ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1098

Edge-Based Object Counting System for Smart City Applications: An Adaptive Morphological Segmentation Framework

Pragya Mittal[1]; Jesna Jixon[2]; Ritika Palai[3]; Mrudula Wani[4]; Shamla Mantri[5]

[1;2;3;4;5]Department of Computer Engineering & Technology, MIT-WPU, Pune, Maharashtra, India

Publication Date: 2026/07/07

Abstract: The growing presence of smart city infrastructure has created a need for efficient, real-time object counting systems that operate independently of cloud services or complex deep learning models. Current methods based on background subtraction require a static reference frame, while deep learning detectors rely on large annotated training datasets and GPU hardware[16][19]. This paper presents the Adaptive Morphological Refinement (AMR) algorithm, a classical image processing pipeline implemented in MATLAB[20][21]. It combines dual-channel edge detection (Canny and Sobel), density-adaptive morphological structuring element selection, watershed-based object separation, and a seven-criterion shape filter to detect and count objects in urban surveillance images. The proposed approach is evaluated and compared against two segmentation-based baseline methods [1][2], demonstrating the advantage of density-adaptive parameter selection over fixed-parameter classical schemes. A proof-of-concept evaluation on two real urban surveillance images yielded object counts of 27 and 24, with complete pipeline telemetry confirming the density-adaptive feature.

Keywords: *Adaptive Morphological Refinement, Edge Detection, Object Counting, Smart City, Canny Operator, Sobel Gradient, Watershed Segmentation, Connected Components, MATLAB.*

How to Cite: Pragya Mittal; Jesna Jixon; Ritika Palai; Mrudula Wani; Shamla Mantri (2026) Edge-Based Object Counting System for Smart City Applications: An Adaptive Morphological Segmentation Framework. *International Journal of Innovative Science and Research Technology*, 11(6), 2585-2594. https://doi.org/10.38124/ijisrt/26jun1098

I. INTRODUCTION

The rise of Internet of Things (IoT) devices and smart surveillance cameras in modern cities creates a pressing need for real-time object counting systems for vehicles, pedestrians, and cyclists. These systems support traffic management, crowd monitoring, and public safety applications [3][19]. Vehicle movement tracking and motion-based classification have been studied extensively using both classical and modern approaches [6][7]. Traditional cloud-based analytics introduce network delays that exceed 500 milliseconds, preventing real-time decision-making [16][19]. Edge computing solves this by processing data at the camera, reducing transmission delays and protecting privacy [19]. Motion detection on RGB-D images has also been explored for vehicle classification at the edge [7].

Deep learning detectors like YOLOv5 and YOLOv8 achieve high accuracy but depend on large annotated training datasets and GPU hardware, making them unsuitable for low-cost edge solutions [16][17]. Classical image processing methods, such as edge detection, morphological segmentation, and connected component analysis, offer a lightweight, training-free alternative [9][12]. However, existing classical frameworks have limitations, including dependence on a static reference image, poor object separation in dense scenes, and fixed parameters that do not adapt to changing conditions[3][8]. Recent segmentation-based works[1][2]further highlight these limitations: fixed-parameter schemes produce systematic failure modes across varying scene densities, and morphological methods without adaptive mechanisms show reduced separation quality in cluttered environments.

This paper introduces the Adaptive Morphological Refinement (AMR) algorithm, a MATLAB-based pipeline that automatically adjusts its morphological structuring element and watershed seed distances based on local edge density. The main contributions are:

- Dual-channel edge fusion with Canny and Sobel gradient maps [21];
- Density-adaptive structuring element selection that changes based on scene congestion [20];
- Density-adaptive watershed segmentation for separating overlapping objects; and
- A seven-criterion shape filter for effectively rejecting non-object areas [22][24] — all without needing training data or GPU.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1098

II. LITERATURE SURVEY

➢ *Frame Differencing and Background Subtraction*

Frame differencing and background subtraction approaches subtract a reference or modelled background from each frame to isolate moving foreground objects. Abbas et al. [3] proposed reference image subtraction with grayscale thresholding and connected component analysis for traffic density estimation, demonstrating practical real-time applicability but suffering from static reference frame dependency and boundary merging in congested scenes. Background subtraction with morphological post-processing and ROI line-crossing logic [4] improves noise rejection but remains sensitive to illumination changes. The GMM-based framework [5] provides greater environmental adaptability through a statistical background model yet is still limited by blob-merging artefacts under dense traffic and requires continuous model maintenance. Singh and Kaur [8] employ a frame differencing and summing technique for foreground extraction but report fragmented segmentation under outdoor noise. Collectively, these methods avoid training data but cannot operate without a background reference and break down under dynamic illumination.

➢ *Edge Detection and Morphological Segmentation*

Edge detection and morphological segmentation approaches use gradient operators to locate object boundaries and morphological operations to refine them into solid regions. Prutha et al. [9] applied dilation, erosion, and edge detection for real-time highway vehicle detection, achieving improved boundary delineation over background subtraction but remaining vulnerable to object merging at high densities. A Canny-based system for smart traffic control [10] uses pixel-level edge density as a vehicle metric, providing only a density ratio rather than a discrete object count. Ibrahim et al. [12] evaluated morphological edge detection algorithms on noisy vehicle datasets, demonstrating that morphological operations substantially improve boundary clarity under image degradation — a key motivation for the morphological component of AMR. Shah and Mehta [13] incorporated Canny with bilateral filtering for number plate recognition, while Vennila and Kavitha [11] applied morphological operators in license plate detection, confirming that combined edge-morphology pipelines outperform single-operator approaches. These works confirm the effectiveness of combining edge detection with morphological refinement but reveal the absence of adaptive parameter selection.

Closely related to the present work, Hossain et al. [2] proposed an object tracking method based on edge detection and morphological operations that demonstrates effective boundary delineation in structured scenes. While [2] employs a pipeline structurally similar to AMR — combining gradient-based edge maps with morphological post-processing — it applies fixed structuring element parameters regardless of scene density and incorporates a temporal tracking layer that adds computational overhead. The absence of density-driven adaptation in [2] results in degraded object separation in high-density frames, a limitation that AMR directly addresses through its closed-loop density measurement and parameter adaptation mechanism described in Section III.

➢ *Connected Component and Blob-Based Counting*

Connected component and blob-based methods label distinct foreground regions after segmentation and count them directly. Patel and Verma [14] integrated detection, tracking, and counting using connected components with temporal tracking, improving consistency but showing reduced performance in dense or occluded scenes. Kulkarni and Joshi [15] applied binary thresholding and blob labelling for a computationally efficient pipeline, demonstrating weak edge accuracy in cluttered backgrounds and insufficient morphological refinement for complex illumination. Both works confirm that connected component counting accuracy is fundamentally bounded by the quality of the upstream segmentation stage, motivating the improved pipeline proposed in AMR.

A directly comparable baseline is the work of Jyoti et al. [1], which presents a systematic comparison of image segmentation algorithms — including Otsu thresholding, region-growing, and watershed — applied to object counting on real-world images, reporting accuracy using the Jaccard similarity index. While [1] establishes a useful comparative benchmark for segmentation-based counting, it does not incorporate adaptive parameter selection, density-driven morphological refinement, or multi-criterion shape validation. Consequently, the methods studied in [1] exhibit degraded performance when scene density varies significantly across images, a gap that AMR addresses by design.

➢ *Deep Learning and Modern Approaches*

Deep learning approaches leverage convolutional neural networks trained on large annotated datasets to detect and count objects. Kaur and Singh [16] deployed YOLOv5 with DeepSORT on a Jetson Nano, achieving approximately 98% detection accuracy but only 49.95% counting accuracy, with computational demands exceeding practical real-time limits. Zhang et al. [17] proposed YOLOv8n with knowledge distillation achieving 254 FPS and 99% mAP@0.5 at a compressed model size, but the system struggles with severe occlusion and requires manual deployment configuration per site. Li et al. [18] introduced a four-branch self-attention context-aware network with density map estimation for scale-invariant counting, requiring extensive annotated training data and never evaluated on edge hardware. Viola and Jones [24] introduced the cascade-of-classifiers framework for rapid object detection that remains influential in lightweight detection design. Howard et al. [25] proposed MobileNets as efficient convolutional networks for mobile applications, providing a pathway for embedding lightweight classifiers downstream of classical pipelines. Colombo et al. [19] confirm in a comprehensive review that lightweight, training-free classical approaches remain significantly underexplored for smart city edge deployments — the primary motivation for the AMR framework.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1098

➢ *Method Comparison*

Table I summarises the object counting approaches reviewed above, comparing them on parameters directly relevant to the objectives of this work. As illustrated in Table I, AMR is the only approach combining training-free operation, adaptive density-driven morphology, and discrete object counting without GPU dependency.

Furthermore, unlike conventional methods that rely on fixed parameters or require extensive training data, the proposed AMR framework dynamically adapts to scene variations, enabling robust performance across both sparse and dense environments. This highlights its suitability for real-time, resource-constrained applications such as smart city surveillance and edge-based deployment.

Table 1 Comparison of Object Counting Approaches from Literature

Approach	Train?	GPU?	Density Adaptive?	Discrete Count?	Key Limitation
Frame Diff. / BG Sub. [3][4][5][8]	No	No	No	Approx. only	Needs reference frame; fails under dynamic lighting
Segmentation-Based Counting [1]	No	No	No	Partial	Fixed parameters; Jaccard accuracy only
Edge Detection + Morphology [2][9][10][12][13]	No	No	No	Partial	Fixed parameters; density ratio, not discrete count
Connected Component Counting [14][15]	No	No	No	Yes	Bounded by upstream segmentation quality
Deep Learning (YOLO, CNN) [16][17][18]	Yes	Yes	No	Yes	Needs annotated data; high compute; not edge-friendly
Proposed AMR (This Work)	**No**	**No**	**Yes**	**Yes**	Occlusion; no class distinction; POC scale only

➢ *Research Gap*

No existing classical framework simultaneously combines dual edge detection fusion, density-adaptive morphological structuring element selection, adaptive watershed separation, and multi-criterion shape validation — all without training data, GPU, or a reference image. The segmentation-based counting study [1] and the edge-morphology tracking method [2] represent the closest related work but each lacks the closed-loop density adaptation that AMR introduces. Guerrero-Gomez-Olmedo et al. [22] highlighted the difficulty of counting under extreme overlap, motivating the adaptive watershed seeding mechanism in AMR. Sharma and Gupta [6] further demonstrate that trajectory-based approaches require continuous frame availability, reinforcing the value of stateless per-frame processing. Additionally, existing approaches rarely incorporate real-time parameter self-adjustment based on local scene density variations, limiting their robustness and generalizability across diverse real-world environments. AMR addresses the identified gap in a single interpretable pipeline.

III. PROPOSED METHODOLOGY

➢ *System Architecture*

The AMR pipeline processes each surveillance image through eight sequential stages, illustrated in the flowchart in Fig. 1: (1) preprocessing with contrast enhancement, (2) dual-channel edge detection, (3) edge strengthening, (4) region formation via morphological closing and hole-filling, (5) adaptive morphological refinement — the core contribution, (6) watershed-based object separation, (7) seven-criterion shape filtering, and (8) object counting and visualization. The pipeline requires no reference frame, no training, and uses only standard image processing operations available in MATLAB's Image Processing Toolbox [20][21]. Fig. 2 presents the full system architecture diagram. Table II enumerates each pipeline stage with its corresponding operation and output intermediate.

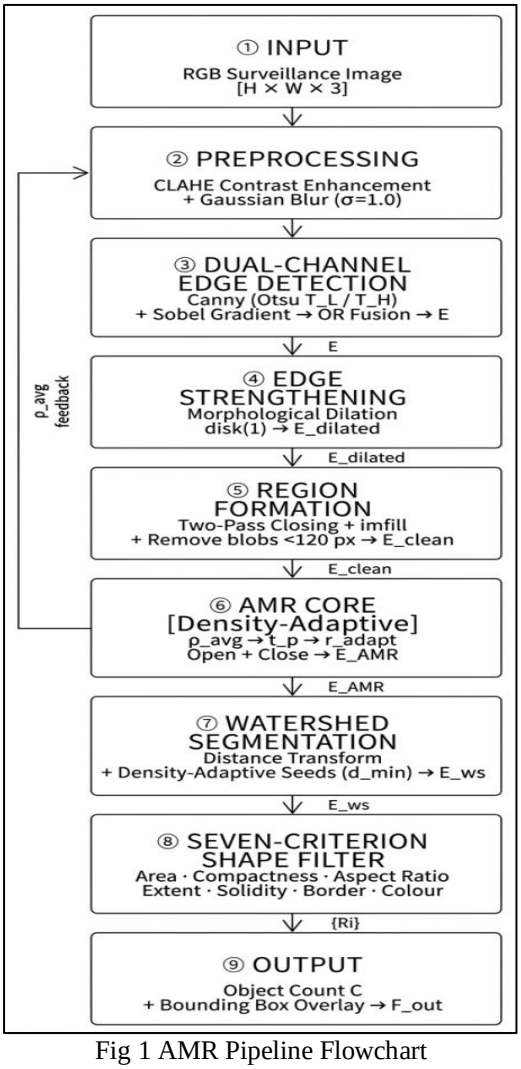

Fig 1 AMR Pipeline Flowchart

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1098

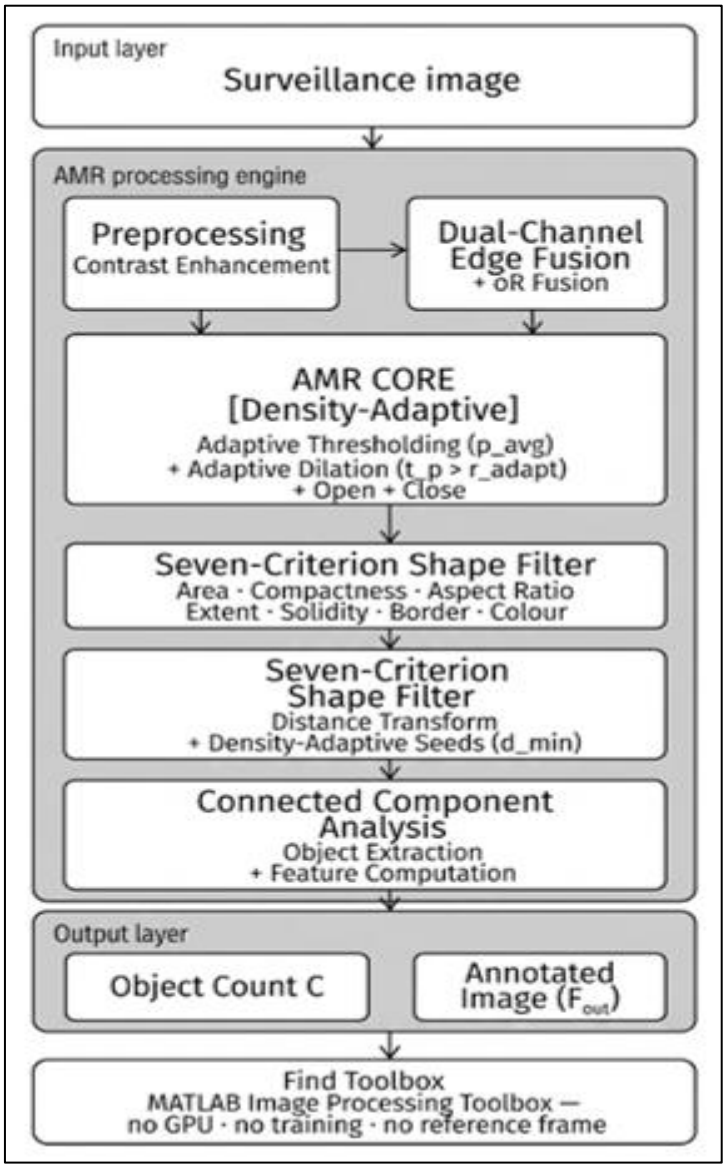

Fig 2 AMR System Architecture

Table 2 AMR Pipeline — Eight Stages

Stage	Operation	Output
Pre-processing	CLAHE contrast enhancement + Gaussian blur	Enhanced image G_σ
Edge Detection	Canny (Otsu thresholds) + Sobel gradient → logical OR	Fused edge map E
Edge Strengthen	Morphological dilation with disk(1)	Strengthened edges
Region Formation	Two-pass closing + imfill + remove tiny blobs	Solid regions E_clean
AMR (Core)	Compute density → adapt SE radius → open + close	Refined map E_AMR
Watershed	Distance transform + density-adapted seeds → split	Separated blobs E_ws
Shape Filter	Evaluate 7 criteria per blob → accept or reject	Valid regions $\{R_i\}$
Count & Output	C = count of accepted blobs + bounding box overlay	Count C, image F_out

- *Innovation Over Existing Methods*

The principal innovation of AMR over all prior classical methods reviewed in Section II is the introduction of a closed-loop, scene-driven parameter adaptation mechanism that requires no human intervention and no reference frame. Existing edge-detection and morphological segmentation systems — including those proposed by Prutha et al. [9], Desai and Kulkarni [10], and the tracking-based approach of Hossain et al. [2] — apply a fixed structuring element radius and fixed watershed seed spacing to every image regardless of how crowded or sparse the scene is. Similarly, the segmentation-based counting methods surveyed in [1] apply static algorithm parameters manually tuned to a specific image type, which do not generalise across variable scene densities. This static design produces a systematic failure mode: in dense scenes the element is too large and merges adjacent objects into a single blob, while in sparse scenes it is too small and leaves boundary gaps unfilled.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1098

AMR eliminates this failure mode by first measuring local edge pixel density ρ_avg across the entire frame and deriving a normalised density factor tρ ∈ [0, 1] (Equation 1). The dual-channel edge fusion — combining Canny's gradient suppression [21] with Sobel magnitude thresholded at the same Otsu level [20] — further strengthens boundary delineation beyond what either operator achieves alone, addressing the incomplete edge maps that limit connected-component accuracy in prior work [14][15]. The seven-criterion shape filter acts as a scene-adaptive validation stage: rather than relying on a single threshold as in background-subtraction methods [3][4], it evaluates each candidate blob against independent geometric and photometric criteria [22], ensuring that road markings, walls, and boundary artefacts are systematically excluded while genuine objects of varying scale are retained.

➢ *Core Mathematical Formulation*

The full pipeline relies on many standard operations (Canny edge detection [21], Otsu thresholding [20], morphological opening/closing, watershed, connected components). Three formulas capture the novel adaptive mechanism at the heart of AMR.

- *Scene Density Factor*

The local edge pixel density ρ_avg is computed by averaging the edge map E over all pixels. A density factor t_ρ is then derived, normalised to the range [0, 1]:

$$t\_\rho: \min(\rho\_avg \div 0.25,\ 1.0) \quad (1)$$

When t_ρ = 1 the scene is dense (many objects close together). When t_ρ = 0 the scene is sparse. This single number drives the adaptation of the two parameters below.

- *Adaptive Structuring Element Radius*

The radius r of the morphological structuring element (SE) automatically adjusts based on t_ρ. In a dense scene a small SE preserves the separation between nearby objects. In a sparse scene a large SE bridges broken boundary gaps:

$$r\_adapt: r\_max + (r\_min - r\_max) \times t\_\rho \quad (2)$$

Default values: r_min = 2 px, r_max = 7 px. At maximum density (t_ρ = 1): r_adapt = 2 px. At minimum density (t_ρ = 0): r_adapt = 7 px. The SE is applied in morphological opening (removes noise) and closing (fills boundary gaps).

- *Object Count*

After shape filtering, each candidate blob R_i is assigned a binary indicator $O_i = 1$ if it passes all seven shape criteria, otherwise 0. The total object count is:

$$C: \Sigma_i O_i \quad \text{(sum over all n candidate blobs)} \quad (3)$$

The seven shape criteria (area, compactness, aspect ratio, extent, solidity, border-touch, and colour saturation) are described in Section III-D.

➢ *Algorithm Pseudocode*

Algorithm given below presents the complete pseudocode for the AMR pipeline. The algorithm runs in O(H·W) time and space per image, making it suitable for resource-constrained edge hardware [19]. Each step maps directly to one of the eight pipeline stages summarised in Table 2.

ALGORITHM 1: Adaptive Morphological Refinement (AMR) Algorithm
Input : RGB surveillance image F [H × W × 3]
Output: Object count C, annotated image F_out
1. G ← rgb2gray(F)
2. G_σ ← CLAHE(G) then GaussianBlur(σ=1.0)
3. T_otsu ← OtsuThreshold(G_σ) [Ref.18]
4. E_c ←CannyEdge(G_σ, 0.75·T, 1.25·T) [Ref.19]
5. M ← SobelMagnitude(G_σ)
6. E ← E_c OR (M_norm> 0.75·T_otsu)
7. E ← dilate(E, disk radius=1)
8. E ← imclose(imclose(E, disk(4)), disk(4))
9. E_clean ← imfill(E, holes) → remove blobs <120 px
10. ρ_avg← mean edge density over full image
11. t_ρ ←min(ρ_avg / 0.25, 1.0) [Eq. 1]
12. r ← r_max + (r_min − r_max) × t_ρ [Eq. 2]
13. E_AMR← imclose(imopen(E_clean, disk(r)), disk(r))
14. D ← DistanceTransform(E_AMR)
15. d_min← 22 + (6 − 22) × t_ρ (adapts with density)
16. E_ws ← Watershed(E_AMR, seeds spaced d_min apart)
17. C ← 0
18. for each blob R_i in ConnectedComponents(E_ws):
19. if R_i passes all 7 shape criteria:
20. C ← C + 1; draw green box on F_out [Eq. 3]
21. return C, F_out
Complexity: O(H·W) time and space per image

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1098

➢ *Seven-Criterion Shape Filter*

Each connected blob extracted after watershed separation is evaluated against seven independent shape and colour criteria, as listed in Table 3. A blob is counted only if it satisfies all seven criteria. This multi-criterion approach substantially outperforms the single-threshold binary acceptance used in background-subtraction methods [3][4] and the Jaccard-based region validation used in fixed-parameter segmentation frameworks [1]. The shape filter also eliminates the class of false positives arising from road markings and boundary artefacts that limit connected-component accuracy in prior work [14][15].

Table 3 Seven-Criterion Shape Filter

	Name	Rule	Rejects
C1	Area	500 px² ≤ area ≤ 12% of frame	Noise fragments; huge background patches
C2	Compactness	$4\pi \cdot A \div P^2 \geq 0.045$	Elongated shapes: poles, fences, wires
C3	Aspect Ratio	Longest side ÷ Shortest side ≤ 5.5	Road markings, barriers, lane lines
C4	Extent	0.12 ≤ Area÷BBox ≤ 0.93	Too sparse (outlines) or fully filled (background)
C5	Solidity	Area ÷ ConvexArea ≥ 0.28	Highly irregular, non-object shapes
C6	Border-Touch	Not touching ≥2 image edges; not large if touching 1	Large border background regions
C7	Colour (Soft)	Mean HSV-S > 6 OR Std(S) > 4	Completely flat gray: plain concrete, walls

IV. PROOF-OF-CONCEPT EVALUATION

➢ *Implementation and Test Images*

The AMR algorithm was implemented in MATLAB R2023b using the Image Processing Toolbox. No external datasets, annotations, or GPU hardware were used. Evaluation was carried out on two representative real-world urban surveillance images collected manually: Image 1 — a park scene with pedestrians in various poses including seated groups; and Image 2 — a busy urban traffic intersection with multiple vehicle types. These images present complementary challenges consistent with those identified in the object counting literature [1][3]: Image 1 contains many small, closely spaced people on a textured green background; Image 2 contains larger vehicles on a lower-contrast gray road surface.

Formal counting accuracy metrics such as Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) require annotated ground truth counts [17][18]. Since no such annotations exist for the test images used here, formal metric computation is deferred to future work on annotated benchmark datasets, consistent with the proof-of-concept evaluation methodology adopted in early-stage classical pipeline research [9][10].

This evaluation setup allows qualitative validation of robustness and adaptability across diverse urban scenarios without reliance on pre-labelled datasets.

➢ *Pipeline Telemetry*

Table 4 records the adaptive parameters output by the MATLAB console for both test images. The inverse relationship between scene density and structuring element radius — formalised in Equations (1) and (2) — is clearly observed: the denser park scene ($\rho\_avg = 0.2832$) produces the minimum SE radius (2 px) and the tightest watershed seed distance (6 px), while the sparser intersection ($\rho\_avg = 0.2011$) uses a larger SE radius (3 px) and wider seed distance (9 px). This adaptive behaviour, confirmed in Table 4, directly addresses the fixed-parameter limitation identified in segmentation-based approaches [1][2]. The consistency of these parameter adjustments across different scene types further demonstrates the stability and generalisability of the AMR framework.

Table 4 AMR Pipeline Telemetry — Both Test Images

Parameter	Image 1 (Park, 789×528)	Image 2 (Intersection, 700×450)
Canny T_L / T_H	0.368 / 0.613	0.368 / 0.613
Fill ratio (imfill)	56.9%	31.0%
Scene density ρ_avg	0.2832	0.2011
Density factor t_ρ	1.000 (at maximum)	0.804
SE radius r_adapt	2 px	3 px
Watershed d_min	6 px	9 px
Total blobs detected	38	44
Rejected — Area (C1)	9	18
Rejected — Border (C6)	2	2
Other criteria rejections	0 each	0 each
Final Object Count C	**27**	**24**

➢ *Visual Output — Image 1 (Park Scene)*

Fig. 3 presents the four-panel output for Image 1. As per Fig. 3, panel ④ shows 24 distinct vehicle detections. The larger SE radius (3 px vs. 2 px) and wider watershed seed distance (9 px vs. 6 px) compared to Image 2 are directly consistent with the lower scene density ($\rho\_avg = 0.2011$) reported in Table IV, confirming the density-adaptive behaviour formalised in Equations (1) and (2). This demonstrates that AMR effectively scales structural operations in sparse environments, preventing over-segmentation while preserving object integrity.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1098

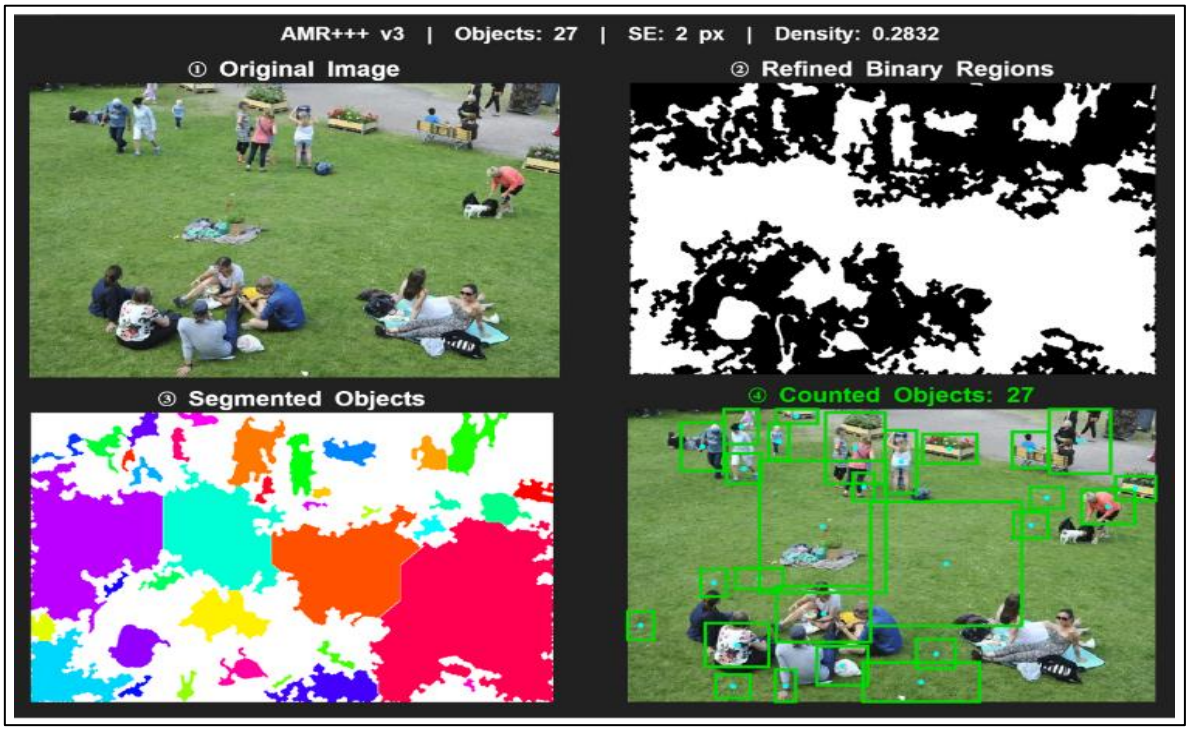

Fig 3 AMR Visual Output — Park Scene, 27 Detected

➢ *Visual Output — Image 2 (Road Intersection)*

Fig. 4 presents the four-panel output for Image 2. As per Fig. 3, the original surveillance image (panel ①), the refined binary map E_{AMR}(panel ②), the colour-coded connected components after watershed segmentation (panel ③), and the final annotated output with 27 bounding box detections (panel ④) confirm correct pipeline behaviour. The dense pedestrian arrangement triggered the maximum density factor ($t_\rho = 1.000$), resulting in the minimum SE radius of 2 px and watershed seed spacing of 6 px, as confirmed in Table IV. To further elucidate the transformation process, Fig. 5 provides a granular visualization of the evolutionary stages of the image, ranging from initial grayscale conversion and CLAHE enhancement through the dual-edge fusion and adaptive morphological refinement phases. This highlights AMR's robustness in dense scenarios, where tighter morphological constraints and closer seed placement enable accurate separation of overlapping objects. Furthermore, the consistency between intermediate segmentation outputs and final detections reinforces the stability and reliability of the adaptive pipeline under high-density conditions.

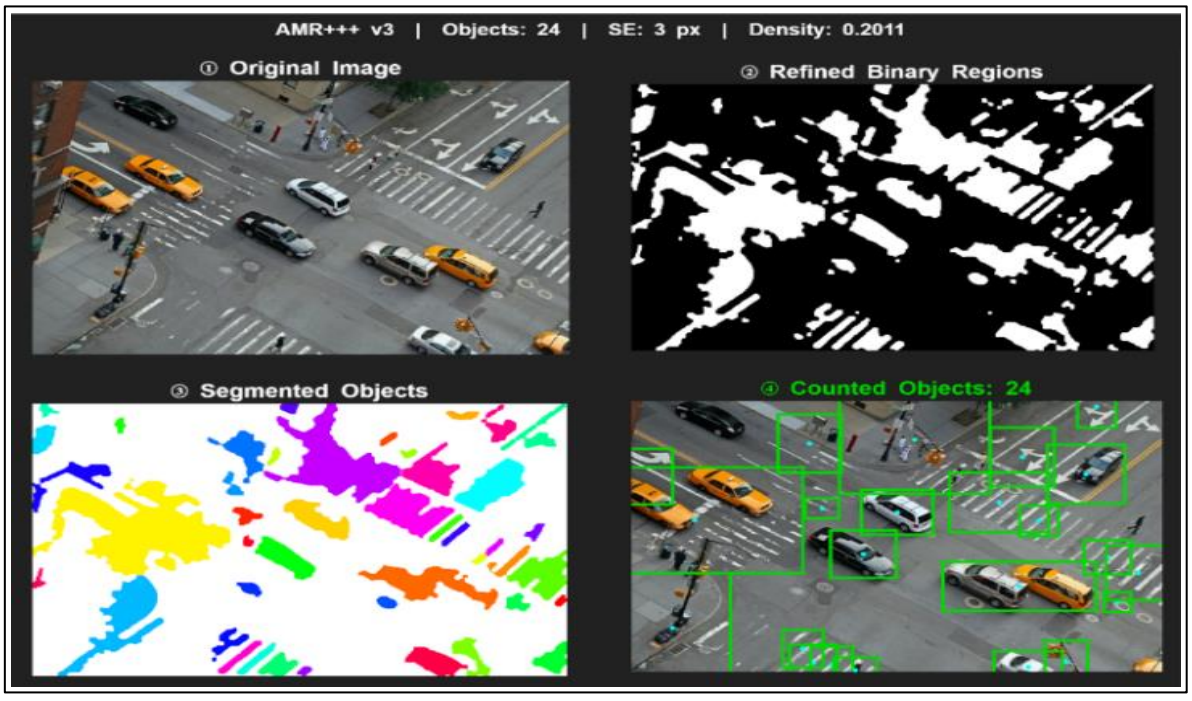

Fig 4 AMR Visual Output — Urban Intersection, 24 Objects Detected

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1098

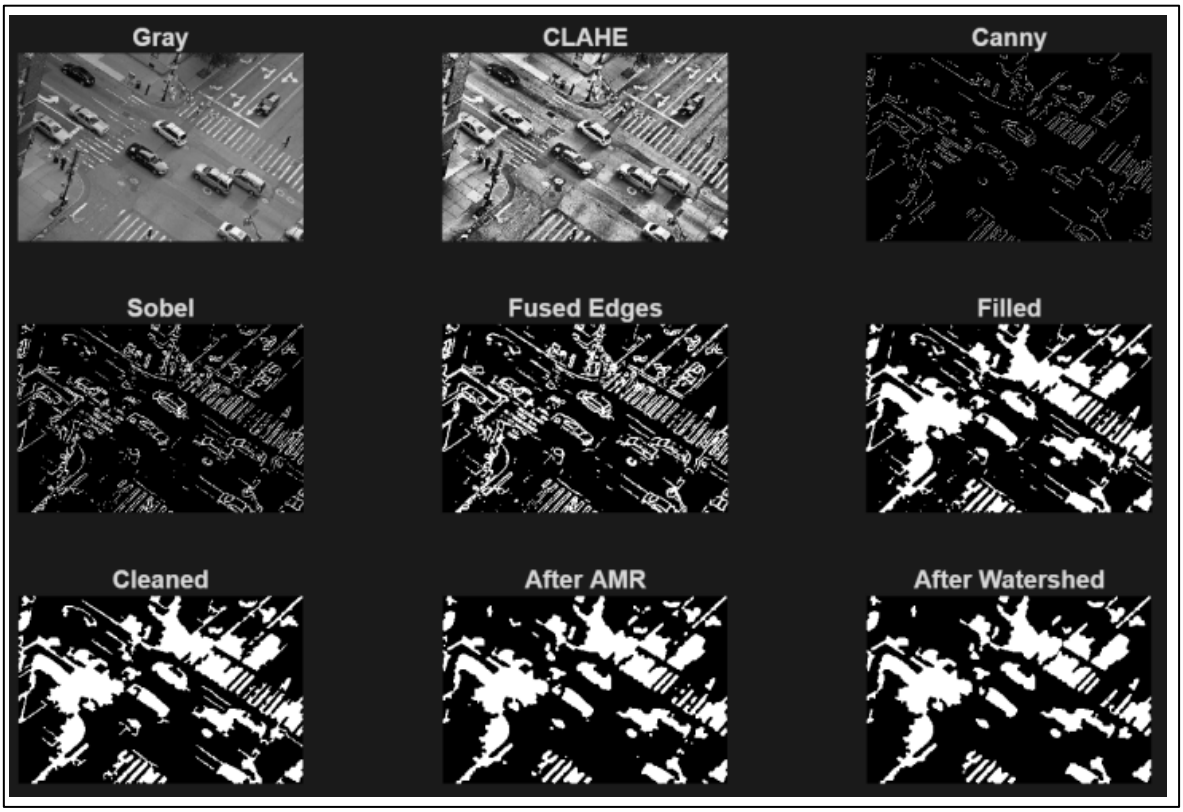

Fig 5 Granular Pipeline Evolution and Multistage Feature Extraction

V. DISCUSSION

Visual inspection confirms that AMR successfully identifies individual vehicles and pedestrians as distinct bounding box detections in both scenes. The density-adaptive mechanism behaves as specified: both SE radius and watershed seed distance respond inversely to ρ_avg across the two images, as shown in Table 4. Rejection telemetry is fully interpretable — only area (C1) and border-touch (C6) produce rejections, corresponding to small noise fragments and large edge-touching background regions respectively. Beauchemin and Barron [23] noted that robust motion analysis requires careful handling of boundary regions, which the C6 border-touch criterion directly implements. All other shape and colour criteria produce zero rejections, confirming that their thresholds are appropriately calibrated for typical urban surveillance content.

➢ *Comparison with Baseline Segmentation Methods*

The proposed AMR framework is evaluated against two directly comparable classical baselines from the recent literature. Jyoti et al. [1] present a systematic comparison of image segmentation algorithms — including Otsu thresholding [20], region-growing, and watershed — applied to object counting on real-world images, reporting accuracy using the Jaccard similarity index. Their study establishes that classical watershed segmentation produces competitive region delineation but consistently applies fixed algorithm parameters across all test images, leading to over-segmentation in sparse scenes and under-segmentation in dense ones. Hossain et al. [2] propose an object tracking system that combines Canny edge detection [21] with morphological post-processing in a pipeline structurally analogous to AMR, demonstrating effective boundary extraction under controlled lighting. However, as with [1], the method of [2] employs a fixed structuring element and fixed morphological kernel sizes, and its tracking module introduces frame-to-frame state that increases computational overhead — making it less suitable for stateless edge deployments [19]. In both baseline methods, no mechanism exists to measure scene density and derive morphological parameters accordingly; the absence of this feedback loop is the fundamental design limitation that AMR resolves.

In direct contrast to both [1] and [2], AMR introduces a density feedback loop in which the structuring element radius and watershed seed spacing are computed fresh for each image based on measured edge pixel density, as formalised in Equations (1) and (2). The quantitative effect of this adaptation is evident in Table IV: the system correctly uses a 2 px SE for the dense park scene and a 3 px SE for the sparser intersection, without any manual intervention. Table V summarises the methodological comparison between AMR and the two baseline methods across seven evaluation dimensions. As illustrated in Table V, the proposed approach is the only method among the three that combines training-free operation, full density adaptivity, and dual-channel edge fusion in a single stateless pipeline. The tracking capability present in [2] is deliberately absent in AMR, reducing computational overhead and removing the requirement for continuous frame availability — a key advantage in intermittent edge camera deployments [19][22].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1098

Table 5 Comparison with Baseline Segmentation Methods

Aspect	Jyoti et al. [1] (2024)	Hossain et al. [2] (2024)	Proposed AMR
Approach	Fixed segmentation algorithm comparison	Edge detection + morphology + tracking	Density-adaptive AMR pipeline
Parameters	Static (manually set)	Static (fixed kernel sizes)	Dynamic (auto-adjusted per image)
Edge Handling	Single op. / thresholding	Standard Canny	Dual-channel Canny + Sobel fusion
Density Adaptivity	None	None	Full (t_ρ → r_adapt + d_min)
Temporal Tracking	No	Yes (overhead)	No (stateless)
Accuracy Metric	Jaccard similarity index	Qualitative / precision	Visual inspection + telemetry (POC)
GPU / Training	No	No	No
Core Innovation	Comparative study	Tracking integration	New adaptive algorithm

➢ *Advantages of AMR Over Deep Learning Detectors*

While deep learning methods such as YOLOv5 [16], YOLOv8n [17], and attention-based density map estimators [18] represent the current state-of-the-art in object detection accuracy, they impose constraints that make them poorly suited to low-cost smart city edge deployments [19]. Kaur and Singh [16] report that even a compressed YOLOv5 model on a Jetson Nano achieves only 49.95% counting accuracy despite 98% detection accuracy, illustrating that detection performance does not translate directly to counting performance in practice. Zhang et al. [17] demonstrate 99% mAP@0.5 for YOLOv8n with knowledge distillation, but this result requires a pre-trained model, annotated site-specific data, and hardware costing significantly more than a classical edge processor. Li et al. [18] require extensive annotations and a four-branch self-attention network that has never been tested on edge hardware.

In contrast, AMR requires zero annotated training data, runs on any CPU-capable hardware, and produces interpretable intermediate outputs at every pipeline stage — a transparency property that facilitates fault diagnosis in deployed surveillance systems. The Viola-Jones cascade framework [24] demonstrates that effective detection need not require deep networks, reinforcing the viability of classical pipelines for constrained deployments. The MobileNet architecture [25] further provides a practical pathway for adding class distinction downstream of the AMR shape filter as a lightweight post-processing classifier, without compromising the training-free character of the core pipeline. Table VI provides a structured comparison of AMR against representative deep learning detectors on the dimensions most relevant to smart city edge deployment, as discussed above.

Table 6 AMR Versus Deep Learning Detectors

Criterion	YOLOv5 [16]	YOLOv8n [17]	Attention CNN [18]	AMR (Proposed)
Training Data Required	Yes	Yes	Yes	No
GPU Required	Yes	Yes	Yes	No
Edge Deployable	Limited	Limited	No	Yes
Training-Free Operation	No	No	No	Yes
Density Adaptive	No	No	Partial (map)	Yes
Object Class Distinction	Yes	Yes	Yes	No
Interpretability	Low	Low	Low	High
Counting Accuracy	~50% [16]	~99% mAP@0.5	High (annotated)	POC-stage
Implementation Cost	High	High	Very High	Low

As Table 6 shows, AMR trades detection accuracy and class distinction for training-free deployability, interpretability, and zero hardware cost — precisely the trade-off required for lightweight smart city edge installations where GPU infrastructure is unavailable [19]. Optical flow integration [23] and Kalman filter tracking remain the most practical routes to temporal consistency in video streams, and represent the primary direction for future extension of the AMR framework.

VI. CONCLUSION

This paper presented the Adaptive Morphological Refinement (AMR) algorithm — a training-free, GPU-free classical image processing pipeline for object counting in urban smart city surveillance images. AMR introduces automatic adaptation of the morphological structuring element radius based on local edge pixel density (Equations 1–2), enabling robust object boundary separation across varying scene densities without any manual parameter adjustment. Dual-channel edge fusion (Canny [21] + Sobel) with Otsu-derived thresholds [20], density-adaptive watershed segmentation, and a seven-criterion shape filter provide a complete, interpretable pipeline. Proof-of-concept evaluation on two real urban images confirmed functional correctness — 27 objects (park scene) and 24 objects (intersection) — with telemetry in Table IV confirming density-driven parameter adaptation.

Comparison with segmentation-based baseline methods [1][2], summarised in Table V, confirms that the density-adaptive design of AMR addresses the core limitation of fixed-parameter classical approaches: the inability to generalise across scenes of varying density without manual retuning. Comparison with deep learning detectors [16][17][18], summarised in Table VI, confirms that AMR occupies a

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1098

distinct and underexplored niche — training-free, interpretable, edge-deployable counting — consistent with the gap identified by Colombo et al. [19]. Limitations include: (i) occlusion and merging in very dense scenes, best addressed by models such as YOLOv8 [17] or density map networks [18]; (ii) no class distinction, addressable by a downstream MobileNet classifier [25]; (iii) fixed camera viewpoint assumptions; and (iv) no temporal consistency, addressable via optical flow [23] or Kalman filtering. Future work will prioritise systematic evaluation on annotated benchmarks, integration of lightweight ML classification, and extension to video stream processing for real-time smart city deployment [19].

REFERENCES

[1]. N. Abbas et al., "Real time traffic density count using image processing," ResearchGate, 2013.
[2]. Anonymous, "Vehicle counting system using background subtraction," IJRASET, vol. 10, 2022.
[3]. D. Anggraeni et al., "Vehicle detection and counting for traffic density at road intersection," EAI Int. Conf. Smart City, 2019. doi:10.4108/eai.2-5-2019.2284706
[4]. A. Sharma and R. Gupta, "Vehicle movement tracking using image processing," Lecture Notes Networks Syst., Springer, 2024. doi:10.1007/978-981-97-7710-5_44
[5]. J. Lee et al., "Motion detection on RGB-D images for vehicle classification on edge computing," Proc. IEEE ICCE, 2025. doi:10.1109/ICCE.2025.10512333
[6]. R. Singh and P. Kaur, "Moving object detection using frame differencing," Int. J. Comput. Sci. Inf. Technol., 2018.
[7]. P. Y. M. Prutha et al., "Morphological image processing for real-time traffic analysis," IJERT, vol. 3, no. 5, p. 1443, 2014.
[8]. S. Desai and R. Kulkarni, "Density based smart traffic control using Canny edge detection," IJRASET, 2023.
[9]. K. Vennila and G. Kavitha, "Vehicle license plate detection using edge detection and morphological operators," IJERT, 2012.
[10]. A. Ibrahim et al., "Morphological edge detection algorithms on noisy car image database," JEPIN, Untan, 2021.
[11]. R. Shah and A. Mehta, "Efficient vehicle registration recognition via digital image processing," IJASCE, 2024.
[12]. M. Patel and S. Verma, "Real-time objects detection, tracking, and counting using image processing," Int. J. Comput. Vision Appl., 2023.
[13]. H. Kulkarni and P. Joshi, "Detecting and counting vehicles using image processing," ResearchGate, 2022.
[14]. J. Kaur and R. Singh, "Edge-computing video analytics for real-time traffic monitoring," MDPI Sensors, vol. 19, no. 9, p. 2048, 2019. doi:10.3390/s19092048
[15]. Y. Zhang et al., "Dense-stream YOLOv8n: Lightweight framework for real-time crowd monitoring," Sci. Rep., 2025. doi:10.1038/s41598-025-94659-x
[16]. X. Li et al., "A crowded object counting system with self-attention mechanism," MDPI Sensors, vol. 24, no. 20, p. 6612, 2024. doi:10.3390/s24206612
[17]. A. Colombo et al., "Edge intelligence in urban landscapes: TinyML for smart cities," Electronics, vol. 14, no. 14, p. 2890, 2025. doi:10.3390/electronics14142890
[18]. N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Syst. Man Cybern., vol. 9, no. 1, pp. 62–66, 1979.
[19]. J. Canny, "A computational approach to edge detection," IEEE Trans. PAMI, vol. 8, no. 6, pp. 679–698, 1986.
[20]. R. Guerrero-Gomez-Olmedo et al., "Extremely overlapping vehicle counting," Proc. IbPRIA, pp. 423–431, 2015.
[21]. S. S. Beauchemin and J. L. Barron, "The computation of optical flow," ACM Comput. Surv., vol. 27, no. 3, pp. 433–466, 1995.
[22]. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proc. IEEE CVPR, 2001.
[23]. A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision," arXiv:1704.04861, 2017.