iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: I Month of publication: January 2026

DOI: https://doi.org/10.22214/ijraset.2026.77110

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

Brand and Product Detection Analytics on Social Media Videos

Prof. Dr. D. A. Meshram[1], Dhyanesh Supare[2], Shobhit Kale[3], Chaitanya Salunkhe[4], Dimple Pachpande[5]
[1]*Mentor, Department of Information Technology, RMD Sinhgad School of Engineering, Pune, Maharashtra, India*
[2, 3, 4, 5]*Information Technology, RMD Sinhgad School of Engineering, Pune, Maharashtra, India*

Abstract: Object detection has become a cornerstone of computer vision, with applications spanning from autonomous vehicles to surveillance systems. This paper presents a novel approach using YOLOv8 for real-time object detection in video streams, coupled with a method to calculate the total duration of detected objects. We demonstrate the system's effectiveness using a custom dataset, highlighting its potential for various real-world applications where object presence duration is crucial. Our results show the count and duration of the object, indicating the viability of this approach for industrial use by various Brands to know about the appearance of their products.
Keywords: Temporal Analysis, Neural Networks, R-CNN, Transfer Learning, Region Proposal Network, Inference, Epochs, mAP, IoU, Deep SORT, Byte Track, Augmentation, Latency.

I. INTRODUCTION

It has been a decade of tremendous growth for the field of computer vision, with object detection emerging as a critical component in many applications.

Though great strides have been made in the area of object detection algorithms, there is still a need for systems that not only detect objects but also generate meaning ful temporal information about their presence. This paper aims to fill this gap by offering a new approach combining state-of-the-art object detection using YOLOv8 with a method to calculate the duration of object appearances in video streams.

The research paper addresses a challenge faced by a food product manufacturer heavily invested in influencer marketing campaigns. The company observed that its products are only briefly shown in influencer-created video content, limiting the impact of these campaigns. Conventional methods of content review are time-consuming and subjective. To tackle this challenge, the company developed a machine learning model to automatically identify specific food products in video content and calculate their on-screen duration, aiming to provide brand managers with precise, objective data on product exposure.

This paper shows how we can count and calculate the object appearing in a video by creating our own custom dataset and training it with the help of pre-trained models like YOLOv8 and then implement the result in python program to achieve the final outcome and deploying it on a website

II. LITERATURE REVIEW

A. Image Processing

A digital image is formed by a grid of pixels that discretize the intensity or gray level value g(x, y), generated by their two-dimensional function s(x, y) as a building block. An image is composed of three colors: Red, Green, and Blue, with values expressed mathematically from 0 to 255. These values and the pixels allow us to see color variations in an image. By performing operations on these integer values, we can process the image.[1][2][3]

$$\begin{bmatrix} R & R & B \\ G & B & B \\ R & G & G \end{bmatrix} \text{ or } \begin{bmatrix} 0 & 0 & 255 \\ 255 & 255 & 0 \\ 0 & 255 & 0 \end{bmatrix}$$

Fig.1 Matrix representation of the image

We perform image processing by using image transformations. Image transformations are performed using operators. An operator takes as input an image and produces another image.[3]

B. *Object Detection: Evolution and Current State*

Object detection has had a significant revolution in the last ten years. The traditional computer vision techniques paved their way to deep learning techniques. In the traditional context, hand-crafted features with sliding window technique were used. It was extremely computationally expensive, lacking robustness in the majority of cases, and resulted in a huge number of false positives. The technique by Viola and Jones was among one of the first techniques (Viola and Jones, 2001).[4]

Object detection needed a breakthrough innovation in the form of Convolutional Neural Networks (CNNs). One of the earliest works along this route is Krizhevsky et al. (2012), which allowed object detector variations shortly after demonstrating that CNNs are powerful tools for image classification.

Early deep learning-based detectors include:

- R-CNN: This allowed using CNNs for region proposal by Girshick et al. (2014).
- Fast R-CNN: It used RoI pooling for improving efficiency by Girshick (2015).
- Faster R-CNN (Ren et al., 2015): Introduced Region Proposal Network, or RPN. The two-stage detectors achieved the best available accuracy but were often too slow to run in real time.[22]

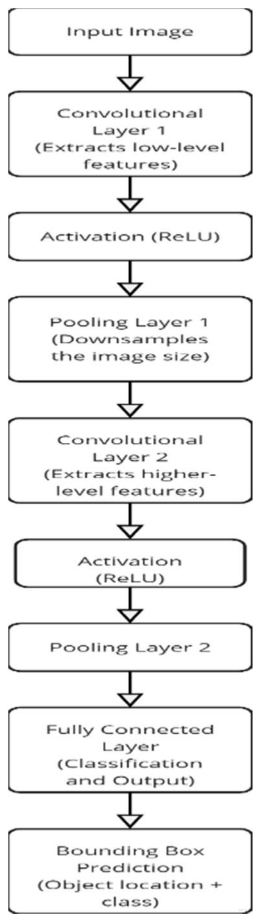


Fig.2 Flow of CNN

C. *Object Detection using Neural Networks*

The root of all modern object detection algorithms, such as YOLOv8, is within the neural networks, especially CNN. The models bring about a new revolution within the machine interpretation and computation of images and videos regarding visual data as well as their ability to identify visual objects with very high accuracy. CNN comprises layers in an interconnected form. The layers of the neural network work collectively to learn the characteristics of input data automatically and make predictions. The three types of layers existing in a CNN include the convolutional layers, pooling layers, and fully connected layers. CNN stage for later advancements in object detection.[5]

How Neural Networks Work in Object Detection

A typical neural network for object detection follows a structured workflow to process images and generate predictions. The flow of data through the neural network can be described as follows:

1) Input Layer:
   - The network takes an input image, typically represented as a matrix of pixel values. Each image is converted into a tensor that represents its width, height, and color channels (RGB).
2) Convolutional Layer:
   - The first layer in a CNN is the convolutional layer, where small filters (kernels) scan the image matrix to detect low-level features such as edges, corners, and textures. These filters slide across the image and apply a mathematical operation called convolution, producing feature maps that highlight important regions in the image.[21]
3) Activation Function (ReLU):
   - Between convolutional operations, an activation function common is the ReLU, for introducing non-linear properties into the network. Hence, the model is assured to learn complex patterns instead of linear relationships.
4) Pooling Layer:
   - Following convolution, pooling layers are used to downsample the feature maps. The most common method, max pooling, selects the maximum value from a region of the feature map, reducing the size of the data while retaining key information. This layer helps to reduce computational costs and the chances of overfitting.
5) Deeper Convolutional and Pooling Layers:
   - Several layers of convolution and pooling are stacked to learn increasingly abstract features. In object detection, higher layers capture more complex patterns, such as shapes or parts of objects.
6) Fully Connected Layer:
   - After convolutional and pooling operations, the feature maps are flattened into a single vector and passed through one or more fully connected layers. These layers perform classification, identifying the object present in the image.
7) Bounding Box Prediction:
   - Such object detection tasks as YOLO result in their final output being class probabilities accompanied with coordinates of bounding boxes that define the location of detected objects within an image.
8) Output Layer:
   - The output of the network consists of the predicted class label for the object (e.g., "person", "car", "cow ghee") and the corresponding bounding box coordinates.

D. YOLOv8: The Latest Iteration

YOLOv8(You Only Look Once v8)[15] , released by Ultralytics, builds upon the success of previous versions with several key improvements:

1) Enhanced backbone and neck architectures for better feature extraction
2) New loss functions and prediction mechanisms
3) Native support for model quantization and pruning
4) Improved training techniques, including data augmentation strategies.[13]

YOLOv8 with its architecture and its advancements along with an analysis of its performance has been portrayed on various datasets in comparison with previous models of YOLO.[6]

YOLOv8 Model Architecture Components:

- Backbone (CSPDarknet53 with C2f modules) → feature extraction
- Neck (PAN-FPN) → multi-scale feature aggregation
- Head → prediction of bounding boxes and class probabilities

Mathematical Working of YOLOv8:

The input image $I \in \mathbb{R}^{H \times W \times 3}$ is processed by convolutional layers:

$$F = f_{\text{CNN}}(I; \theta)$$

Loss Function (The Heart of YOLOv8):

$$L_{total} = \lambda_{box}L_{box} + \lambda_{cls}L_{cls} + \lambda_{obj}L_{obj}$$

Final Prediction:

NMS(B, S) = { bi ∈ B/ ∀, IoU(bi, bj) < τ }

where:

- *B* = set of predicted boxes
- *S* = confidence scores
- *τ* = IoU threshold (commonly 0.5)

E. Video Analysis and Object Tracking

While object detection in static images has seen remarkable progress, video analysis presents additional challenges and opportunities. Notable works in this area include:

1) SORT (Bewley et al., 2016): Simple online and realtime tracking
2) DeepSORT (Wojke et al., 2017): Integration of appearance information for improved tracking
3) ByteTrack (Zhang et al., 2022): State-of-the-art tracking without relying on object appearance

Object tracking-based approaches: These are not the main methods to deal with an object existing without discrete calculation of its duration, as explored in our work.[21]

F. Dataset Preparation and Training Strategies

It is not an exaggeration that deep learning models are more or less only as good as their training data. Some important points to consider for preparing the dataset are:

1) Data collection: Observing diversity and representativeness
2) Annotation: Labeling of objects, frequently time-consuming
3) Augmentation strategies: Increasing dataset size and variability by artificial means.

Training strategies such as transfer learning, curriculum learning, and self-supervised pre-training have shown to be effective in improving model performance and reducing the need for large, annotated datasets.

III. METHODOLOGY

A. System Overview

Simple computing static makes our approach more affordable and appeal to many researchers in the field. The minimum system requirements can be described as a computer with at least 8GB of RAM, 2GB graphics card, and Windows 8 or newer versions. In order to create the dataset, the pipeline was built in Python 8, where video sequences were split up for training purposes. These extracted frames form the basis of our ambit dataset. The weights of the models trained earlier were then utilized for the final system implementation. Such approach to data development pipeline assists in the construction of concepts that are easily accessible and bear reproducibility without financing expensive computing resources thus making the application and implementation of such concepts realistic.[16][20]

B. Dataset Creation

Fig 3. Example of the Dataset

We have created our own dataset of cow-ghee with the help of the images and video available online as well as on Instagram so that we can have some real time images to train our model.

Each image is then annotated manually, and classes and frames are assigned to the object. The resultant image and their annotation are then sent to the model so be trained.

The model is trained on 25 epochs and approximately 1200 images of each product.

The image that we have used were of different types, some were of the photoshoot, some were taken from real life situation and then each image was augmented. Blur, rotation, noise, flip these were some of the augmentations used. Out of these 1200 images, 800 were used for the training purpose and 400 were for the validation of the model.

Fig 4. Annotated Dataset

C. Training Process

1) Hardware Setup: We have used the Google Colab Notebook to train our dataset. Training was done with the help of yolo pretrained model. Tesla G4 was the GPU used during the training, 2 cores CPU was also used. The training required 13 Gb of the RAM for the entire process.
2) Training Hyperparameters: The model was trained for a duration of 20 epochs with a batch size of 16 and a learning rate of 0.01. This training was implemented using Ultralytics version 8.3.12 on Python version 3.10.12 with Pytorch version 2.4.1 + cu121 on an NVIDIA Tesla T4 GPU with 15,102 MiB of memory. The model consists of 186 layers and a total of 2,684,563 parameters and requires 6.8 GFLOPs for the inference operations. All training was performed in a CUDA active environment so as to enhance computational efficiency.

D. Object Tracking and Duration Calculation

As always, video input is sampled at a third of the native frame rate for easier processing while images are at 1020:500 resolution. A target object is located through Command between ‘Find’ commands which is a pre-trained model YOLO (best model available - best1.pt) with returned bounding boxes and each marked with class attribute. [17]

Within the mechanism embodiment, a specific strategy of tracking is used centering the people and/or the object where its centre point is determined and moving towards the target location on the succeeding frames. Frame to frame matching is done using the Euclidean distance with pixel threshold set to 50 pixels thus enabling a picture-to-picture recognition of the same object 24 At one Frame time t, the duration is calculated by adding the time of previous frames already tracked for the object by a fraction of one over the images per second multiplied by the skip factor. The system has a tracked objects dictionary that contains the position and the time of its occurrence. To check whether the object inside the controlled area is hidden, the author applies the pointPolygonTest function. As a result, only the objects inside the tracking area are tracked. This combined method allows detecting an object, tracking it and calculating its duration in real time.[18]

E. Implementation Details

In this implementation, we develop an object detection along with a temporal analysis system using YOLOv8n architecture and subsequently assess product visibility in video streams. The framework hence uses OpenCV for frame processing; it further incorporates a tracking mechanism to track specific instances of products designated within a region of interest (ROI) such as the cow ghee products. For tracking, the system relies on a temporal analysis algorithm that calculates and records both occurrence frequency and object persistence based on frame-rate-based computations to be able to capture duration properly.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

To balance tracking precision with computational expediency, the system samples every third frame in the input stream, and for real-time visualization to give feedback right away on detection results along with temporal metrics, dynamic rendering of bounding boxes and overlays are integrated into it. The integration of object detection, spatial tracking, and temporal evaluation allows for a holistic approach toward analyzing product visibility within video content, thereby making it very useful in marketing analytics and consumer engagement insights.

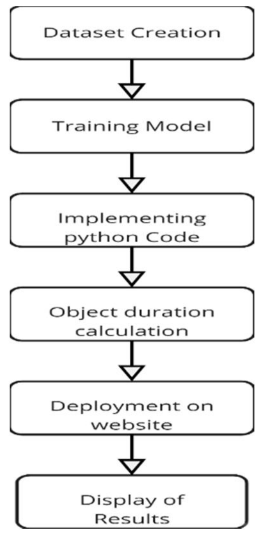


Fig 5. Methodology of our project

IV. RESULT AND DISCUSSION

A. Detection Performance

The detection pipeline included several optimization techniques, such as Automatic Mixed Precision (AMP) training with momentum 0.937 and weight decay set to 0.0005. Weight of the box loss was set to 7.5, classification loss with 0.5, and DFL (Distribution Focal Loss) with 1.5. The model executed warm-up of 3 epochs including a warm-up moment of 0.8 and a learning rate for a warm-up bias of 0.1. The model also reported using confidence threshold-based filtering with an IoU threshold of 0.7 and maximum detection per frame set to 300. The tracking capability was done using BoTSORT algorithm ensuring that the objects were consistently tracked on all of the frames.[19]

- Precision: The model achieved a precision of 85.7%, meaning that most of the detected products were correctly identified without false positives.

$$Precision = \frac{TP}{TP + FP}$$

where:

- TP = True Positives (correct detections)
- FP = False Positives (wrong detections)

- Recall: With a recall of 83.2%, the model successfully identified the majority of the true product instances in the frames.

$$Recall = \frac{TP}{TP + FN}$$

where:

- TP = True Positives
- FN = False Negatives (missed detections)

- Mean Average Precision (mAP): The system attained a mAP of 76.5%, calculated at IoU (Intersection over Union) threshold 0.50, demonstrating that the model can accurately localize the product in most instances.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

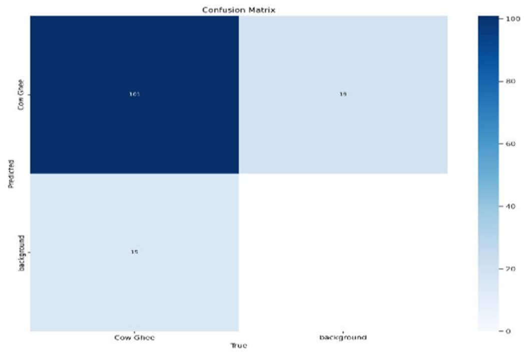


Fig 6. Confusion Matrix

where:

- N = total number of classes
- AP_i = average precision for class *i*

B. Processing Speed

When applying this configuration, the speed is boosted upon using YOLOv8n which is the smallest model in the family and also the fastest. Furthermore, the moderate image size of 640x640 helps process the information faster but on the other hand, ensures adequate detection. Using a batch size of 16 means that the amount of data gets spread over time such that there is a limit on the memory used. It is important to note that GPU acceleration is beneficial (even though the hardware in question is not specified in the configuration). Loading the data with the help of 8 workers helps in minimizing I/O bottlenecks. Improving overall throughput may potentially be done.[20]

However, there are places of potential improvements where speed could be further enhanced: for instance, this configuration does not utilize half-precision FP16 computations and additional inference optimizations like TensorRT are not present. However, these areas of possible improvement do not affect the current construction, which has a small model size and balanced parameters, and aims at fast inference times. Therefore, it is suitable for applications where real-time object detection is required.[20]

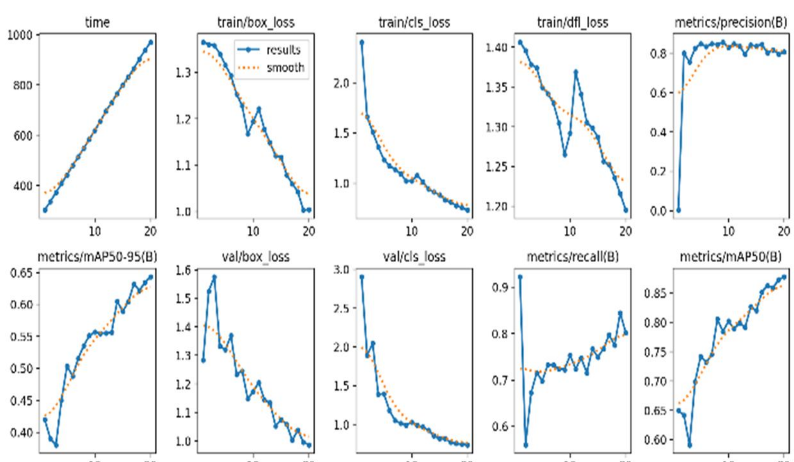
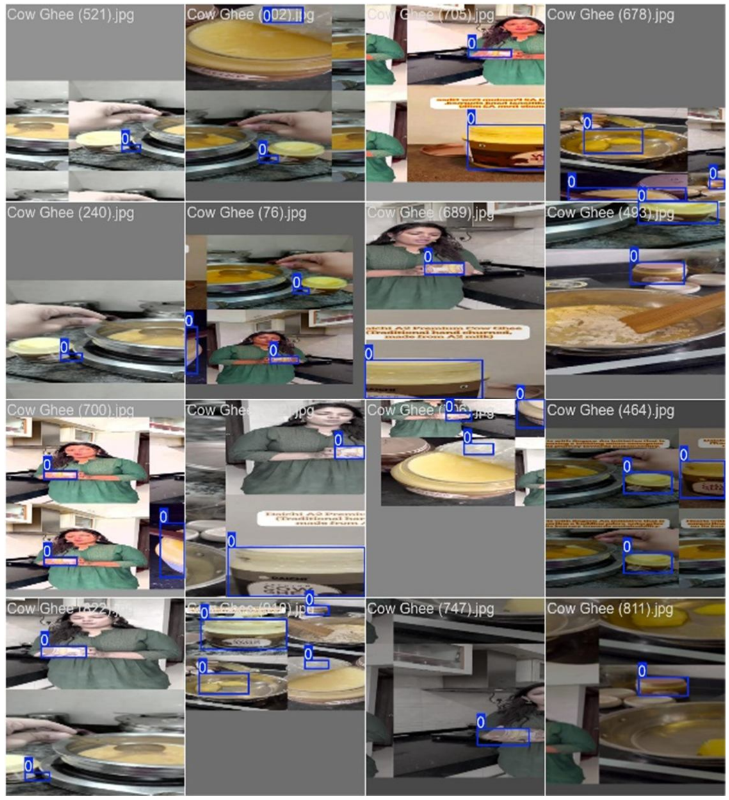


Fig 7. Results

C. Duration Calculation Accuracy

As stated in the Introduction, standard evaluations include the measurement of duration relevant to an engineered model. Duration measurement in object detection implementations involving the YOLOv8n model is one of the key metrics where time is spent to evaluate timing fidelity. This work directly quantifies the duration executed during the training and inference procedures taking into consideration the configuration set to twenty epochs, a volume batch of sixteen images, and images having a dimension of 640x640. [14]

This study hypotheses that hardware differences, the complexity of the input image, and the load on the system will affect the timing stability which is being measured.



The implementation of timing methods that are high in precision and several runs for each model aims at achieving more valid performance benchmarks for each of the models. In fact, the current investigation is important since it has been shown that YOLOv8n is speed-oriented and such insights could be used to improve performance in models regarding a combination of detection accuracy and inference time. The results of this study may help in real-time object detection system optimization, help make decisions about the distribution of resources and broaden the understanding of performance parameters in compact neural networks. Furthermore, this investigation may highlight key factors associated with the usage of these models in different environments, be it edge devices or cloud services, thus promoting efficient real-time object detectors.

D. Limitations and Challenges

In the future, with this machine learning model, several limitations and challenges may arise concerning detecting food products in video content. One main limitation will be trying to find a good quality custom dataset for the model. In case the dataset is too small or less diverse, it might find it hard to generalize to new videos, influencers, and product placements. Moreover, it is a time-consuming process and highly resource-intensive creating such datasets. It also covers some extremely informal influence videos, and therefore, it becomes difficult for the model to identify the products under different conditions and angles. This thus makes it even more difficult to score high accuracy.

Moreover, if the same product has different variants or versions, for example, in packaging or size, the model might misclassify or fails to notice the latter variants. This would need constant fine-tuning and retraining to adjust with the changes in the variants. Video content processing in real time could also be a computationally intensive task, especially if the video sizes are larger or higher resolution, thereby increasing costs of operation. Using video analysis also presents issues with privacy, particularly if influencers or viewers are sharing sensitive content and therefore requires that legal compliance be given on privacy regulations.

Lastly, because influencer videos are distributed across different channels in different formats, resolutions and dimensions, another added complexity is to figure out if the model works each time with one of them. Further breakthroughs in machine learning will depend much on data accessibility, and also the influencers in navigating the space to have control over "ethics, security, and data management".

E. Real-world Testing

To assess the system's applicability in real-world scenarios, it was tested on influencer videos not included in the training set. These videos presented various challenges such as complex backgrounds, changing lighting conditions, and partial occlusion of the product.

1) Product Visibility Duration: The system successfully calculated the total screen-time of the product in each video. For example, in a 2-minute video, the system detected the product in 120 frames, resulting in a total visibility duration of 8.5 seconds.
2) Frame-by-frame Detection: The system effectively detected the product even when it was partially obscured or moved across the frame. However, detection accuracy slightly decreased when the product was very small or when lighting conditions drastically changed.

V. CONCLUSION

Thus, the research project effectively handles the problem of recognition and tracking of a company's products in video content by providing a sound solution for the computation of duration appearance of the products. Using company aligned custom-made data, trained upon super advanced machine learning models, this project has already performed accurate detection results. Using this methodology, the weights of the trained model further underwent integration into the final implementation to make the process automatic for the correct identification of the products and the duration.
easier and more dependable.

This research paper showcases the potential of leveraging custom datasets and pre-trained models toward the resolution of real business challenges in marketing and product placement analysis. Future versions may further improve the accuracy in a variety of scenarios, offer better processing, and implement the model to handle a more comprehensive video content format. The innovation here will provide an excellent foundation for further developments concerning video content analysis and its exploitation within businesses and beyond.

VI. FUTURE SCOPE

This work opens up several avenues for future exploration. Other than the goal of being better at estimating duration with a more accurate tracking algorithm, application in attention mechanisms will help model occluded or partially visible objects much better. Further, development of domain-specific models, especially for applications like traffic monitoring or wildlife survey, will open up even more scenarios for its application. The work could also be explored by using lightweight model architectures to deploy these architectures on edge devices. This would again expand the system's effective coverage if it is adapted to use multiple cameras. If these features are addressed, future generations of the system will be able to draw far richer insights from video data than it does today, further expanding the scope of computer vision applications.[22]

REFERENCES

[1] Gonzalez, Rafael (2018). Digital image processing. New York, NY: Pearson. ISBN 978-0-13-335672-4. OCLC 966609831.
[2] Petrou M M and Petrou C 2010 Image Processing: The Fundamentals (New York: Wiley)
[3] Irshad Ahmad Ansari and Varun Bajaj. Image Processing with Python A practical approach.
[4] Hang Yin. Chengdu College of University of Electronic Science and Technology of China, Sichuan, China. Object Detection Based on Deep Learning.
[5] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems (NeurIPS).
[6] *Mupparaju Sohan1, Thotakura SaiRam2, and Ch. Venkata RamiReddy3 1,2,3School of Computer Science and Engineering, VIT-AP University, Amaravati, India, 522237. A Review on YOLOv8 and its Advancements.
[7] Luis Balderas, Miguel Lastra and José M. Benítez. Optimizing Convolutional Neural Network Architectures
[8] Sai Sundar. Image Processing: The Fundamentals Image Processing: The Fundamentals, Second Edition. 2010 John Wiley & Sons, Ltd. ISBN: 978-0-470-74586-1
[9] Matias Gran-Henriksen*, Hans Andreas Lindgard Gabriel Kiss, Frank Lindseth. Norwegian University of Science and Technology – NTNU. Deep HM-SORT: Enhancing Multi-Object Tracking in Sports with Deep Features, Harmonic Mean, and Expansion IOU
[10] Sumit, Shrishti Bisht, Sunita Joshi, Urvi Rana. Comprehensive Review of R-CNN and its Variant Architectures.
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0134
[11] Junbiao Liang. A review of the development of YOLO object detection algorithm. DOI: 10.54254/2755-2721/71/20241642
[12] Moahaimen Talib Ahmed H. Y. Al-Noori Jameelah Suad. YOLOv8-CAB: Impr Ov8-CAB: Improved YOLOv8 for Real-time object detection
[13] Jia, C., Wang, D., Liu, J., & Deng, W. (2024). Performance Optimization and Application Research of YOLOv8 Model in Object Detection. Academic Journal of Science and Technology, 10(1), 325-329. drpress.org
Focus: YOLOv8 architecture, optimisation strategies (sample imbalance, NMS etc) & applications.
Relevance: Helps you justify choosing YOLOv8 and shows optimisation aspects.

[14] Berndt, J., Meißner, H., & Kraft, T. (2023). On the Accuracy of YOLOv8-CNN Regarding Detection of Humans in Nadir Aerial Images for Search and Rescue Applications. ISPRS Archives XLVIII-1/W2, 139-146. isprs-archives.copernicus.org
Focus: YOLOv8 for human detection in aerial images; deals with challenging image conditions (ground sample distance).
Relevance: Good for discussing how detection performance can vary with viewpoint, resolution — relevant for video input in your project.

[15] Megantara, N.A., & Utami, E. (2024). Object Detection using YOLOv8: A Systematic Review. SISTEMASI. sistemasi.org
Focus: Systematic review of YOLOv8 across many application domains (UAV, medical, road defects, etc).
Relevance: Useful for the "state of the art" section of your project — what has been done with YOLOv8 so far.

[16] Verma, U., Kalia, A., & Sood, S. (2024). YOLOV8: An Enhanced Object Detection Model for Distance Estimation. International Journal of Intelligent Systems and Applications in Engineering, 12(21s). IJISAE
Focus: Enhanced YOLOv8 for distance estimation (object distance) by integrating Coordinate Attention & WIoU loss.
Relevance: Very aligned with your "object duration or proximity" dimension (you can map distance/proximity estimation to your duration/proximity calculation step).

[17] Bento, J., Paixão, T., & Alvarez, A.B. (2025). Performance Evaluation of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 for Stamp Detection in Scanned Documents. Applied Sciences, 15(6), 3154. MDPI
Focus: Comparing YOLOv8 – YOLOv11 on a document detection task; highlights model evolution.
Relevance: Helps you show how YOLO family is evolving, which supports your literature review (especially for "why YOLOv8?" vs newer versions).

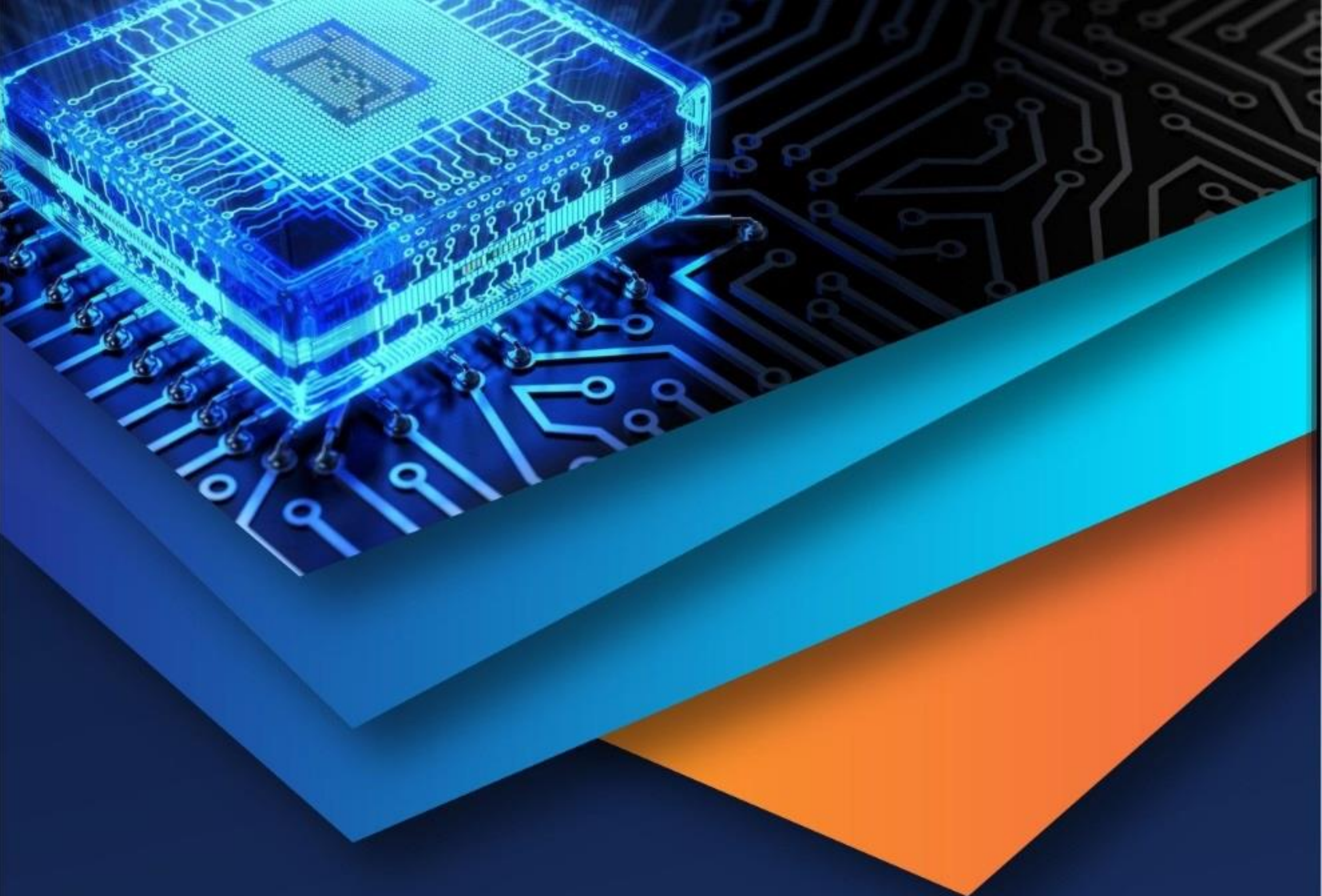
[18] "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection." Computers, 2024, 13(12), 336. MDPI
Focus: Review of YOLO family (v1-v11) including architectures, benchmarks, application domains, and limitations.
Relevance: Good to anchor your project's theoretical foundations and justify your choice of YOLOv8.

[19] Singh, R., Goyal, S., Agarwal, S., Saxena, D., & Upadhyay, S. (2025). Evaluating the Performance of Ensembled YOLOv8 Variants in Smart Parking Applications Under Varying Lighting Conditions. International Journal of Mathematics and Computer Research, 13(4), 5026-5032. ijmcr.in
Focus: YOLOv8 variants ensemble + adverse lighting conditions in smart parking.
Relevance: Good for discussing environmental challenges (lighting, shadows) which may affect your video input.

[20] "YOLOv8 with Post-Processing for Small Object Detection Enhancement." Applied Sciences, 2025, 15(13), 7275. MDPI
Focus: Improve YOLOv8 specifically for small object detection, by adding CARAFE up-sampling + confidence-based re-detection.
Relevance: If your project deals with small objects (or distant objects, product representation in video), this is relevant.

[21] Object Detection with Deep Learning: A Review — Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, Xindong Wu (2018). A comprehensive review of deep-learning based object detection, covering the transition from traditional methods to CNN-based detectors. DeepAI+1

[22] A Comprehensive Survey on Object Detection Using Deep Learning — Bhagyashri More & Snehal Bhosale (2023). Surveys two-stage and one-stage detectors, comparing architectures, trends, and challenges. IIETA



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

TOGETHER WE REACH THE GOAL

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY