iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VI **Month of publication:** June 2024

DOI: https://doi.org/10.22214/ijraset.2024.63521

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

YOLO Models for Security and Surveillance Applications

Manas Bajpai[1], Makam Kiran Kumar[2], Aaradhya[3]
Military College of Telecommunication Engineering, Ministry of Defence, Mhow, India

Abstract: ***Since 2015, the YOLO (You Only Look Once) series has evolved to YOLO-v8, prioritizing real-time processing and high accuracy for security and surveillance applications. Architectural enhancements in each iteration, culminating in YOLO-v9, cater to rapid detection, precision, and adaptability to resource-constrained edge devices. This study examines YOLO's evolution, emphasizing its relevance to security and surveillance contexts. Notable improvements in architecture, coupled with practical deployments for defect detection, underscore YOLO's alignment with stringent security and surveillance requirements.***
Index Terms: ***You Only Look Once (YOLO), Anchor box, Real-Time, FLOPS***

I. INTRODUCTION

The human visual cortex, pivotal for processing visual information, inspires artificial neural networks (ANNs) within deep learning, particularly in computer vision (CV). CV, including image classification, object detection, and object segmentation, relies on convolutional neural networks (CNNs). In manufacturing, computer vision automates quality inspection processes, enhancing efficiency and overcoming human-based limitations. Object detection, crucial in identifying defects, offers single-stage and two-stage detectors, with YOLO architectures combining classification and regression processes for real-time deployment. Despite YOLO's dominance, there's a lack of comprehensive reviews on evolving variants and their industrial applicability. This paper aims to fill this gap by examining YOLO variants and their impact on accuracy, speed, and computational efficacy in industrial settings. The paper explains how the versatility of YOLO algorithms has made them quite useful and promising even for a large number of military applications. It also discusses literature on YOLO-based defect detection applications in manufacturing, production, and Research and Development industry transforming the modes of operations by optimization for ergonomic outputs.

II. YOLO: GENESIS AND ITS TRANSFORMATION

A. YOLOv1

The YOLO-v1 algorithm, introduced in 2015 by Joseph Redmon et al., marked a paradigm shift in object detection by formulating it as a single-pass regression problem. This novel approach unified the prediction of bounding boxes and class probabilities, fundamentally altering the landscape of computer vision. The algorithm leverages a grid overlay onto the input image, partitioning it into grid cells. Each grid cell is tasked with predicting multiple bounding boxes and their associated confidence scores, encapsulating the probability of object presence within each box.

Analyzing the mathematical underpinnings of YOLO-v1: Consider an input image partitioned into an S × S grid, where each grid cell is responsible for predicting B bounding boxes. For each bounding box, the algorithm predicts five parameters: (x, y, w, h), representing the center coordinates, width, and height of the bounding box, respectively, along with a confidence score (Pr(object)), indicating the likelihood of an object being present within the box.

The confidence score (Pr(object)) is computed as follows:

Pr(object) = IoU_truth_pred × Pr(object)_predicted

where IoU_truth_pred represents the intersection over union (IoU) between the predicted bounding box and the ground truth bounding box.

To accommodate for object absence and mitigate overlapping predictions, YOLO-v1 employs non-maximum suppression (NMS). NMS involves discarding predicted bounding boxes with an IoU lower than a specified threshold.

The loss function of YOLO-v1 accounts for prediction errors in bounding box coordinates and confidence scores. It comprises two components: the localization loss (ℓ_loc) and the confidence loss (ℓ_conf). The localization loss penalizes errors in predicting bounding box coordinates, while the confidence loss penalizes errors in predicting confidence scores.

©IJRASET: All Rights are Reserved

The localization loss (ℓ_loc) is calculated as:
ℓ_loc = λ_coord Σ_(i=0)^(S^2) Σ_(j=0)^B 𝟙_obj^ij ((x_i - x̂_i)^2 + (y_i - ŷ_i)^2)
where λ_coord is a regularization parameter, (x_i, y_i) are the predicted center coordinates of the bounding box, (x̂_i, ŷ_i) are the ground truth center coordinates, and 𝟙_obj^ij is an indicator function that equals 1 if object j is assigned to grid cell i, otherwise 0.
The confidence loss (ℓ_conf) is given by:
ℓ_conf = Σ_(i=0)^(S^2) Σ_(j=0)^B 𝟙_obj^ij (IoU_truth_pred - Pr(object)_predicted)^2
where IoU_truth_pred is the IoU between the predicted and ground truth bounding boxes, and Pr(object)_predicted is the predicted confidence score.
In summary, YOLO-v1's innovative approach, mathematical rigor, and efficient implementation have propelled it to the forefront of object detection, laying the groundwork for subsequent iterations and advancements in computer vision.

B. *Advanced Versions of YOLO Algorithms*

The evolution of YOLO from YOLOv2 to YOLOv6 has witnessed significant advancements in object detection within computer vision. YOLOv2 introduced anchor boxes to improve bounding box prediction accuracy, enabling better localization of objects. Additionally, it incorporated batch normalization and high-resolution classifiers, enhancing detection performance. YOLOv3 further refined the architecture by integrating feature pyramid networks (FPN) and adopting multi-scale training, improving detection accuracy across different object sizes and scales. YOLOv4 introduced groundbreaking features such as CSPDarknet53 and PANet, significantly boosting detection performance and robustness. YOLOv5, built on PyTorch, streamlined the architecture and introduced efficient training strategies, achieving impressive speed and accuracy in real-time object detection. Finally, YOLOv6 continues this trajectory of innovation with further optimizations and advancements, solidifying YOLO's position as a leading framework in object detection.

III. PIVOTAL FEATURES OF SUCCESSIVE YOLO ALGORITHMS

A. *Key Features of YOLOv5*

1) Transition to PyTorch: YOLO-v5 departed from the Darknet framework, becoming the first YOLO variant natively implemented in PyTorch. This shift aimed to enhance accessibility and configurability, catering to a wider audience within the computer vision community.
2) *Automated Anchor Box Learning:* YOLO-v5 introduced automated anchor box learning, a novel approach where the network dynamically learns the best-fit anchor boxes during training. This integration streamlines the model's adaptability to diverse datasets, improving overall performance and accuracy.
3) *Variant Configurations:* YOLO-v5 offered variant configurations with different computational parameters, empowering users to customize the model according to specific requirements. This flexibility enhances usability and facilitates seamless integration into various computer vision applications.

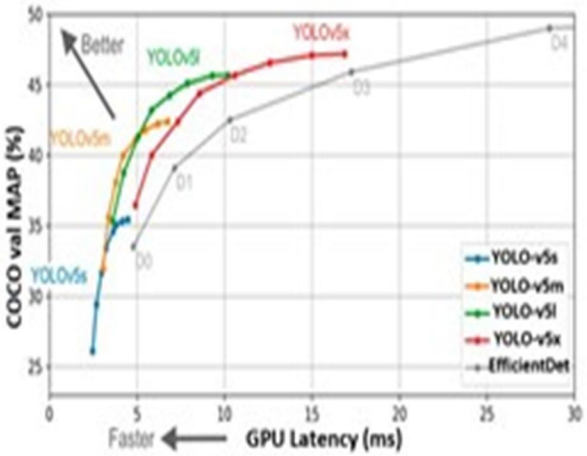
B. *Key Features of YOLOv6*

1) *Anchor-Free Approach:* YOLO-v6 diverges from its predecessors by adopting an anchor-free approach. This strategic shift results in a remarkable performance boost.
2) *Revised Reparametrized Backbone and Neck:* YOLO-v6 introduces the EfficientRep backbone and Rep-PAN neck, departing from conventional designs where regression and classification heads shared identical features.
3) *Two-Loss Function:* YOLO-v6 employs a two-loss function strategy. It utilizes Varifocal loss (VFL) as the classification loss, assigning varying degrees of importance to positive and negative samples. Additionally, Distribution focal loss (DFL) is employed alongside SIoU/GIoU as the regression loss, particularly effective in scenarios with blurred ground truth box boundaries.
4) *Knowledge Distillation:* YOLO-v6 integrates knowledge distillation, employing a teacher model to train a student model. The predictions of the teacher model serve as soft labels alongside ground truth for training the student. This approach enables training a smaller (student) model to replicate the high performance of a larger (teacher) model without significantly increasing computational costs.
5) *Performance Comparison with Predecessors:* Figure 10 demonstrates the superior performance of YOLO-v6 compared to its predecessors, including YOLO-v5, on the benchmark COCO dataset. YOLO-v6 achieves higher mean average precision (mAP) at various frames per second (FPS), highlighting its efficacy and suitability for diverse industrial applications.

C. YOLO-v7: Architectural Reforms for Enhanced Accuracy and Speed

1) *E-ELAN Backbone:* Implements the Extended Efficient Layer Aggregation Network (E-ELAN) to optimize accuracy and speed by considering factors such as memory access cost and gradient path.
2) *Compound Model Scaling:* Allows coherent scaling of width and depth in concatenation-based networks, accommodating different size priorities.
3) *Average Weighted Model:* Averages a set of model weights to create a more robust network, enabling module-level re-parameterization for adaptive parameterization strategies.
4) *Auxiliary Head Coarse-to-Fine Concept:* Introduces an auxiliary head to assist in training and provide additional supervision signals for better convergence.
5) *Trainable Model Scaling:* Introduces dynamic adjustments in model size during training, enhancing versatility and resource utilization.

D. YOLO-v8: Advancements for Increased Hardware-Efficiency and Inference Speed



Variant	mAP 0.5:0.95 (COCO-val)	FPS Tesla T4	Parameters (Million)
YOLO-v6-N	35.9 (300 epochs)	802	4.3
YOLO-v6-T	40.3 (300 epochs)	449	15.0
YOLO-v6-RepOpt	43.3 (300 epochs)	596	17.2
YOLO-v6-S	43.5 (300 epochs)	495	17.2
YOLO-v6-M	49.7	233	34.3
YOLO-v6-L-ReLU	51.7	149	58.5

1) *Hardware-Efficient Architectural Reforms:* Showcases improved throughput compared to YOLO-v5 and YOLO-v6, optimizing resource utilization and computational efficiency.
2) *Inference Speed and Parameter Efficiency:* Outperforms YOLO-v5 and YOLO-v6 in terms of throughput, emphasizing high-inference speed, especially beneficial for constrained edge device deployment.
3) *Constrained Edge Device Deployment:* Optimized for various edge GPU setups, ensuring optimal performance in diverse deployment scenarios.

E. YOLOv9 Establishes a New Benchmark for Efficiency

By integrating the architectural advancements of GELAN with the training enhancements from PGI, YOLOv9 achieves unparalleled efficiency and performance:

1) In comparison to previous iterations of YOLO, YOLOv9 achieves superior accuracy with 10-15% fewer parameters and 25% fewer computations. This results in significant enhancements in speed and capability across different model sizes. YOLOv9 outperforms other real-time detectors such as YOLO-MS and RT-DETR in terms of parameter efficiency and FLOPs. It demands significantly fewer resources to achieve a specified level of performance. Even smaller variants of YOLOv9 surpass larger pre-trained models like RT-DETR-X. Despite utilizing 36% fewer parameters, YOLOv9-E achieves a superior 55.6% AP through more streamlined architectures. By addressing efficiency at both the architecture and training levels, YOLOv9 establishes a new standard for maximizing performance within resource constraints.
2) GELAN - Optimized Architecture for Efficiency: YOLOv9 introduces the General Efficient Layer Aggregation Network (GELAN), a novel architecture designed to maximize accuracy while operating within a limited parameter budget. Building upon the foundation of previous YOLO models, GELAN optimizes various components specifically for efficiency.

©IJRASET: All Rights are Reserved

IV. YOLO FOR SECURITY AND SURVEILLANCE: A VISION FOR AI-POWERED NATIONAL SECURITY

Utilizing the YOLO family of object detectors has presented a significant advancement in computer vision, particularly in domains like national security. From its inception with YOLO-v1 in 2015 to the latest YOLO-v8, each iteration has introduced architectural innovations, driving its popularity and applicability in various fields. YOLO algorithms have enabled the development of solutions for object detection in both videos and static images, particularly on edge devices due to their compactness and lower computational requirements. The rising popularity of YOLO can be attributed to two key factors. Firstly, its architectural design, optimized for one-stage detection and classification, ensures computational efficiency, vital for real-time applications. Secondly, the transition from Darknet to the PyTorch framework, notably with YOLO-v5, has expanded its user base, making the architecture more accessible to a wider audience.

In the realm of national security applications within the Indian Military, the efficiency and accessibility of YOLO algorithms are crucial. The adoption of the PyTorch framework has streamlined integration and adaptation, enabling swift development and deployment tailored to specific defense-related use cases. The capabilities of the YOLO family extend beyond traditional computer vision applications, finding relevance in industrial defect detection. YOLO variants, such as YOLO-v5, offer real-time compliance and flexibility, aligning well with the stringent requirements of surface defect detection in manufacturing. For the Indian Military, leveraging YOLO algorithms presents opportunities in various scenarios. The ability to modify internal modules to meet specific needs without sacrificing real-time compliance resonates with the dynamic and diverse demands of military applications.

V. HOMELAND SECURITY APPLICATIONS OF YOLO

A. *YOLO Application in Counter Insurgency/Disturbed Area Scenarios*

1) *Vehicle Tracking:* YOLO's robustness to occlusion and varying lighting conditions enhances its effectiveness in vehicle tracking applications. Even when vehicles are partially obscured or subjected to changing illumination, YOLO can reliably detect and track them, ensuring continuity and accuracy in the tracking process. Additionally, YOLO's real-time performance is particularly advantageous for live monitoring and surveillance applications. Its ability to process video streams at high speeds enables rapid detection and tracking of vehicles, facilitating timely response and intervention in scenarios such as traffic management, law enforcement, and border security. Moreover, YOLO's versatility allows for seamless integration with existing surveillance systems and hardware, making it adaptable to different deployment scenarios and operational requirements. Whether deployed on fixed surveillance cameras, drones, or mobile vehicles, YOLO can provide real-time vehicle tracking capabilities with minimal latency.

2) *Facial Recognition under Occlusion:* In counterinsurgency scenarios, facial recognition plays a critical role in identifying and tracking individuals involved in illicit activities. However, conventional facial recognition systems often struggle with occlusion, where facial features are partially or completely obstructed by objects such as masks, helmets, or scarves commonly worn by insurgents to conceal their identity. Leveraging the YOLO (You Only Look Once) object detection framework, specifically adapted for facial recognition under occlusion, presents a promising solution. YOLO's single-stage architecture, optimized for real-time object detection, offers several advantages in this context. By integrating deep learning techniques, YOLO can effectively detect and classify faces even when partially obscured, thanks to its ability to learn complex patterns and features. Additionally, YOLO's versatility allows for seamless integration with existing surveillance systems and hardware, facilitating rapid deployment in the field. To address occlusion challenges, YOLO can be enhanced with advanced feature extraction methods and robust facial recognition algorithms. Techniques such as multi-scale feature fusion and attention mechanisms can improve the model's ability to extract discriminative features from partially visible faces, enhancing recognition accuracy under challenging conditions. Furthermore, YOLO's adaptability enables dynamic adjustments to model parameters and architectures, allowing for fine-tuning based on specific counterinsurgency scenarios and environmental factors. This flexibility ensures optimal performance in varying conditions, crucial for reliable facial recognition in dynamic operational environments.

3) *Satellite/Drone-Based Surveillance:* In the domain of satellite and drone-based object change detection, the YOLO (You Only Look Once) object detection framework presents a robust solution for identifying alterations in the environment over time. By analyzing high-resolution imagery captured by satellites or drones, YOLO can swiftly identify and track objects of interest with accuracy and efficiency. A significant advantage of utilizing YOLO for object change detection in satellite and drone imagery lies in its capacity to swiftly process large datasets. YOLO's architecture facilitates rapid detection and categorization of objects within images, enabling timely recognition of environmental changes. Furthermore, YOLO's versatility and adaptability make it suitable for detecting various objects, including buildings, vegetation, infrastructure, and vehicles. Through training on

diverse datasets representing different objects and environmental contexts, YOLO can proficiently detect and monitor changes across diverse landscapes. Moreover, YOLO's resilience to occlusion and varying lighting conditions enhances its performance in satellite and drone imagery, where objects may be partially obstructed or subjected to fluctuating light conditions. This ensures reliable detection and monitoring of changes, even in challenging scenarios. Its real-time processing capabilities enable swift analysis of imagery, facilitating prompt detection and response to changes as they occur. This feature is particularly beneficial for applications such as disaster monitoring, urban planning, and environmental conservation, where timely detection of changes is essential for decision-making and response efforts.

VI. CONCLUSION

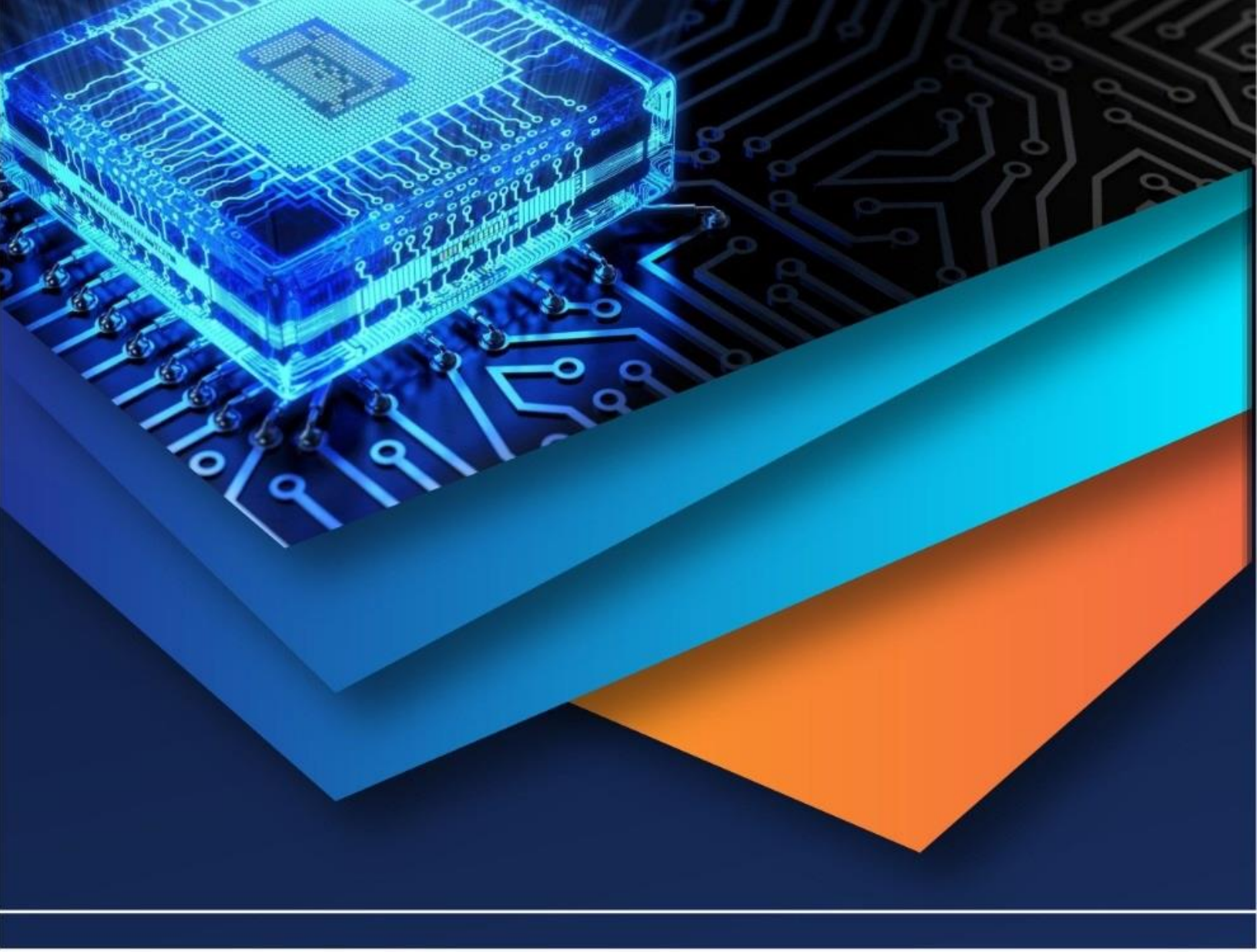
In summary, this review not only chronicles the evolutionary journey of YOLO variants but also underscores their potential ramifications for national security, particularly within the purview of the Indian Military. The emphasis on YOLO's suitability for constrained edge deployment, its lightweight design, and real-time capabilities positions it as a valuable asset for augmenting armed forces equipment across various applications. In the homeland security context, these advancements hold the promise of bolstering deployment capabilities, conserving energy resources, and achieving high rates of inference—essential attributes for military equipment operating in diverse and demanding environments. As research institutions continue to invest in refining YOLO architectures with a focus on edge-friendly deployment, the Indian Military stands to gain from technological progress that aligns with its national security imperatives. The prospective integration of YOLO into armed forces equipment, coupled with adaptation to various hardware platforms and Internet of Things (IoT) devices, offers the prospect of enhancing existing processes with minimal resource expenditure. Looking ahead, the Indian Security Forces can explore opportunities to integrate YOLO algorithms into its equipment, fostering innovation and fortifying defense capabilities. The accessibility, versatility, and performance of YOLO render it an enticing choice for applications where computer vision can significantly contribute to safeguarding national security, encompassing domains such as surveillance, reconnaissance, target tracking, and threat detection.

REFERENCES

[1] Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao arXiv:2402.13616v2 [cs.CV] https://doi.org/10.48550/arXiv.2402.13616 arXiv:2402.13616v2 [cs.CV] 29 Feb 2024
[2] Pollen, D.A. Explicit neural representations, recursive neural networks and conscious visual perception. Cereb. Cortex 2003, 13, 807–814.
[3] Using artificial neural networks to understand the human brain. Res. Featur. 2022.
[4] Improvement of Neural Networks Artificial Output. Int. J. Sci. Res. (IJSR) 2017, 6, 352–361.
[5] Dodia, S.; Annappa, B.; Mahesh, P.A. Recent advancements in deep learning based lung cancer detection: A systematic review. Eng. Appl. Artif. Intell. 2022, 116, 105490.
[6] Ojo, M.O.; Zahid, A. Deep Learning in Controlled Environment Agriculture: A Review of Recent Advancements, Challenges and Prospects. Sensors 2022, 22, 7965.
[7] Jarvis, R.A. A Perspective on Range Finding Techniques for Computer Vision. IEEE Trans. Pattern Anal. Mach. Intell. 1983, PAMI-5, 122–139.
[8] Hussain, M.; Bird, J.; Faria, D.R. A Study on CNN Transfer Learning for Image Classification. 11 August 2018.
[9] Yang, R.; Yu, Y. Artificial Convolutional Neural Network in Object Detection and Semantic Segmentation for Medical Imaging Analysis. Front. Oncol. 2021, 11, 638182.
[10] Haupt, J.; Nowak, R. Compressive Sampling vs. Conventional Imaging. In Proceedings of the 2006 International Conference on Image Processing, Las Vegas, NV, USA, 26–29 June 2006; pp. 1269–1272.
[11] Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. Pattern Recognit. 2018, 77, 354–377.
[12] Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. Pattern Recognit. 2018, 77, 354–377.
[13] Perez, H.; Tah, J.H.M.; Mosavi, A. Deep Learning for Detecting Building Defects Using Convolutional Neural Networks. Sensors 2019, 19, 3556.
[14] Hussain, M.; Al-Aqrabi, H.; Hill, R. PV-CrackNet Architecture for Filter Induced Augmentation and Micro-Cracks Detection within a Photovoltaic Manufacturing Facility. Energies 2022, 15, 8667.
[15] Hussain, M.; Dhimish, M.; Holmes, V.; Mather, P. Deployment of AI-based RBF network for photovoltaics fault detection procedure. AIMS Electron. Electr. Eng. 2020, 4, 1–18.
[16] Hussain, M.; Al-Aqrabi, H.; Munawar, M.; Hill, R.; Parkinson, S. Exudate Regeneration for Automated Exudate Detection in Retinal Fundus Images. IEEE Access 2022.
[17] Hussain, M.; Al-Aqrabi, H.; Hill, R. Statistical Analysis and Development of an Ensemble-Based Machine Learning Model for Photovoltaic Fault Detection. Energies 2022, 15, 5492.
[18] Singh, S.A.; Desai, K.A. Automated surface defect detection framework using machine vision and convolutional neural networks. J. Intell. Manuf. 2022, 34, 1995–2011.
[19] Weichert, D.; Link, P.; Stoll, A.; Ruping, S.; Ihlenfeldt, S.; Wrobel, S. A review of machine learning for the optimization of production processes. Int. J. Adv. Manuf. Technol. 2019, 104, 1889–1902.

[20] Wang, J.; Ma, Y.; Zhang, L.; Gao, R.X.; Wu, D. Deep learning for smart manufacturing: Methods and applications. J. Manuf. Syst. 2018, 48, 144–156.
[21] Weimer, D.; Scholz-Reiter, B.; Shpitalni, M. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Ann. 2016, 65, 417–420.
[22] Kusiak, A. Smart manufacturing. Int. J. Prod. Res. 2017, 56, 508–517.
[23] Soviany, P.; Ionescu, R.T. Optimizing the Trade-Off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction. In Proceedings of the 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Romania, 20–23 September 2018.
[24] Du, L.; Zhang, R.; Wang, X. Overview of two-stage object detection algorithms. J. Phys. Conf. Ser. 2020, 1544, 012033.
[25] Sultana, F.; Sufian, A.; Dutta, P. A Review of Object Detection Models Based on Convolutional Neural Network. In Advances in Intelligent Systems and Computing; Springer: Singapore, 2020; pp. 1–16.
[26] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
[27] Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. arXiv 2017, arXiv:1701.06659.
[28] Cheng, X.; Yu, J. RetinaNet with Difference Channel Attention and Adaptively Spatial Feature Fusion for Steel Surface Defect Detection. IEEE Trans. Instrum. Meas. 2020, 70, 2503911.
[29] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
[30] Wang, Z.J.; Turko, R.; Shaikh, O.; Park, H.; Das, N.; Hohman, F.; Kahng, M.; Chau, D.H.P. CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization. IEEE Trans. Vis. Comput. Graph. 2020, 27, 1396–1406.
[31] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Commun. ACM 2017, 60, 84–90.
[32] Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 2014, arXiv:1409.1556.
[33] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 12 June 2015.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)