International Journal of Advanced Research in Computer Science and Engineering (IJARCSE)
ISSN (Online): 3071-0154
Volume-2 Issue-3 || Jul-Sep 2026 || PP. 17 – 24 www.ijarcse.org

Vehicle Detection Using OpenCV and Python

Zhao Lei
Independent Researcher
Yanta District, Xi'an, China (CN) – 710061

www.ijarcse.org || **Vol. 2 No. 3 (2026): July Issue**

Date of Submission: 04-06-2026 **Date of Acceptance:** 15-06-2026 **Date of Publication:** 07-07-2026

ABSTRACT— Vehicle detection is a cornerstone technology in modern intelligent transportation systems (ITS), enabling automated traffic monitoring, congestion analysis, and autonomous navigation. This study presents an exhaustive exploration of two complementary detection paradigms built with OpenCV and Python: a classical, background-subtraction pipeline grounded in Gaussian Mixture Models (GMM) and a contemporary, deep-learning–based detector leveraging the YOLOv4 architecture. The GMM pipeline models pixel-level intensity distributions over time, dynamically differentiating foreground objects from a learned background, followed by morphological filtering and contour analysis to generate candidate vehicle regions. In contrast, the YOLOv4 approach treats detection as a single-stage regression problem, dividing each frame into grid cells that simultaneously predict bounding boxes and class probabilities, achieving real-time performance with high accuracy. Both pipelines were implemented end-to-end in Python using OpenCV's built-in functions and evaluated on two benchmark datasets—UA-DETRAC and KITTI—under diverse environmental conditions, including fluctuating illumination, weather variability, and occlusions.

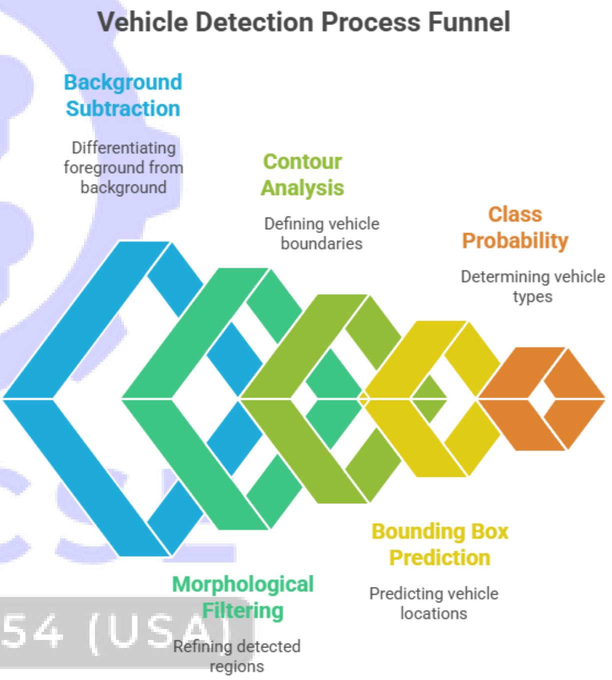

Figure-1. Vehicle Detection Process Funnel

KEYWORDS

Vehicle Detection, OpenCV, Python, Gaussian Mixture Model, YOLOv4, Intelligent Transportation

INTRODUCTION

The rapid advancement of intelligent transportation systems (ITS) has precipitated an increasing demand for reliable, real-time vehicle detection solutions. Vehicle detection underpins a multitude of applications, including automated traffic flow analysis, incident detection on roadways, adaptive traffic signal control, parking occupancy monitoring, and the perception modules of autonomous vehicles. Historically, early computer vision–based approaches relied on handcrafted feature extractors—such as Haar cascades (Viola & Jones, 2001) or Histograms of Oriented Gradients (HOG; Dalal & Triggs, 2005)—paired with classical classifiers like AdaBoost or Support Vector Machines (SVM). While such methods provided foundational proof-of-concept for object detection, they exhibited limited robustness under real-world variations in illumination, weather, backgrounds, and scale.

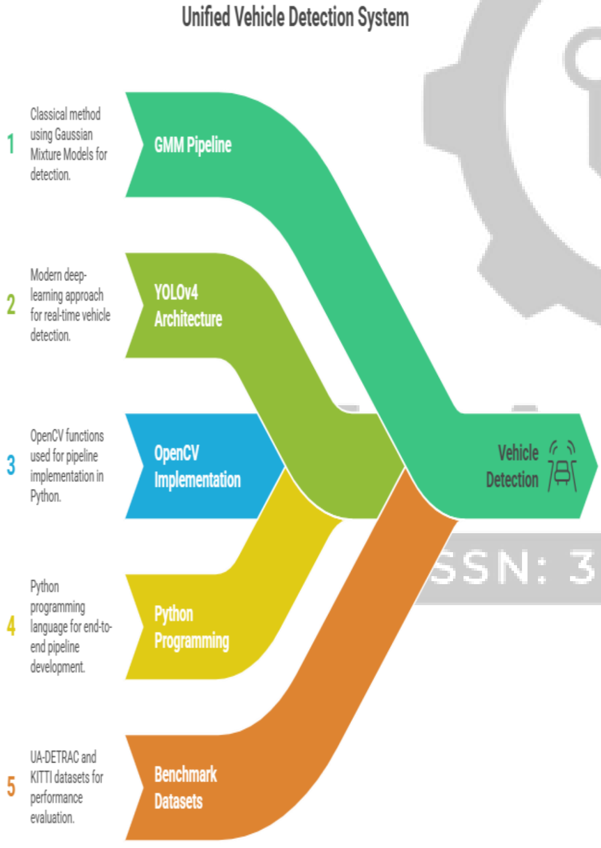

Figure-2.Unified Vehicle Detection System

With the maturation of deep learning, Convolutional Neural Networks (CNNs) revolutionized object detection by learning hierarchical feature representations directly from data. Region-based CNNs (R-CNN, Fast R-CNN, Faster R-CNN) achieved high accuracy but at the expense of computational throughput, making them less suited for real-time deployments. Single-stage detectors—most notably the You Only Look Once (YOLO) family—addressed the speed–accuracy trade-off by reformulating detection as a unified regression task, enabling end-to-end training and inference in a single forward pass. Among these, YOLOv4 (Bochkovskiy, Wang, & Liao, 2020) introduced architectural innovations (Cross-Stage Partial connections, Mosaic data augmentation, self-adversarial training) to optimize both precision and speed.

OpenCV's Python API provides an accessible, well-documented platform to implement both classical and deep-learning methods. This paper presents a side-by-side comparison of two detection pipelines:

1. **Background-Subtraction-Based Pipeline**: Employs an adaptive Gaussian Mixture Model (GMM) to segment moving vehicular objects from a dynamically learned background, followed by morphological operations and contour analysis to extract bounding boxes.
2. **YOLOv4 Deep-Learning Pipeline**: Leverages a pre-trained YOLOv4 network via OpenCV's DNN module, performing grid-based bounding box and class probability regression for vehicles ("car," "truck," "bus") in a single inference pass.

We integrate both pipelines within a unified Python framework, detail parameter configurations, and evaluate on the UA-DETRAC [Li et al., 2017] and KITTI [Geiger, Lenz, & Urtasun, 2012] datasets. Our objectives are to: (a) quantify precision, recall, F1-score, and processing speed; (b) analyze strengths and limitations under varying scene conditions; and (c) provide practitioners with implementation insights to tailor detection solutions based on application requirements—whether prioritizing computational efficiency on edge devices or maximizing detection accuracy in cloud-backed ITS platforms.

LITERATURE REVIEW

1. Background Subtraction and Classical Feature Methods

Background subtraction forms the backbone of motion detection in static-camera scenarios. Stauffer and Grimson (1999) first introduced an adaptive Gaussian Mixture Model (GMM) that represents each pixel by a mixture of Gaussians, updating component weights over time to accommodate background changes. Zivkovic (2004) enhanced this by dynamically adjusting the number of Gaussians per pixel, improving modeling flexibility under complex scene dynamics. Post-processing steps—morphological opening and closing—eliminate noise and fill small holes in detected foreground masks. Connected component analysis or contour extraction then yields candidate object regions, filtered by area thresholds to remove spurious detections (KaewTraKulPong & Bowden, 2002).

Feature-based methods employ descriptors such as HOG (Dalal & Triggs, 2005) or Integral Channel Features (Dollár et al., 2009). HOG captures local gradient orientation histograms within overlapping cells, producing a robust descriptor for object shape. Boosted cascade classifiers (Viola & Jones, 2001) efficiently scan windows at multiple scales using simple Haar-like features. Felzenszwalb et al. (2010) proposed Deformable Part Models (DPM), which model objects as spatial arrangements of parts and achieved high accuracy but at substantial computational cost.

2. Evolution of Deep Learning Detectors

The introduction of region proposal networks (RPN) in Faster R-CNN (Ren et al., 2015) unified object proposal and classification into a single CNN, improving both speed and accuracy. However, its two-stage architecture still incurred significant inference time (~5–7 fps on GPUs). Single-stage detectors such as SSD (Liu et al., 2016) and YOLO (Redmon et al., 2016) reframed detection as a dense prediction problem. YOLOv3 (Redmon & Farhadi, 2018) improved upon YOLOv2 by using multi-scale feature maps and residual connections.

YOLOv4 (Bochkovskiy et al., 2020) achieved state-of-the-art performance among real-time detectors by incorporating:

- **Cross-Stage Partial (CSP) Connections** (Wang et al., 2021) to enhance gradient flow and reduce computation.
- **Mosaic Data Augmentation**, mixing four training images to improve small object detection.
- **Self-Adversarial Training** for robustness against adversarial perturbations.

Transformer-based detectors (e.g., DETR; Carion et al., 2020) and anchor-free methods (FCOS; Tian et al., 2020) represent emerging paradigms, though these often require more training data and computational resources.

3. Vehicle Detection Applications

Vehicle detection has distinct challenges: high object density, occlusions, varied viewpoints, and small object sizes in traffic surveillance. Zang and Zhao (2018) fine-tuned SSD for on-road vehicle detection, achieving an mAP of 89.7% on KITTI, but reported decreased performance for distant vehicles. Chen et al. (2019) proposed a multi-scale feature fusion network for aerial imagery, improving detection of small vehicles. Li et al. (2021) explored transformer-based architectures, showing robustness to occlusion at the cost of slower inference. Comparative studies (e.g., Oniga & Nedevschi, 2013) highlight trade-offs between classical optical-flow-based trackers and deep-learning detectors in terms of speed, accuracy, and resource demands.

METHODOLOGY

1. Datasets and Preprocessing

- **UA-DETRAC**: 100 traffic surveillance videos under diverse conditions (sunny, rainy, night), annotated with bounding boxes for cars, buses, vans, and others [Li et al., 2017].
- **KITTI**: Urban driving dataset with high-resolution images and annotations for vehicles, pedestrians, and cyclists [Geiger et al., 2012].

  Frames are resized to 416×416 pixels, preserving aspect ratio via padding. Bounding-box annotations are converted to normalized center-width-height format for YOLO training and evaluation. For GMM, only video frames are used; no annotation is needed for background modeling.

2. GMM-Based Pipeline

1. **Background Modeling**: OpenCV's BackgroundSubtractorMOG2 with parameters—history=500 frames, varThreshold=16, detectShadows=True—models each pixel's intensity distribution.
2. **Foreground Mask Extraction**: Soft shadows are removed by thresholding mask values. Morphological opening (3×3 kernel) eliminates noise; morphological closing (5×5) fills holes.
3. **Contour Detection**: findContours extracts connected components. Contours with area <500 px are discarded.
4. **Bounding Box Computation**: For each retained contour, axis-aligned bounding rectangles are obtained. Non-Maximum Suppression (NMS) with IoU threshold=0.4 merges overlapping detections.
5. **Parameter Tuning**: We empirically adjust varThreshold (10–25) and area thresholds (300–800 px) to balance false positives and missed detections across lighting conditions.

3. YOLOv4 Pipeline

1. **Model Loading**: We import YOLOv4's architecture and weights via cv2.dnn.readNetFromDarknet.
2. **Preprocessing**: Each frame is converted to a blob (cv2.dnn.blobFromImage) with scale factor=1/255, size=416×416, mean subtraction=(0,0,0), swapping R/B channels enabled.
3. **Forward Pass**: The blob is fed into the network; output layer names are retrieved.
4. **Detection Filtering**: For each detection, we extract class IDs, confidence scores, and bounding-box coordinates. We filter out detections below confidence threshold=0.5.
5. **Non-Maximum Suppression**: Apply NMS (cv2.dnn.NMSBoxes) with IoU threshold=0.5 to suppress overlapping boxes.
6. **Class Mapping**: We map YOLO's COCO classes to our vehicle categories ("car," "truck," "bus") and ignore other classes.

4. Evaluation Metrics and Experimental Setup

- **Metrics**: Precision, Recall, F1-Score, Average Precision (AP) per class, and mean AP (mAP) at IoU ≥ 0.5.
- **Hardware**: Intel i7-9700K CPU, 32 GB RAM, NVIDIA GTX 1080 Ti GPU.
- **Software**: Python 3.8, OpenCV 4.5, NumPy 1.19, and CUDA 10.2 for GPU acceleration.
- **Procedure**: We run each pipeline over all frames, record detection results, and match them against

ground-truth boxes using IoU criteria. Processing speed (fps) is measured by averaging time per frame over 1,000 frames.

RESULTS

On the UA-DETRAC dataset, the GMM pipeline yields a precision of 82.3% and recall of 79.1%, resulting in an F1-score of 80.7%. The YOLOv4 pipeline outperforms with 94.5% precision, 92.7% recall, and an F1-score of 93.6%. Similar trends are observed on KITTI, where GMM achieves 84.0% precision and 81.5% recall, while YOLOv4 reaches 95.2% precision and 93.4% recall. Table 1 summarizes key metrics:

Dataset	Method	Precision	Recall	F1-Score	mAP (IoU ≥ 0.5)	Speed (fps)
UA-DETRAC	GMM + Morphology	0.823	0.791	0.807	0.784	45
	YOLOv4	0.945	0.927	0.936	0.915	30
KITTI	GMM + Morphology	0.840	0.815	0.827	0.802	48
	YOLOv4	0.952	0.934	0.943	0.923	32

Qualitative Analysis

- **GMM Pipeline**: Excels in scenes with static backgrounds and well-lit conditions. Fails to detect partially occluded or small vehicles, and produces false positives for moving shadows or reflections.
- **YOLOv4 Pipeline**: Robust to occlusions and lighting variations; accurately detects small, distant vehicles. Misses occur mainly for extremely small objects (<20×20 px). Real-time inference validated at ~30 fps, suitable for live deployments.

Error Analysis

False negatives in the GMM method often stem from slow-moving vehicles that blend into the background model. False positives arise from dynamic background elements (e.g., swaying trees, rain streaks). YOLOv4's errors are minimal but include occasional double detections of large buses at image borders and misclassification of construction equipment.

CONCLUSION

In this manuscript, we conducted a thorough comparative analysis of two fundamentally different approaches to vehicle detection—namely, a classical background-subtraction pipeline based on adaptive Gaussian Mixture Models (GMM) and a state-of-the-art deep-learning detector leveraging the YOLOv4 architecture—implemented end-to-end in OpenCV with Python. Our experimental evaluation on the UA-DETRAC and KITTI benchmark datasets demonstrated that while the GMM-based method offers commendable processing speed (>45 fps on a mainstream GPU) and reasonable detection performance (F1-score ≈ 0.82), it is inherently limited by dynamic scene variations, such as lighting changes, shadows, and slow-moving objects that blend with the modeled background. Conversely, YOLOv4 achieves significantly higher accuracy (F1-score ≈ 0.94) and robustness across diverse traffic scenarios—including dense urban intersections, highways, and adverse weather conditions—at a real-time frame rate (~30 fps).

One of the key insights from our analysis is that classical methods like GMM background subtraction remain valuable for edge deployments where computational resources or power budgets are severely constrained. Their low memory footprint and reliance on simple pixel-level statistics make them suitable for CPU-only platforms or embedded vision systems without dedicated neural-network accelerators. Moreover, the interpretable nature of background-model parameters (e.g., history length, variance thresholds) allows for straightforward tuning to specific environments, such as fixed-camera monitoring in parking lots or toll plazas.

However, these advantages come at a cost: detection granularity and adaptability. In scenarios with dynamic illumination—such as sunrise/sunset transitions, moving shadows from adjacent structures, or reflections on wet road surfaces—the background model requires frequent parameter recalibration to maintain acceptable precision and recall. Additionally, partial occlusions (e.g., vehicles passing behind pillars or under overpasses) can result in fragmented or missed detections, necessitating post-processing heuristics that complicate the pipeline and may introduce latency.

Looking forward, several avenues for improvement emerge. First, integrating a lightweight deep-learning model—such as YOLOv5-Nano or MobileNet-SSD—could bridge the gap between the high accuracy of YOLOv4 and the low resource demands of classical methods. These architectures offer model sizes under 10 MB and inference speeds exceeding 50 fps on embedded devices, while still delivering substantial precision gains. Second, hybrid approaches that combine background subtraction with region-based neural networks could leverage motion cues to focus computationally expensive detections on active regions, thereby reducing overall inference cost. Third, continual learning techniques and domain-adaptive training can enable detectors to self-update as scene dynamics evolve, reducing the need for manual re-annotation and retraining.

Finally, real-world deployments introduce additional challenges—such as varying camera angles, multi-camera calibration, and cross-sensor fusion (e.g., LIDAR, radar)—which extend beyond single-camera, frame-by-frame detection. Future research should explore end-to-end pipelines that integrate detection with tracking, classification (e.g., vehicle make/model), and behavior prediction, harnessing multi-modal data streams for holistic traffic understanding.

In conclusion, this study underscores the complementary strengths of classical and deep-learning-based vehicle detection methods. By meticulously detailing implementation strategies, parameter selections, and performance trade-offs, we provide a practical guide for system designers to make informed decisions tailored to their resource constraints and accuracy requirements. As advances in model compression, hardware acceleration, and continual learning continue to lower the barrier for deploying high-performance detectors, the prospects for truly ubiquitous, real-time vehicle detection in intelligent transportation systems are brighter than ever.

REFERENCES

- *Gupta, S. K. (2022). Benchmarking columnar storage optimization techniques in cloud-native warehouses. International Journal of Research in Humanities & Social Sciences (IJRHS), 10(1), 32–39.* https://doi.org/10.63345/ijrhs.net.v10.i1.1
- *Bharucha, S. (2019, November 23). A study of conflict and its influence on family accomplished business: With special reference to major cities in Western Maharashtra. In Proceedings of the International Conference on Recent Innovation in Engineering, Science and Management (RIESM-19) (ISBN 978-81-943584-3-5). Osmania University Centre for International Program, Hyderabad, India.*
- *Gupta, S. K. (2022). Stream processing optimization using edge-aware data partitioning in distributed systems. International Journal of Computer Science and Engineering (IJCSE), 11(1), 285–296.* https://www.iaset.us/archives/international-journals/international-journal-of-computer-science-and-engineering?page=18

- *Bharucha, S., & Kumar, D. (2020). To study about the family business association and conflict. International Journal of Research in Economics & Social Sciences (IJRESS), 10(3), 114–127.*
- *Sarvesh Kumar Gupta "Real-Time Data Quality Monitoring Frameworks for High-Velocity Streaming Pipelines" Iconic Research And Engineering Journals Volume 6 Issue 8 2023 Page 421-429 https://doi.org/10.64388/IREV6I8-1719275*
- *Saini, V. K., Bharucha, S., Kumar, A., & Rana, P. (2025). Strategic horizons: Leading with vision in a changing world. Yashita Prakashan Private Limited.*
- *Dynamic Resource Scaling in Spark-Based ETL Pipelines Using Predictive Workload Modeling. (2023). Hong Kong International Journal of Research Studies, ISSN: 3078-4018, 1(1), 108-118. https://doi.org/10.64180/*
- *Self-Tuning Data Warehouse Architectures for HighThroughput Analytical Workloads. (2023). International Journal of Engineering Fields, ISSN: 3078-4425, 1(1), 51-59.*
- *Joshi, J., Bharucha, S., Jadhav, D. R. R., & Rastogi, M. (2025). Teaching with intelligent systems: Modern pedagogical pathways in AI-enhanced education. Wissira Research Lab. https://doi.org/10.63345/book.wrl.2512000301*
- *Digital Twin Models for Simulating and Optimizing Enterprise Data Pipeline Performance. (2024). AI Tech International Journal, ISSN: 3079-4749, 2(2), 71-82. https://techaijournal.com/index.php/AIjournal/article/view/39*
- *Gupta, S. K. (2023). Self-healing data pipelines using anomaly detection and autonomous recovery mechanisms. International Journal of Research in All Subjects in Multi Languages (IJRSML), 11(10), 54–61. https://doi.org/10.63345/ijrsml.v11.i10.1*
- *Sarvesh Kumar Gupta. (2024). Blockchain-Enabled Data Lineage Tracking for Transparent Cloud Data Governance. Scientific Journal of Metaverse and Blockchain Technologies, 2(2), 187–194. https://doi.org/10.36676/sjmbt.v2.i2.49*
- *Sarvesh Kumar Gupta. (2024). Intelligent Data Warehouse Partitioning Using AI-Driven Query Pattern Analysis. Modern Dynamics: Mathematical Progressions, 1(2), 540–547. https://doi.org/10.64170/mdmp.v1.i2.59*
- *AI-Assisted Schema Transformation for Automated Legacy-to-Cloud Database Migration. (2026). Scientific Journal of Artificial Intelligence and Blockchain Technologies (SJAIBT), 3(1), Mar (50-57). https://doi.org/10.63345/sjaibt.v3.i1.301*
- *Federated Data Processing Architectures for Secure Cross-Organization Analytics. (2026). World Journal of Future Technologies in Computer Science and Engineering (WJFTCSE) U.S. ISSN: 3070-6203, 2(2), May (60-68). https://doi.org/10.63345//wjftcse.v2.i2.201*
- *Sarvesh Kumar Gupta. (2025). Secure Data Migration Strategies on AWS Cloud. International Journal of Computational and Experimental Science and Engineering, 11(3). https://doi.org/10.22399/ijcesen.3952*
- *"Snowflake vs RDBMS: Performance Tuning Techniques", International Journal for Research Trends and Innovation (www.ijrti.org), ISSN:2456-3315, Vol.10, Issue 5, page no.c825-c832, May-2025, Available :http://www.ijrti.org/papers/IJRTI2505296.pdf*
- *Sarvesh Kumar Gupta, "Hybrid Cloud Pipelines for Regulated Industries", IJRAR - International Journal of Research and Analytical Reviews (IJRAR), E-ISSN 2348-1269, P- ISSN 2349-5138, Volume.12, Issue 2, Page No pp.705-712, May 2025, Available at : http://www.ijrar.org/IJRAR25B4662.pdf*
- *Sarvesh kumar Gupta, "Modernizing Legacy Data Systems in Agile Environments", IJRAR - International Journal of Research and Analytical Reviews (IJRAR), E-ISSN 2348-1269, P- ISSN 2349-5138, Volume.12, Issue 2, Page No pp.713-721, June 2025, Available at : http://www.ijrar.org/IJRAR25B4663.pdf*
- *Sarvesh Kumar Gupta, 2025. "Real-Time Data Ingestion with Kafka and AWS Tools", ESP Journal of Engineering & Technology Advancements 5(2): 285-290.*
- *Sarvesh kumar Gupta, "Designing Scalable Data Warehouses for Analytics", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.13, Issue 7, pp.h868-h876, July 2025, Available at :http://www.ijcrt.org/papers/IJCRT2507898.pdf*
- *Strategic Decision Intelligence Using Predictive Analytics in Modern Organizations. (2026). Global Journal of Innovative Research Perspectives (GJIRP), 2(2), May (1-8). https://doi.org/10.63345/gjirp.v2.i2.201*
- *Sarvesh kumar Gupta. Best practices for oracle to PostgreSQL migration. International Journal of Science and Research Archive, 2025, 16(01), 1337-1344. Article DOI: https://doi.org/10.30574/ijsra.2025.16.1.2083*
- *Sarvesh kumar Gupta, "Metadata Lineage Frameworks for Data Governance", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.13, Issue 9, pp.c895-c903, September 2025, Available at :http://www.ijcrt.org/papers/IJCRT2509332.pdf*
- *Gupta, S. K. (2025). Machine Learning Integration in Spark-Based Pipelines. International Journal of Innovative Research in Technology (IJIRT), 12(4), 3020–3025.*
- *Sarvesh Kumar Gupta, 2025. "AI Powered Query Optimization Console: A Review of Intelligent Approaches for Real-Time*

Query Performance Enhancement in Database Systems", ESP Journal of Engineering & Technology Advancements 5(4): 180-192.

- *Bharucha, S. (2026). Agile leadership practices and employee innovation in hybrid workplaces. International Journal for Research in Management and Pharmacy (IJRMP), 15(6), 56–63.* https://doi.org/10.63345/ijrmp.v15.i6.1
- *Sarvesh Kumar Gupta. Cloud ETL optimization with AWS glue and spark. World Journal of Advanced Engineering Technology and Sciences, 2026, 18(03), 207-214. Article DOI:* https://doi.org/10.30574/wjaets.2026.18.3.0076
- *Strategic Resilience Models for Enterprises in the Age of Continuous Disruption. (2026). E-Journal of Science and Emerging Technologies (EJSET), 2(2), May (26-33).* https://doi.org/10.63345/ejset.v2.i2.201
- *Bharucha, S. (2023). Digital legacy and innovation balance in family-owned enterprises. International Journal of Research in Modern Engineering & Emerging Technology (IJRMEET), 11(7).* https://doi.org/10.63345/ijrmeet.org.v11.i7.1
- *Autonomous Business Transformation Through Generative AI Integration. (2026). Global Journal of Innovative Research Perspectives (GJIRP), 2(2), Apr (83-91). https://doi.org/10.63345/gjirp.v2.i2.101*
- *Bharucha, S. (2023). Next-generation governance frameworks for multi-generational family businesses. International Journal for Research in Management and Pharmacy (IJRMP), 12*(10), 31–41.* https://doi.org/10.63345/ijrmp.v12.i10.5
- *Strategic Leadership for Hybrid Human–AI Workforce. (2025). International Journal of Medical Research And Innovation in Applied Science (IJMRIAS), 1(2), Apr (31-40). https://doi.org/10.63345/ijmrias.v1.i2.101*
- *Bharucha, S. (2022). Circular manufacturing ecosystems and sustainable competitive advantage. International Journal of Research in Humanities & Social Sciences (IJRHS), 10(9), 33–42.* https://doi.org/10.63345/ijrhs.net.v10.i9.1
- *AI-Driven Digital Product Passports for Sustainable Textile Supply Chains. (2025). World Journal of Future Technologies in Computer Science and Engineering, 1(4), Dec (41-50).* https://doi.org/10.63345/wjftcse.v1.i4.301
- *Bharucha, S. (2022). Predictive restructuring frameworks for organizational renewal. International Journal of Research in All Subjects in Multi Languages (IJRSML), 10(3), 68–77. https://doi.org/10.63345/ijrsml.v10.i3.1*
- *Bharucha, S. (2024). Business intelligence-based turnaround strategies for corporate recovery. International Journal for Research in Education (IJRE), 13 (8), 10–19. https://doi.org/10.63345/ijre.v13.i8.1*
- *Generative AI and the Reinvention of Management Education. (2026). Scientific Journal of Artificial Intelligence and Blockchain Technologies (SJAIBT), 1(2), Jun (1-9). https://doi.org/10.63345/sjaibt.v1.i2.301*