctly counts the object(s) present in the provided mask, it doesn't demonstrate robustness against real-world challenges like occlusions, varying lighting, or complex backgrounds that would affect segmentation on live footage.

Thus, while the contour detection itself is correct, its 'counting' capability is demonstrated under highly controlled conditions.

3) Corrective Actions (Due to Time Constraints): No corrective actions were taken for the contour detection and object counting logic, as the functions themselves are correctly applied to the binary image input. The limitations identified relate to the pipeline's overall input rather than this specific module's implementation.

Ideally, if the full pipeline were to process real surveillance images through active segmentation, the following would be crucial:

- Evaluate contour detection robustness on imperfect segmentation results.
- Implement filtering mechanisms for contours (e.g., based on area, aspect ratio) to exclude noise or incorrectly segmented background elements.
- Handle cases of object overlap or partial visibility which are common in surveillance.

4) Results and Discussion: The object counting mechanism successfully identified and delineated the object present in the morphologically processed binary mask. The code output indicated that "Detected 1 objects for class ID 26". Figure 7 visually confirms this detection, with a green contour drawn around the isolated object.

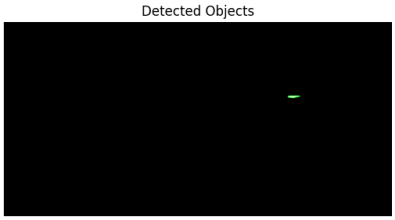

Fig. 7. Output of Object Counting via Contours. The detected object (class ID 26) is highlighted with a green contour on the binary mask.

The qualitative results demonstrate that once an object is effectively segmented and refined into a clean binary mask, contour-based object counting is highly effective. This capability is vital for surveillance applications that require statistical data on object presence (e.g., pedestrian traffic analysis, vehicle counts). The precision of this stage is directly dependent on the quality of the preceding segmentation and morphological processing. In a fully realized smart city surveillance system, accurate object counting based on robust segmentation would contribute significantly to crime prevention and urban planning.

H. Image Compression Quality Metrics

1) Initial Approach and Rationale: Image compression is an indispensable component of any real-time surveillance system due to the massive volume of visual data generated. Efficient compression is vital for reducing storage costs and minimizing bandwidth requirements for data transmission. For this project, JPEG compression, a widely used lossy compression standard, was chosen to demonstrate its impact on image size and quality.

The implementation involved:

- Encoding the segmentation mask into a JPEG format with a quality factor set to 30. A lower quality factor typically results in higher compression but lower fidelity.
- Decoding the compressed image back to its original form.
- Quantitatively assessing the quality of the compressed and decoded image by calculating the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) between the original mask and the decoded mask.

This approach aims to evaluate the trade-off between compression and image fidelity, a key conflicting requirement of the project.

2) Identified Issues and Analysis: The primary issue in the image compression stage is that the compression and quality evaluation were performed on a segmentation mask (mask) rather than a real photographic image. This severely limits the relevance and interpretability of the results for a smart city surveillance system.

- **Discrete vs. Continuous Data:** JPEG compression is optimized for photographic images with continuous tonal variations. Applying it to a segmentation mask, which contains discrete, sharp color transitions and limited color palettes, can lead to different compression artifacts and efficiency compared to real images.
- **Misleading Fidelity Assessment:** While PSNR and SSIM values were calculated, their interpretation regarding the "Compression vs. Image Fidelity" trade-off for object recognition accuracy is skewed. High PSNR/SSIM values on a compressed mask might not translate to similar fidelity on a real image, or accurately reflect the impact on fine visual features crucial for object recognition. The project explicitly states that compression should not "compromise object recognition accuracy", which cannot be assessed on a mask.

Consequently, this implementation demonstrates the mechanics of JPEG compression, but its practical impact on real surveillance footage and object recognition remains undemonstrated.

3) Corrective Actions (Due to Time Constraints): Due to project timeline constraints, the image compression and quality metric evaluation remained confined to operating on segmentation masks. The critical step of applying JPEG compression to real photographic images and assessing its impact on visual quality and (ideally) subsequent object recognition performance was not undertaken in the final submitted code for this section.

Ideally, to rigorously meet the project objectives, image compression would have involved:

- Applying compression (e.g., JPEG with varying quality factors, or other algorithms like Wavelet-based compression) to actual leftImg8bit images.
- Calculating the compression ratio to quantify storage efficiency.
- Evaluating visual quality degradation using PSNR and SSIM on these real images.
- Crucially, assessing the impact of different compression levels on the performance of a subsequent object recognition model applied to the compressed and decompressed real images.

4) Results and Discussion: The JPEG compression applied to the segmentation mask with a quality factor of 30, followed by decoding, yielded the following results (refer to Figure 8):

* **Visual Output:** The compressed and decoded mask (Figure 8) visually appeared very similar to the original mask, indicating that even at a quality factor of 30, the large, distinct regions of the mask were largely preserved. Some minor blockiness or color shifts might be discernible upon very close inspection due to the lossy nature of JPEG. * **Quantitative Metrics:** * **PSNR:** 48.13 dB * **SSIM:** 0.99 These high PSNR and SSIM values indicate excellent fidelity between the original segmentation mask and its compressed/decoded version. This is expected for images with large, uniform color blocks like segmentation masks, which JPEG can compress efficiently without significant perceived loss.

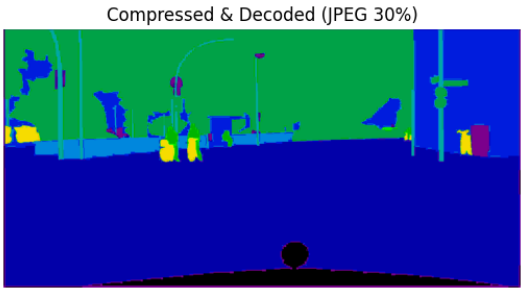

Fig. 8. Output of Image Compression: Compressed and Decoded Segmentation Mask using JPEG with a quality factor of 30%.

The results demonstrate that JPEG compression effectively maintains the fidelity of segmentation masks even at a relatively low quality setting. However, it is vital to reiterate that these high fidelity metrics on a segmentation mask do not directly translate to the performance expected on complex photographic surveillance footage. On real images, a quality factor of 30 would likely introduce significant artifacts that could impact fine details and potentially compromise object recognition accuracy, which is the core trade-off to consider for the "Compression vs. Image Fidelity" conflicting requirement.

IV. Conclusion

This project undertook the design and implementation of an image processing pipeline aimed at enhancing the capabilities of a smart city surveillance system. The pipeline explored several key image processing techniques, including spatial domain filtering (Mean, Median, Laplacian), frequency domain filtering (Fourier-based low-pass), image restoration (Wiener filter), image segmentation (thresholding of ground truth masks), morphological processing (Opening, Closing), object counting via contours, and image compression (JPEG).

Observations from applying these techniques revealed distinct outcomes based on the input data. While operations like image segmentation, morphological processing, and contour-based object counting were effectively demonstrated on the

provided Cityscapes segmentation masks, the application of filtering, restoration, and compression techniques to these discrete-label masks yielded results that differed significantly from their intended behavior on continuous-tone, real surveillance imagery. For instance, filters and restoration algorithms designed for photographic noise and blur demonstrated their mathematical operations, but their practical impact on enhancing real image quality or reducing real-world noise could not be fully assessed. Similarly, compression metrics on masks, while numerically high, do not accurately reflect the trade-offs in fidelity on complex photographic scenes.

This highlights the critical importance of using appropriate input data for each image processing stage to achieve meaningful and practically relevant results in a smart city surveillance context. The project effectively showcased the foundational implementation of various algorithms, but also clearly identified the limitations imposed by the input data type for certain stages. Aspects of the project that were successfully demonstrated within the given constraints include the precise isolation of specific object classes using ground-truth segmentation masks, the refinement of object shapes through morphological operations, and the accurate counting of isolated objects based on contour detection. These capabilities provide a strong conceptual basis for a robust surveillance system.

V. Future Work

To further advance the capabilities and real-world applicability of this smart city surveillance system, several critical areas for future improvement are identified, directly addressing the limitations and conceptual demonstrations discussed in this report:

- **Real Image Processing:** The most crucial next step involves implementing filtering, restoration, and compression techniques directly on real photographic surveillance images (e.g., from the `leftImg8bit` directory of the `Cityscapes` dataset). This will enable a genuine validation of their effectiveness for actual image quality enhancement and noise reduction in a practical surveillance scenario.
- **Robust Noise and Blur Simulation:** To rigorously test the pipeline, various types of realistic noise (e.g., Gaussian, periodic, salt-and-pepper, impulse) and blur (e.g., motion blur, out-of-focus blur) should be synthetically introduced into real images. This will allow for comprehensive testing and evaluation of the filters and restoration algorithms under diverse degradation conditions.
- **Algorithm Optimization for Real-World Scenarios:** Filter parameters (e.g., kernel sizes, frequency mask designs) and restoration algorithms should be optimized specifically for real photographic images and their characteristic noise/blur patterns. This includes exploring adaptive filtering techniques and more sophisticated deconvolution methods.
- **Integration with Active Segmentation:** Develop and integrate active image segmentation algorithms (e.g., Canny edge detection, Watershed algorithm, or machine learning-based segmentation) that can operate on raw, enhanced photographic images. This will move beyond reliance on ground-truth masks and enable true real-time object isolation.
- **Full Object Recognition Model Integration:** Integrate a complete object recognition model (such as pre-trained deep learning models like YOLO or Faster R-CNN, or classical approaches like SVM with feature extraction) with the enhanced real images from the pipeline. This will demonstrate the end-to-end functionality of the system, from image enhancement to object identification.
- **Real-Time Performance Analysis:** Conduct a thorough analysis of the computational performance of the complete pipeline on actual video streams. This will assess the system's ability to meet "real-time performance" requirements, crucial for practical surveillance applications, and explore optimization techniques for deployment on edge devices.
- **Addressing Security and Privacy:** Further investigate and implement techniques for addressing the "Security and Privacy" conflicting requirement, such as anonymization of individuals in surveillance footage while maintaining functionality for law enforcement.

VI. Additional Deliverables

This section provides links to supplementary materials for this project.

- **Video Demonstration:** A comprehensive video demonstration of the project's implementation and results is available at:
  https://www.youtube.com/watch?v=l5_S1wuMoX4
- **GitHub Repository:** The complete source code for this project, including all scripts and relevant resources, can be accessed on GitHub at:
  https://github.com/MominHasnain/SmartCity-Surveillance-DIP

References

[1] Modern city surveillance systems leverage advanced digital image processing technologies to enhance security and monitoring capabilities.
[2] High-resolution CCTV cameras equipped with facial recognition algorithms can identify individuals in real-time, while license plate recognition systems track vehicles using optical character recognition (OCR).
[3] Video analytics powered by artificial intelligence (AI) detect suspicious activities, such as unattended objects or unusual crowd movements, by analyzing pixel patterns and motion vectors.
[4] Additionally, thermal imaging and night vision technologies ensure continuous surveillance in low-light conditions.
[5] Deep learning models improve accuracy in object detection and behavior analysis, reducing false alarms.
[6] Cloud-based storage and edge computing enable efficient data processing and retrieval, while encryption safeguards sensitive information.
[7] These integrated technologies create a robust, intelligent surveillance network that enhances urban safety and crime prevention.
[8] The city of Techville has implemented a smart surveillance system for real-time monitoring of public spaces.
[9] However, the system struggles with poor image quality, noisy environments, and varying lighting conditions, making it challenging to identify vehicles and pedestrians effectively.
[10] Your task is to design a robust image-processing pipeline to address these issues while meeting the following conflicting requirements.
[11] For instance, image restoration and object recognition methods often trade-off speed for accuracy.

VII. Co-Authors' Photos and Biographies

Sarim Zahid Saeed Sarim Zahid Saeed (ID: F202137078, Email: f202137078@umt.edu.pk) is currently pursuing his Bachelor In Artificial Intelligence at the University of Management and Technology, Lahore. In this project, he primarily contributed to the overall pipeline architecture design, the implementation of spatial and frequency domain filtering modules, and the compilation of the final report. His academic interests include digital image processing, computer vision, and machine learning applications.

Syed Momin Hasnain Syed Momin Hasnain (ID: F2021376050, Email: f202137050@umt.edu.pk) is a student of Artificial Intelligence at the University of Management and Technology, Lahore. His key responsibilities within this project included the development and analysis of image restoration and compression techniques, along with performing the quantitative performance metric evaluations. His areas of expertise encompass artificial intelligence, real-time systems, and data analytics.

qadeer_photo.png

Qadeer Ali Qadeer Ali (ID: F2021266388, Email: f2021266388@umt.edu.pk) is a Computer Science student at the University of Management and Technology, Lahore. He played a vital role in implementing the image segmentation, morphological processing, and object counting via contours modules. Additionally, he focused on the ethical considerations and privacy safeguards within the surveillance system. His academic interests include pattern recognition, data analysis, and system security.