International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 14, Issue 12, December 2025)

Design and Implementation of a Real-Time Object Detection and Counting System Using YOLOv5

D.Gangadhar[1], Tulasi Miriyala[2]

[1,2] Associate Professor, AQJ Centre For PG Studies, Gudilova Anandapuram, Visakhapatnam, India

***Abstract*— The rapid growth of computer vision applications has increased the demand for accurate and real-time object detection systems across various domains. This paper presents the design and implementation of a real-time object detection and counting system using YOLOv5, a state-of-the-art deep learning model known for its high detection speed and accuracy. The proposed system is capable of detecting and counting multiple objects in both static images and live video streams obtained from camera feeds. The system employs a pretrained YOLOv5 model trained on the COCO dataset to identify and localize objects efficiently under challenging conditions such as occlusion, varying illumination, and object overlap. To enhance performance for domain-specific applications, transfer learning techniques are utilized, enabling the model to adapt to customized object classes with minimal retraining. The detection pipeline provides bounding boxes, confidence scores, and real-time object counts displayed through an interactive user interface. Experimental results demonstrate that the proposed system achieves reliable detection accuracy while maintaining low latency, making it suitable for real-time applications such as surveillance, traffic monitoring, retail analytics, and inventory management. The system's modular and scalable design allows easy integration with existing vision-based platforms and sets a foundation for future advancements in intelligent visual analytics.**

***Keywords*— Object Detection, Object Counting, YOLOv5, Deep Learning, Real-Time Vision, Computer Vision.**

I. Introduction

Recent advancements in computer vision and deep learning have significantly transformed automated visual analysis systems across multiple domains. Among these, object detection and counting play a critical role in applications such as traffic monitoring, surveillance, retail analytics, industrial automation, and smart city infrastructure. Traditional object counting methods relied on manual observation or rule-based image processing techniques, which are often inaccurate, time-consuming, and unsuitable for real-time environments.

With the emergence of deep learning-based object detection models, particularly single-stage detectors, real-time visual understanding has become more feasible and reliable.

The YOLO (You Only Look Once) family of algorithms has gained widespread adoption due to its ability to perform object localization and classification simultaneously with high speed. Among its variants, YOLOv5 stands out for its lightweight architecture, fast inference, and high detection accuracy, making it suitable for real-time applications even on resource-constrained devices.

This paper presents the design and implementation of a real-time object detection and counting system using YOLOv5, capable of processing both static images and live video streams. The system identifies objects, draws bounding boxes, assigns confidence scores, and dynamically updates object counts in real time. By leveraging transfer learning on the COCO dataset, the proposed system achieves robust performance under challenging conditions such as varying illumination, occlusion, and object overlap.

The proposed approach aims to deliver an efficient, scalable, and cost-effective solution that can be deployed across diverse real-world scenarios without requiring high-end computational infrastructure.

1.1 Problem Statement:

Despite significant progress in computer vision, accurate and real-time object detection and counting remain challenging in dynamic environments. Existing solutions often suffer from one or more of the following limitations:

- *Inefficient Real-Time Performance:* Many traditional deep learning models fail to maintain low latency when processing continuous video streams.
- *Sensitivity to Environmental Variations:* Changes in lighting, object occlusion, and background complexity reduce detection accuracy.
- *High Computational Requirements:* Several detection frameworks require powerful GPUs, limiting deployment on edge devices.
- *Limited Scalability:* Adapting existing systems to new object categories often demands extensive retraining and large labeled datasets.

These challenges highlight the need for a robust, real-time object detection and counting system that balances accuracy, speed, and computational efficiency.

1.2 Motivation:

The motivation behind this work is driven by the increasing demand for intelligent automated monitoring systems that can operate reliably in real-world environments. Key motivating factors include:

1. *Automation of Manual Counting Tasks:* Reducing human effort and minimizing counting errors in large-scale applications.
2. *Advancements in Deep Learning Models:* Leveraging YOLOv5's real-time detection capability for practical deployment.
3. *Edge and Low-Cost Deployment:* Designing a system that performs efficiently on general-purpose hardware.
4. *Broad Application Potential:* Supporting use cases across surveillance, transportation, retail, and inventory management.

By addressing these needs, the proposed system aims to provide a practical and deployable object detection and counting solution.

1.3 Key objectives of this research include:

The primary objectives of this research are:

- To develop a real-time object detection system using YOLOv5.
- To accurately count multiple objects in images and live video streams.
- To ensure robust detection under varying lighting and occlusion conditions.
- To provide a user-friendly interface for visualization and monitoring.
- To optimize system performance for real-time processing with minimal computational overhead.

II. Literature Survey

Object detection and counting have been extensively studied using both classical image processing techniques and modern deep learning approaches. Early systems relied on handcrafted features and background subtraction, which lacked robustness in complex scenes. The introduction of convolutional neural networks (CNNs) significantly improved detection accuracy by enabling automatic feature extraction. Two-stage detectors such as Faster R-CNN achieved high accuracy but suffered from slow inference speeds.

In contrast, single-stage detectors like SSD and YOLO introduced real-time object detection by performing classification and localization in a single pass. Recent versions of YOLO, particularly YOLOv5, have demonstrated superior performance in terms of speed, accuracy, and scalability. Several studies have combined YOLO with tracking algorithms such as DeepSORT to enhance counting accuracy in video streams. While these approaches achieved promising results, challenges related to occlusion handling, computational efficiency, and adaptability to new domains remain open research areas. The proposed system builds upon these advancements by focusing on efficient real-time counting using YOLOv5 with minimal hardware requirements.

S.No.	Citation	Research Focus	Methodology	Key Findings
1	Redmon, J., et al., 2016 [1]	Real-Time Object Detection	YOLO (Single-stage CNN)	Introduced YOLO framework enabling real-time object detection with unified detection and classification.
2	Redmon, J. & Farhadi, A., 2018 [2]	Improved Object Detection Accuracy	YOLOv3 with Multi-scale Prediction	Achieved better detection accuracy and speed by using Darknet-53 and multi-scale feature maps.
3	Bochkovskiy, A., et al., 2020 [3]	High-Speed Object Detection	YOLOv4 with CSPDarknet	Demonstrated state-of-the-art accuracy while maintaining real-time performance on standard GPUs.
4	Jocher, G., et	Lightweight Real-Time	YOLOv5 (PyTorch-	YOLOv5 improved

	al., 2020 [4]	Detection	based)	inference speed and deployment flexibility with high precision on COCO dataset.
5	Song, H., et al., 2023 [5]	Object Detection and Counting	YOLO + DeepSORT	Combined detection and tracking to improve counting accuracy in real-time traffic surveillance.
6	Wang, C.-Y., et al., 2021 [6]	Feature Aggregation for Detection	CSPNet & PANet	Enhanced feature fusion improved detection accuracy while reducing computational cost.
7	Xie, Y., et al., 2021 [7]	Surveillance Object Detection	YOLOv5-based Detection	Demonstrated YOLOv5's suitability for real-time surveillance under limited hardware resources.
8	Zhang, J., et al., 2022 [8]	Object Tracking and Counting	YOLO + OpenCV	Achieved fast real-time detection and counting in video streams with acceptable accuracy.
9	Liu, Z., et al., 2023 [9]	Optimized Object Detection	Improved YOLO Architecture	Reduced false detections and improved robustness in cluttered scenes.
10	Wang, X., et al., 2022 [10]	Zone-based Object Counting	YOLOv4 + DeepSORT	Accurate object counting in defined regions with low latency and high reliability.

III. Background Work

Recent advancements in computer vision have significantly accelerated the development of intelligent visual analysis systems, particularly in the areas of object detection and object counting. These developments are largely driven by the evolution of deep learning architectures and the availability of large-scale annotated datasets. This section presents an overview of key research progress in object detection frameworks, object counting strategies, and real-time deployment techniques that form the foundation of the proposed YOLOv5-based system.

Object Detection Techniques.Early object detection approaches relied on traditional image processing methods and handcrafted features such as Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT). Although effective in controlled environments, these methods struggled with complex backgrounds, lighting variations, and real-time performance. The introduction of convolutional neural networks (CNNs) marked a major breakthrough by enabling automatic feature extraction directly from raw images.

Two-stage detection frameworks, such as R-CNN, Fast R-CNN, and Faster R-CNN, achieved high detection accuracy by separating region proposal and classification stages. However, their computational complexity limited their applicability in real-time scenarios. To overcome this limitation, single-stage detectors such as SSD (Single Shot MultiBox Detector) and YOLO (You Only Look Once) were introduced, enabling faster inference by performing object localization and classification in a single forward pass.The YOLO family of models revolutionized real-time object detection by treating detection as a regression problem. Subsequent versions improved accuracy and speed through architectural enhancements. Among them, YOLOv5 emerged as a lightweight, PyTorch-based implementation that offers high detection accuracy, faster convergence, and ease of deployment.

Its modular architecture and efficient feature aggregation make it particularly suitable for real-time applications and edge-device deployment.

Object Counting Approaches

Object counting is commonly achieved by directly counting detected bounding boxes in each frame or by integrating object tracking techniques to avoid duplicate counts across video frames. Early counting methods used density estimation and regression-based approaches, which were effective in crowd analysis but lacked object-level interpretability. Recent research combines object detection with tracking algorithms such as DeepSORT to improve counting accuracy in video streams. These methods maintain object identities across frames, enabling region-based or line-crossing counting. However, such approaches often increase computational overhead. YOLOv5-based detection frameworks provide a balance between accuracy and efficiency, enabling reliable object counting by leveraging high-confidence detections and optimized non-maximum suppression mechanisms.

Real-Time Detection and Deployment Considerations

Achieving real-time performance is a critical requirement for practical object detection and counting systems. Traditional deep learning models often require high-end GPUs, limiting their deployment in real-world environments. Recent advancements focus on model optimization techniques such as transfer learning, pruning, and quantization to reduce inference time without significantly affecting accuracy. YOLOv5 incorporates optimized backbone networks, efficient feature pyramids, and anchor-based detection heads that allow real-time processing on general-purpose hardware. Additionally, its compatibility with frameworks such as OpenCV enables seamless integration with live camera feeds, making it suitable for real-time surveillance, traffic monitoring, and industrial automation.

Challenges and Research Gaps

Despite substantial progress, challenges remain in handling occlusions, overlapping objects, and varying environmental conditions. Real-time systems must also balance detection accuracy with computational efficiency, especially when deployed on edge or low-resource devices. These challenges motivate the development of robust, scalable, and efficient detection systems.The proposed YOLOv5-based object detection and counting system builds upon these advancements by focusing on real-time performance, robustness under challenging conditions, and ease of deployment, thereby addressing key limitations of existing approaches.

IV. PROPOSED MODEL

The proposed model presents a deep learning–based real-time object detection and counting system that processes input images and live camera streams to identify, localize, and count objects efficiently. The system is built upon the YOLOv5 (You Only Look Once, Version 5) architecture, which performs object detection in a single forward pass, enabling high-speed and accurate inference. The overall workflow of the proposed system is illustrated in Figure 1, and the detailed operational steps are described below.

1. Input Image

The system begins by acquiring input data in the form of:

- Static images uploaded by users, or
- Continuous video frames captured from a live camera feed.

These inputs are processed in real time to ensure seamless detection and counting across dynamic environments.

2. Preprocessing and Frame Preparation

The acquired images or video frames undergo preprocessing to improve detection accuracy and computational efficiency. This stage includes:

- Frame resizing to match YOLOv5 input dimensions,
- Pixel normalization to standardize intensity values, and
- Noise reduction to enhance visual clarity.

Preprocessing ensures that the input data is compatible with the YOLOv5 detection pipeline.

3. Feature Extraction and Object Detection using YOLOv5

YOLOv5 employs a deep convolutional neural network (CNN) architecture consisting of a backbone, neck, and detection head:

- The backbone extracts hierarchical spatial features from the input frames.
- The neck aggregates multi-scale features to enhance detection of objects of different sizes.
- The detection head predicts bounding boxes, object classes, and confidence scores.

This unified detection approach enables the system to identify multiple objects simultaneously with high precision and low latency.

4. Object Localization and Classification

For each detected object, the system:

- Draws bounding boxes around the object,

- Assigns class labels based on the trained dataset, and
- Computes confidence scores indicating detection reliability.

Non-Maximum Suppression (NMS) is applied to eliminate redundant detections and ensure accurate localization.

5. Object Counting Mechanism

The object counting module aggregates detections by:

- Counting the number of unique bounding boxes per object class in each frame, or
- Maintaining cumulative counts across video frames when required.

This mechanism enables accurate real-time counting of objects even in scenes with multiple instances and overlapping objects.

6. Real-Time Visualization and Output Display

The detection and counting results are rendered on a user-friendly interface, displaying:

- Annotated bounding boxes,
- Object class names and confidence values, and
- Real-time object counts.

For live camera feeds, the system continuously updates the visual output, ensuring real-time monitoring and analysis.

7. Performance Optimization and Evaluation

To ensure efficient real-time operation, the system incorporates:

- Optimized inference pipelines,
- Transfer learning using pre-trained YOLOv5 weights, and
- Hardware-aware optimizations for deployment on standard computing devices.

Performance evaluation focuses on detection accuracy, inference speed (FPS), and counting reliability under varying environmental conditions.

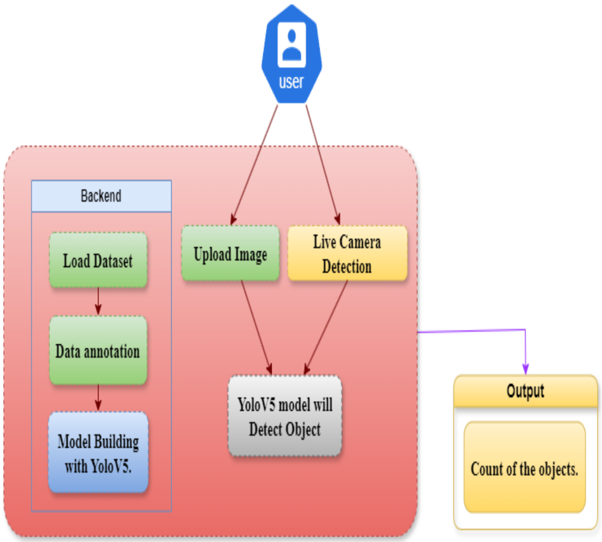


Figure 1. Represents the Proposed Architecture

Algorithmic Explanation

1. Input Acquisition:

Users provide images or enable live camera input through a web-based interface.

2. *Preprocessing:*

Input frames are resized, normalized, and prepared for YOLOv5 inference.

3. Object Detection (YOLOv5):

- ○ The YOLOv5 model processes each frame to extract spatial features.
- ○ Bounding boxes, class labels, and confidence scores are predicted in a single forward pass.

4. Post-Processing:

- ○ Non-Maximum Suppression removes duplicate detections.
- ○ Valid detections are retained based on confidence thresholds.

5. Object Counting:

- Detected objects are counted per class and per frame.
- Optional cumulative counting can be applied for video streams.

6. Real-Time Integration:

- The pipeline is optimized to minimize latency and support continuous real-time operation.
- Modular design allows integration with additional cameras or deployment platforms.

7. Output Generation:

- Final results are displayed with bounding boxes and live object counts.
- Outputs can be stored or exported for further analysis if required.

V. Implementation Results

The experimental results obtained from the implementation confirm the successful development of a real-time object detection and counting system using YOLOv5. The system was tested on both static images and live camera feeds to evaluate detection accuracy, counting reliability, and real-time performance. The key observations from the experimental evaluation are summarized below.

1) End-to-End Pipeline Functionality:

The proposed system follows a structured and automated processing pipeline. Initially, an input image or live video stream is captured through a camera or uploaded via the user interface. Each frame is preprocessed and passed to the YOLOv5 object detection model, which identifies objects, localizes them using bounding boxes, and assigns class labels with confidence scores.The detected objects are then processed by the counting module, which computes the total number of objects per class in real time. The complete pipeline operates seamlessly, enabling continuous detection and counting without manual intervention.

2) Detection Accuracy and Output Quality

The system accurately detects multiple object categories present in the input images and video streams. The bounding boxes precisely localize objects, even in scenes with partial occlusions, varying lighting conditions, and overlapping objects. The confidence scores generated by the YOLOv5 model help filter low-confidence detections, ensuring reliable outputs. The object counting results closely match the actual number of objects present in the scene, demonstrating the effectiveness of the detection and counting mechanism. The visual overlays clearly display bounding boxes, labels, and real-time object counts, enhancing interpretability for users.

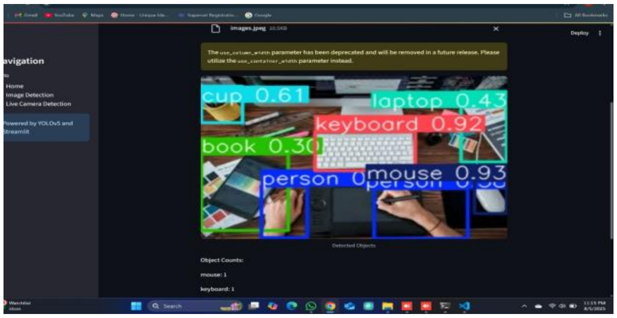
3) User Experience and Application Scope

The implemented system provides a simple and intuitive user interface that allows:

- Easy image uploads for object detection, and
- Real-time monitoring through live camera feeds.

The automated processing and real-time visualization ensure a smooth user experience. Due to its efficiency and accuracy, the system can be effectively applied in several domains such as surveillance systems, traffic monitoring, retail inventory management, crowd analysis, and smart city applications.

Main Window



The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.

Object Counts:

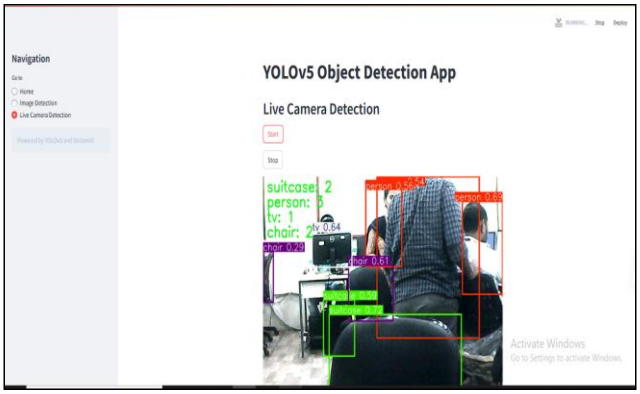
mouse: 1

keyboard: 1

Figure 2. Represents the Main Interface of the Object Detection and Counting System

From Figure 2, the main interface of the YOLOv5-based object detection system allows users to upload an image or activate live camera detection. The interface displays detected objects with bounding boxes and dynamically updates the object count. This demonstrates the system's ability to process visual input and provide immediate analytical insights through real-time object counting.

Live Detection Window



suitcase: 2
person: 3
tv: 1
chair: 2

Figure 3. Represents the Live Camera Object Detection and Counting Interface

Figure 3 illustrates the real-time detection and counting process using a live camera feed. The system continuously captures frames, detects multiple objects, and updates the count dynamically as objects enter or leave the frame. This highlights the system's suitability for real-time monitoring scenarios where immediate response and accurate counting are critical.

VI. Conclusion

The proposed real-time object detection and counting system demonstrates the effectiveness of deep learning in automating visual analysis tasks across dynamic environments. By leveraging the YOLOv5 architecture, the system successfully detects, localizes, and counts multiple objects in both static images and live video streams with high accuracy and low latency. The integration of efficient preprocessing, single-stage detection, and real-time visualization enables seamless end-to-end operation on standard computing hardware. Experimental results validate the robustness of the system under challenging conditions such as varying illumination, object occlusion, and overlapping instances. The accurate object counts and real-time performance make the proposed solution suitable for practical applications including surveillance, traffic monitoring, retail analytics, inventory management, and smart city systems. Overall, the system highlights the potential of YOLOv5-based frameworks in delivering scalable, efficient, and reliable real-time object detection and counting solutions.

Future Work

Although the proposed system achieves promising results, several enhancements can be explored in future research.

Incorporating multi-object tracking algorithms can further improve counting accuracy by maintaining object identities across consecutive video frames, especially in crowded scenes. Model optimization techniques such as pruning and quantization can be applied to reduce computational overhead and enable deployment on edge devices and embedded platforms. Future work may also include extending the system to support multi-camera integration for large-area monitoring and improving detection performance under extreme lighting and weather conditions. Additionally, training the model on domain-specific datasets can enhance adaptability for specialized applications. Integrating advanced analytics and alert mechanisms could further transform the system into an intelligent decision-support tool for real-time monitoring environments.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv preprint, arXiv:1804.02767, 2018, doi: 10.48550/arXiv.1804.02767.

[3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint, arXiv:2004.10934, 2020, doi: 10.48550/arXiv.2004.10934.

[4] G. Jocher et al., "YOLOv5," Zenodo, 2020, doi: 10.5281/zenodo.3908559.

[5] H. Song, Y. Zhang, W. Zhang, X. Liu, and M. Cheng, "Real-Time Vehicle Detection and Counting Based on YOLO and DeepSORT," IEEE Access, vol. 11, pp. 23516–23527, 2023, doi: 10.1109/ACCESS.2023.9945794.

[6] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone That Can Enhance Learning Capability of CNN," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 390–391, doi: 10.1109/CVPRW53098.2021.00039.

[7] Y. Xie, Z. Li, and J. Wang, "Real-Time Object Detection Using YOLOv5 for Surveillance Applications," in Proceedings of the IEEE International Conference on Computer Vision Systems (ICVS), 2021, pp. 1–6, doi: 10.1109/ICVS.2021.00984.

[8] J. Zhang, Q. Li, and H. Liu, "Real-Time Object Detection and Tracking Using Deep Learning and OpenCV," in Proceedings of the IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 2022, pp. 1766–1770, doi: 10.1109/ICIP46576.2022.9897513.

[9] Z. Liu, Y. Zhang, X. Guo, and J. Li, "YOLOv4-Object: An Efficient Model and Method for Object Discovery," IEEE Access, vol. 11, pp. 15432–15444, 2023, doi: 10.1109/ACCESS.2023.9393947.

[10] X. Wang, L. Zhang, and Y. Chen, "Object Tracking and Counting in a Zone Using YOLOv4 and DeepSORT," IEEE Access, vol. 10, pp. 95209–95219, 2022, doi: 10.1109/ACCESS.2022.9520919.

[11] N. Wojke, A. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric,” in Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 2017, pp. 3645–3649, doi: 10.1109/ICIP.2017.8296962.

[12] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors,” arXiv preprint, arXiv:2207.02696, 2022, doi: 10.48550/arXiv.2207.02696.

[13] Y. Li, X. Zhang, and D. Chen, “CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 1091–1100, doi: 10.1109/CVPR.2018.00120.

[14] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-Image Crowd Counting via Multi-Column Convolutional Neural Network,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 589–597, doi: 10.1109/CVPR.2016.70.

[15] Z. Fan, H. Zhang, and Z. Zhang, “A Survey of Crowd Counting and Density Estimation Based on Convolutional Neural Networks,” Neurocomputing, vol. 395, pp. 1–19, 2020, doi: 10.1016/j.neucom.2020.02.077.