International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

An Improved Lightweight Parameters Network for Strawberry Flowers Detection

K. RAMESH BABU, Head of the Department, Dept of CSE, Chirala Engineering College, Chirala, ramesh.cs04@gmail.com

SHAIKJOHN ISMAIL, PG Student - MCA, Dept of MCA, Chirala Engineering College, Chirala, johnismail916@gmail.com

Abstract: Accurate and efficient detection of target crops is paramount for the advancement of intelligent agriculture, enabling precision farming and resource optimization. While numerous studies have focused on improving detection algorithms, the challenge lies in implementing them on embedded devices due to escalating computing demands. This project addresses this challenge by introducing a lightweight parameter network tailored for crop detection. Leveraging grouped convolution and integrating convolutional layers with Batch Normalization, the proposed network achieves accelerated inference without compromising accuracy. Experimental validation on diverse datasets, including Strawberry Flower, Tomato, Wind Turbine, and PASCAL, demonstrates the efficacy of the network through comparative analysis with baseline models. In particular, experiments on remote aerial satellite images utilizing the Super YOLO model showcase significant performance gains. The Conv-BP-YOLOs model achieves a remarkable 98% mean Average Precision (mAP), while further enhancements are achieved through exploration of advanced techniques such as YOLO V5x6 and YOLOV8, surpassing 99% mAP. Additionally, model implementation on Colab utilizing Faster R-CNN, SSD, EfficientDet RetinaNet, and various YOLO versions underscores the versatility and applicability of the proposed approach. This research contributes to the development of efficient crop detection solutions crucial for the evolution of intelligent agriculture.

Index Terms: Lightweight, grouped convolution, real-time detection, embedded platforms.

1. INTRODUCTION

In today's technologically driven world, computer vision technology has become ubiquitous, permeating various facets of daily life. However, despite significant advancements, achieving human-like speed and accuracy in decoding image information remains a formidable challenge [1]. At the forefront of computer vision tasks lies object detection, a critical and complex endeavor aimed at classifying and localizing objects within images or videos [2], [3]. The importance of fast and accurate object detection cannot be overstated, as it underpins numerous downstream applications, including the utilization of robots for tasks such as pollinating strawberry flowers [4], [5].

The rapid and precise detection of strawberry flowers is essential for tasks such as yield estimation and the development of pollination robots, highlighting the critical role of object detection in advancing agricultural practices [4], [5]. Over the years, the field of object detection has witnessed a remarkable evolution, transitioning from manual feature-based methods, such as the Viola-Jones Face Detector (VJ Det), to deep learning-based approaches like the You Only Look Once (YOLO) series [6]. Research institutions and universities continually strive to enhance detection algorithms, proposing new models with improved accuracy [6].

However, a significant hurdle in the widespread adoption of object detection algorithms lies in the substantial computing power demands they entail. Both traditional algorithms and deep learning-based approaches require dedicated large computing devices, posing challenges for deployment on resource-constrained platforms such as Unmanned Aerial Vehicles (UAVs) or mobile robots [7]. The computing capabilities of mobile devices are often insufficient to support high-precision detection algorithms, leading to compromises in performance and significantly reduced device lifetimes due to overload computing [7]. Thus, there is an urgent need for algorithms that not only deliver high accuracy but also exhibit low arithmetic power requirements, particularly tailored for deployment on mobile devices.

While substantial progress has been made in addressing the computing power demands of object detection algorithms, significant challenges persist. Some works focus solely on improving detection accuracy without considering the algorithm's computing power requirements, rendering them unsuitable for deployment on embedded devices [8]. Additionally, overlooking the significant parameters of algorithms, which contribute to the high computing power demand, poses another challenge. Some approaches only address post-training methods like pruning or quantization to reduce weight, potentially compromising detection accuracy [8].

Despite these challenges, the field of object detection continues to advance, with researchers exploring innovative solutions to address the complex interplay between accuracy and computational efficiency. Traditional object detection methods, from VJ Det to the Deformable Parts Model (DPM), have witnessed notable improvements in detection speed and accuracy [9]. The advent of Convolutional Neural Networks (CNNs) has further revolutionized object detection, with models like R-CNN and YOLO demonstrating substantial leaps in detection performance compared to traditional algorithms [9].

In summary, while significant progress has been made in object detection, challenges persist, particularly concerning the balance between accuracy and computing power demands. Addressing these challenges requires a concerted effort from the research community to develop algorithms that not only deliver superior accuracy but also exhibit efficiency in terms of computational requirements. By navigating these complexities, researchers aim to unlock the full potential of object detection technology, driving advancements across a myriad of domains.

2. LITERATURE SURVEY

Object detection, a crucial task in computer vision, remains challenging despite significant advancements [1]. Fast and accurate object detection is essential for various applications, such as robotics for strawberry flower pollination, where precise detection is crucial for yield estimation and robot development [4], [5]. From manual feature-based methods like VJ Det to deep learning-based approaches like the YOLO series, object detection has evolved rapidly [6]. However, both traditional and deep learning-based algorithms pose significant computing power demands, limiting their deployment on resource-constrained platforms like UAVs or mobile robots [7].

While traditional methods like VJ Det and DPM have shown improvements in speed and accuracy [9], the advent of CNNs has revolutionized object detection [9]. Models like R-CNN and YOLO have significantly improved detection speed and accuracy compared to traditional algorithms [9]. Despite these advancements, challenges persist. Some works focus solely on improving detection accuracy, neglecting computing power requirements, hindering deployment on embedded devices [8]. Additionally, overlooking algorithm parameters contributes to high computing power demands. Some approaches address this post-training, risking decreased accuracy [8].

In agricultural contexts, deep learning-based object detection has shown promise. Dias, Tabb, and Medeiros successfully detected apple flowers using deep convolutional networks [19]. Their approach facilitates tasks like yield estimation and orchard management. Moreover, Zhang et al. developed a real-time strawberry detection system using deep neural networks on embedded systems, enabling high-performance detection in agricultural environments [25]. However, deploying deep learning models on mobile devices poses challenges due to computational constraints [27]. ShuffleNet addresses this challenge, offering an efficient CNN architecture tailored for mobile platforms [27].

International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

Similarly, Liu et al. proposed a lightweight neural network framework for human activity recognition on mobile devices, achieving high performance with minimal computational complexity [28].

In conclusion, object detection research spans diverse applications and methodologies. While advancements have been made in accuracy and efficiency, challenges remain in balancing computational demands and deployment on resource-constrained platforms. Addressing these challenges requires interdisciplinary efforts to develop efficient and accurate object detection solutions for real-world applications.

3. METHODOLOGY

a) Proposed Work:

The proposed work introduces a novel YOLOv5-Conv-BN system tailored for efficient and high-performance strawberry flower detection. This system emphasizes lightweight parameter networks, leveraging convolutional and Batch Normalization layers to optimize computational efficiency. A key aspect of the proposed approach is its comparative analysis with other state-of-the-art models, including Faster R-CNN[33,34], YOLOv5x, SSD[35], and RetinaNet. This comparative analysis serves to evaluate the effectiveness and suitability of the YOLOv5-Conv-BN system in relation to existing methods. By assessing factors such as detection accuracy, speed, and computational efficiency, the proposed work aims to provide valuable insights into model selection for strawberry flower detection tasks. Overall, the YOLOv5-Conv-BN system represents a promising advancement in object detection methodologies, offering a balance between efficiency and performance for real-world applications in precision agriculture.

b) System Architecture:

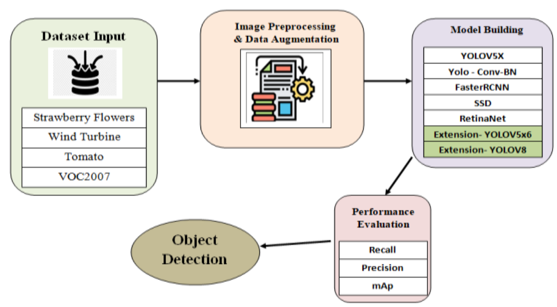


Fig 1 Proposed Architecture

The system architecture encompasses dataset input involving Strawberry Flowers, Wind Turbine, Tomato, and VOC2007, followed by image processing and augmentation. Model building integrates Faster R-CNN[33,34], SSD[35], EfficientDet RetinaNet, and various YOLO versions including YOLOv5X, YOLO-Conv-BN, YOLOv5x6, and YOLOv8. Performance evaluation metrics such as Recall, mAP, and Precision are utilized to assess model efficacy. Object detection is then conducted using the trained models, enabling accurate identification and localization of target objects. This comprehensive approach facilitates the selection of

International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

the most suitable object detection algorithm for precision agriculture and related applications.

c) Dataset:

*Strawberry Flowers:*The dataset includes images capturing various aspects of strawberry plants, focusing on flowers for detection tasks. These images showcase the diverse appearance and arrangement of strawberry flowers, presenting challenges such as variations in size, color, and clustering.

Fig 2 Flowers Dataset

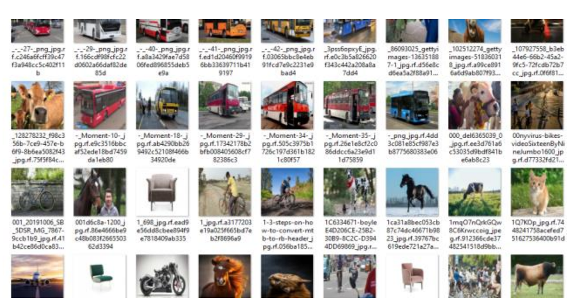
*Wind Turbine:*Images of wind turbines are included in the dataset, highlighting the detection of these structures in different environmental contexts. Variability in turbine size, orientation, and background scenery poses challenges for accurate detection algorithms.

Fig 3 Wind Turbine Dataset

*Tomato:*The dataset contains images of tomatoes, emphasizing detection tasks related to tomato plants and fruits. Variations in tomato ripeness, shape, and clustering patterns are observed, presenting challenges for object detection models.

Fig 4 Tomato Dataset

*VOC2007:*The VOC2007 dataset provides a comprehensive collection of images encompassing a wide range of object categories, including people, animals, vehicles, and household items. Exploration of this dataset allows for broader insights into object detection across diverse domains, facilitating comparative analysis with specialized datasets like Strawberry Flowers, Wind Turbine, and Tomato.



Reading the Image: The first step in dataset exploration involves programmatically reading the images from each category. This process enables access to pixel values, dimensions, and metadata, providing the foundation for further analysis and processing.

Plotting the Image: After reading the images, the next step is to plot them for visual inspection and analysis. Visualization aids in understanding the characteristics of the dataset, including variations in object appearance, lighting conditions, and backgrounds. This step is crucial for gaining insights into the dataset's complexities and informing subsequent steps in model development and evaluation.

d) Image Processing:

Converting to Blob Object: Images are converted into blob objects, which are a format suitable for input into neural network models. This process involves resizing the images to a predefined size, converting them to the appropriate color space, and normalizing pixel values.

Defining the Class: Classes are defined to represent the objects of interest in the dataset. Each class corresponds to a specific object category, such as "strawberry flower," "wind turbine," or "tomato."

Declaring the Bounding Box: Bounding boxes are declared to indicate the location and extent of objects within the images. Each bounding box is defined by its coordinates (x, y) and dimensions (width, height) relative to the image size.

Convert the Array to a Numpy Array: The image data is converted into a numpy array, a widely used data structure in Python for numerical computing. This allows for efficient manipulation and processing of image data within the Python environment.

Loading the Pre-trained Model:

Reading the Network Layers: The pre-trained model's architecture is read, comprising a series of interconnected layers responsible for processing input images and producing output predictions. This step involves loading the model's configuration and weights.

Extracting the Output Layers: Output layers are extracted from the pre-trained model, representing the final layers responsible for generating predictions. These layers typically produce output in the form of class probabilities and bounding box coordinates.

Appending the Image - Annotation File and Images: The image data is combined with annotation files containing information about object classes and bounding box coordinates. This facilitates supervised learning, where the model learns to associate image features with corresponding object labels.

Converting BGR to RGB: If necessary, images are converted from the BGR (Blue-Green-Red) color

International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

space to RGB (Red-Green-Blue). This ensures consistency in color representation across different platforms and libraries.

Creating the Mask: Masks are created to highlight regions of interest within the images. These masks can be used for tasks such as segmentation, where the goal is to partition images into meaningful regions corresponding to different objects.

Resizing the Image: Images are resized to a standardized size to ensure consistency in input dimensions across the dataset. This step is essential for compatibility with the pre-trained model and efficient processing during training and inference.

Data Augmentation:

Randomizing the Image: Images are randomly augmented to introduce variations in lighting, contrast, and other visual characteristics. This helps improve the model's robustness to different environmental conditions and enhances its generalization capabilities.

Rotating the Image: Images are rotated by a certain angle to simulate variations in object orientation. This augmentation technique helps the model learn to recognize objects from different viewpoints and angles.

Transforming the Image: Various transformations such as scaling, shearing, and flipping are applied to the images to further augment the dataset. These transformations introduce additional variations, making the model more robust and resilient to real-world conditions.

e) Algorithms:

YOLOv5X: YOLOv5X is an extension of the YOLO (You Only Look Once) object detection architecture, known for its real-time performance and high accuracy. YOLOv5X utilizes a deep neural network with a large number of layers and parameters to detect objects in images efficiently. It adopts a single-stage approach, where object detection and classification are performed simultaneously. YOLOv5X employs anchor boxes and feature pyramid networks to handle objects of varying sizes and scales effectively.

YOLO - Conv-BN: YOLO - Conv-BN is a variant of the YOLO architecture that incorporates batch normalization (BN) layers after convolutional layers. Batch normalization helps stabilize and accelerate the training process by normalizing the input to each layer. This variant improves the convergence speed and generalization performance of the YOLO[36,37] model by reducing internal covariate shift during training.

Faster R-CNN: Faster R-CNN is a two-stage object detection framework that consists of a region proposal network (RPN) followed by a region-based convolutional neural network (R-CNN). The RPN generates candidate object bounding boxes, which are then refined and classified by the R-CNN. Faster R-CNN[33,34] achieves high accuracy by leveraging a shared convolutional backbone for feature extraction and employing region-based strategies for object localization and classification.

SSD (Single Shot MultiBox Detector): SSD is a single-stage object detection framework that directly predicts object bounding boxes and class probabilities from feature maps. SSD[35] achieves real-time performance by simultaneously predicting multiple bounding boxes of different aspect ratios and scales at each feature map location. This enables efficient detection of objects at various sizes and aspect ratios without the need for region proposal generation.

RetinaNet: RetinaNet is a single-stage object detection architecture designed to address the imbalance between foreground and background classes in the training data. It introduces a novel focal loss function that focuses on hard examples during training, mitigating the problem of class imbalance. RetinaNet achieves high accuracy by effectively handling objects of varying sizes and complexities while maintaining real-time performance.

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

mAP: The mAP for object detection is the average of the AP calculated for all the classes. mAP@0.5 means that it is the mAP calculated at IOU threshold 0.5. The general definition for the Average Precision(AP) is finding the area under the precision-recall curve.

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k$$

$$AP_k = \text{the AP of class } k$$

$$n = \text{the number of classes}$$

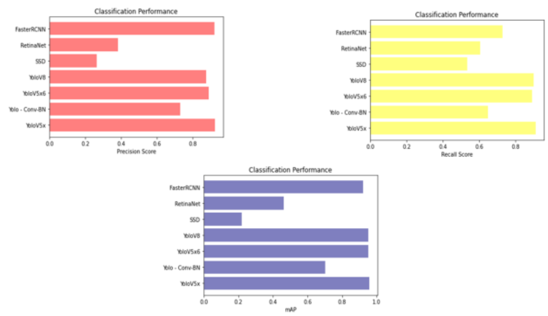


Fig 5 Comparison Graphs- Flowers Data

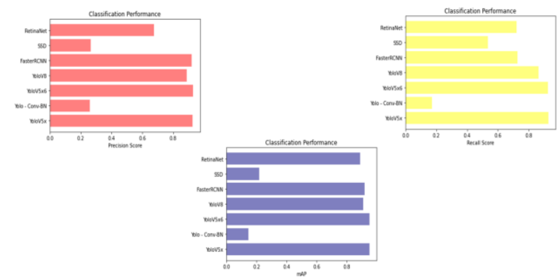


Fig 7 Comparison Graphs- Pascal Data

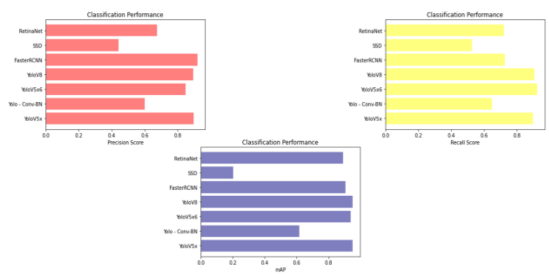


Fig 5 Comparison Graphs- Tomato Data

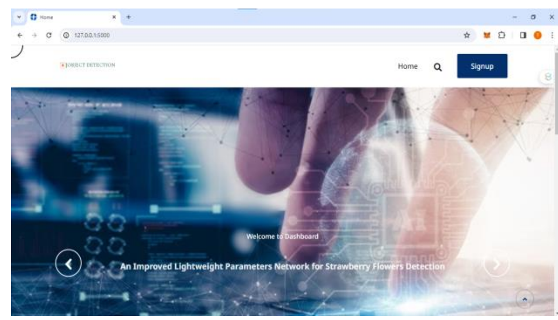


Fig 8 Home Page

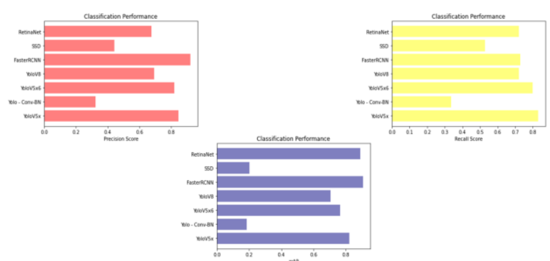


Fig 6 Comparison Graphs- Wind Turbines Data

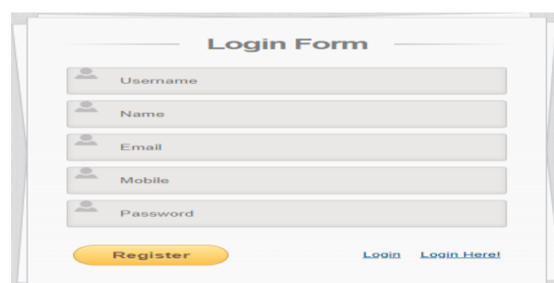


Fig 9 Registration Page

International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

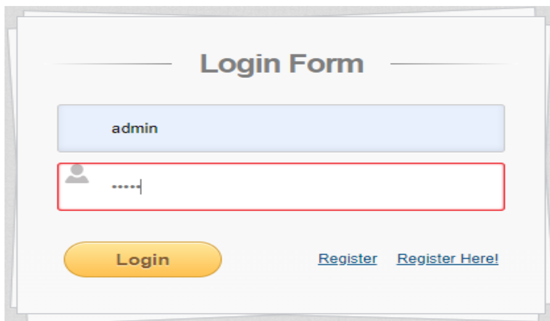


Fig 10 Login Page

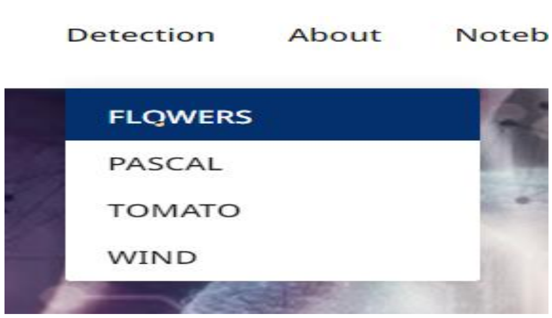


Fig 11 For Flowers

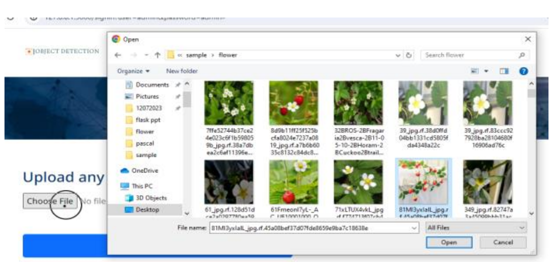


Fig 12 Upload Input Image

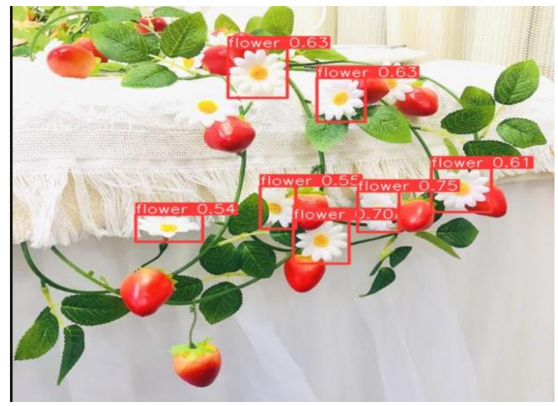


Fig 13 Predicted Results

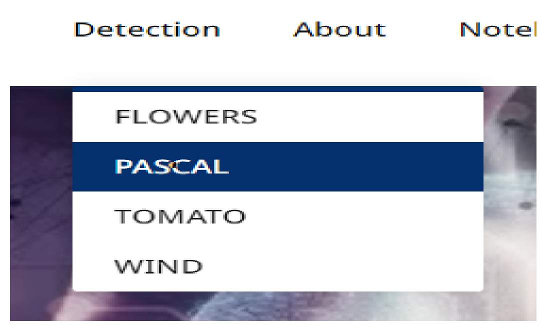


Fig 14 For Pascal

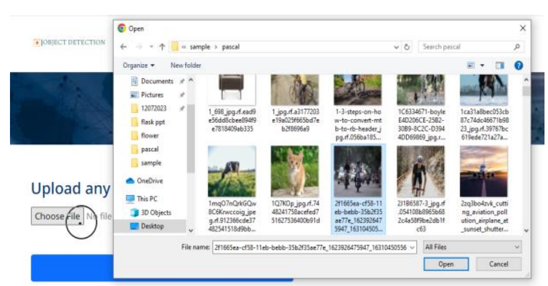


Fig 15 Upload Input Image

International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

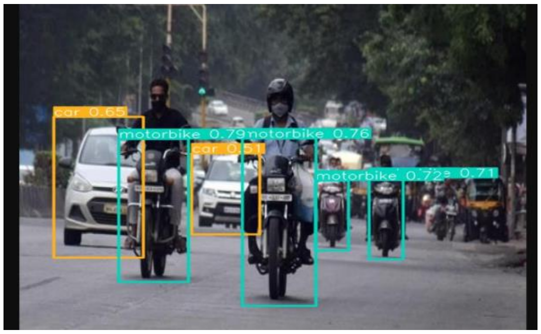


Fig 16 Final Outcome

Fig 19 Predicted Results

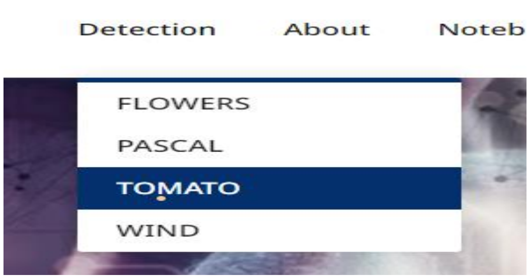


Fig 17 For Tomato

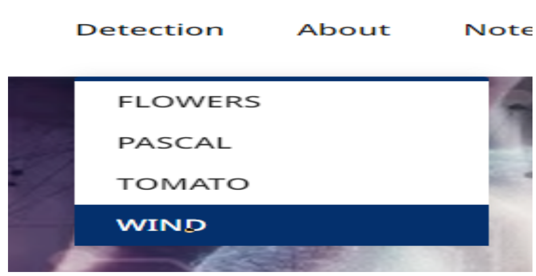


Fig 20 For Wind

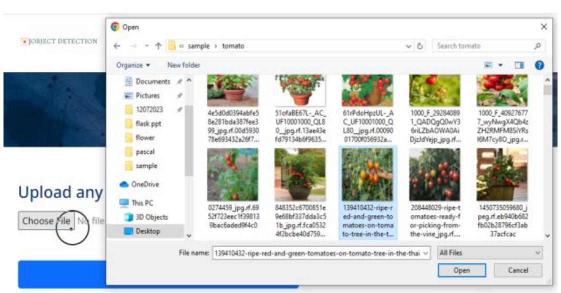


Fig 18 Upload Input Image

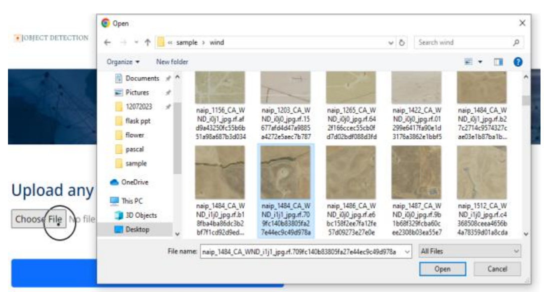


Fig 21 Upload Input Image

International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

Fig 22 Final Outcome

5. CONCLUSION

In summary, the exploration and assessment of various object detection algorithms, including Faster R-CNN[33,34], SSD[35], EfficientDet RetinaNet, and multiple versions of YOLO (You Only Look Once), have provided valuable insights into their performance across different computer vision applications, notably in precision agriculture. YOLO variants, particularly YOLOv5X[36] and YOLO with Conv-BN, demonstrated promising accuracy and efficiency, making them suitable for deployment on resource-constrained devices like mobile robots and UAVs. While Faster R-CNN, SSD, and EfficientDet RetinaNet offer higher precision, they often require more computational resources, limiting their practicality in real-world scenarios. Balancing accuracy, efficiency, and adaptability is crucial for deploying object detection algorithms effectively in industries such as precision agriculture, robotics, and UAV applications. Future research should focus on refining and optimizing these algorithms to address specific challenges, driving advancements in computer vision technology for practical applications.

6. FUTURE SCOPE

Looking ahead, future research could focus on several avenues to further enhance the capabilities and applications of the lightweight object detection model. Firstly, continued optimization and refinement of the network architecture could lead to even greater efficiency and accuracy, opening up possibilities for broader deployment across various industries and scenarios. Additionally, exploring advanced techniques such as transfer learning and ensemble methods could improve the model's generalization capabilities and robustness to diverse environmental conditions.

Moreover, extending the application of the model to other agricultural crops and objects beyond strawberry flowers could broaden its impact and relevance. Collaborations with agricultural experts and industry stakeholders could facilitate the integration of the proposed technology into real-world farming practices, driving innovation and efficiency in precision agriculture. Lastly, continued efforts in user interface development and integration with UAVs and mobile robots could enhance usability and accessibility, enabling seamless adoption and deployment in practical scenarios. Overall, the future holds promising opportunities for leveraging lightweight object detection models to address critical challenges and drive advancements in various fields, from agriculture to robotics and beyond.

REFERENCES

[1] K. V. Sriram and R. H. Havaldar, ‘‘Analytical review and study on object detection techniques in the image,’’ Int. J. Model., Simul., Sci. Comput., vol. 12, no. 5, Oct. 2021, Art. no. 2150031.

[2] X. Wu, D. Sahoo, and S. C. H. Hoi, ‘‘Recent advances in deep learning for object detection,’’ Neurocomputing, vol. 396, pp. 39–64, Jul. 2020.

[3] Z. Zhao, P. Zheng, S. Xu, and X. Wu, ‘‘Object detection with deep learning: A review,’’ IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[4] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, ‘‘Object detection in 20 years: A survey,’’ Proc. IEEE, vol. 111, no. 3, pp. 257–276, Mar. 2023.

[5] C. Zheng, A. Abd-Elrahman, and V. Whitaker, ‘‘Remote sensing and machine learning in crop phenotyping and management, with an emphasis on applications in strawberry farming,’’ Remote Sens., vol. 13, no. 3, p. 531, Feb. 2021.

[6] S. Zhao, J. Liu, and S. Wu, ‘‘Multiple disease detection method for greenhouse-cultivated strawberry based on multiscale feature fusion faster R_CNN,’’ Comput. Electron. Agricult., vol. 199, Aug. 2022, Art. no. 107176.

[7] J. Deng, Z. Shi, and C. Zhuo, ‘‘Energy-efficient real-time UAV object detection on embedded platforms,’’ IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 39, no. 10, pp. 3123–3127, Oct. 2020.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, ‘‘Object detection with discriminatively trained part-based models,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[9] W. Zhiqiang and L. Jun, ‘‘A review of object detection based on convolutional neural network,’’ in Proc. 36th Chin. Control Conf. (CCC), Dalian, China, Jul. 2017, pp. 11104–11109.

[10] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, ‘‘RepVGG: Making VGG-style ConvNets great again,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Nashville, TN, USA, Jun. 2021, pp. 13728–13737.

[11] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, ‘‘Path aggregation network for instance segmentation,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 8759–8768.

[12] Y. Liao, S. Lu, Z. Yang, and W. Liu, ‘‘Depthwise grouped convolution for object detection,’’ Mach. Vis. Appl., vol. 32, no. 6, pp. 1–13, Sep. 2021.

[13] Q. Lü, J. R. Cai, B. Liu, L. Deng, and Y. J. Zhang, ‘‘Identification of fruit and branch in natural scenes for citrus harvesting robot using machine vision and support vector machine,’’ Int. J. Agricult. Biol. Eng., vol. 7, no. 2, pp. 115–121, Nov. 2014.

[14] F. Kurtulmus, W. S. Lee, and A. Vardar, ‘‘Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network,’’ Precis. Agricult., vol. 15, no. 1, pp. 57–79, Feb. 2014.

[15] R. Horton, E. Cano, D. Bulanon, and E. Fallahi, ‘‘Peach flower monitoring using aerial multispectral imaging,’’ J. Imag., vol. 3, no. 1, p. 2, Jan. 2017.

[16] A. McCarthy and S. Raine, ‘‘Automated variety trial plot growth and flowering detection for maize and soybean using machine vision,’’ Comput Electron Agricult., vol. 194, Mar. 2022, Art. no. 106727, doi: 10.1016/j.compag.2022.106727.

[17] C. L. Zhou, W. S. Lee, and R. Lin, ‘‘Strawberry flower detection using fluorescence imaging,’’ in Proc. ASABE Annu. Int. Meeting, St. Joseph, MI, USA, Jul. 2020, pp. 13–15.

[18] P. Lin, W. S. Lee, Y. M. Chen, N. Peres, and C. Fraisse, ‘‘A deep-level region-based visual representation architecture for detecting strawberry flowers in an outdoor field,’’ Precis. Agricult., vol. 21, no. 2, pp. 387–402, Apr. 2020.

[19] P. A. Dias, A. Tabb, and H. Medeiros, ‘‘Apple flower detection using deep convolutional networks,’’ Comput. Ind., vol. 99, pp. 17–28, Aug. 2018.

[20] P. A. Dias, A. Tabb, and H. Medeiros, ‘‘Multispecies fruit flower detection using a refined semantic segmentation network,’’ IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 3003–3010, Oct. 2018.

[21] F. Palacios, G. Bueno, J. Salido, M. P. Diago, I. Hernández, and J. Tardaguila, ‘‘Automated grapevine flower detection and quantification method based on computer vision and deep learning from on-the-go imaging using a mobile sensing platform under field conditions,’’ Comput. Electron. Agricult., vol. 178, Nov. 2020, Art. no. 105796.

[22] Y. Tian, G. Yang, Z. Wang, E. Li, and Z. Liang, ‘‘Instance segmentation of apple flowers using the improved mask R-CNN model,’’ Biosyst. Eng., vol. 193, pp. 264–278, May 2020.

[23] G. Li, R. Suo, G. Zhao, C. Gao, L. Fu, F. Shi, J. Dhupia, R. Li, and Y. Cui, ‘‘Real-time detection of kiwifruit flower and bud simultaneously in orchard using YOLOv4 for robotic pollination,’’ Comput.

International journal of basic and applied research

www.pragatipublication.com

ISSN 2249-3352 (P) 2278-0505 (E)

Cosmos Impact Factor-5.86

Electron. Agricult., vol. 193, Feb. 2022, Art. no. 106641.

[24] D. Wu, S. Lv, M. Jiang, and H. Song, ‘‘Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments,’’ Comput. Electron. Agricult., vol. 178, Nov. 2020, Art. no. 105742.

[25] Y. Zhang, J. Yu, Y. Chen, W. Yang, W. Zhang, and Y. He, ‘‘Real-time strawberry detection using deep neural networks on embedded system (RTSD-Net): An edge AI application,’’ Comput. Electron. Agricult., vol. 192, Jan. 2022, Art. no. 106586.

[26] C. Wang, H. M. Liao, Y. Wu, P. Chen, J. Hsieh, and I. Yeh, ‘‘CSPNet: A new backbone that can enhance learning capability of CNN,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Seattle, WA, USA, Jun. 2020, pp. 1571–1580.

[27] X. Zhang, X. Zhou, M. Lin, and J. Sun, ‘‘ShuffleNet: An extremely efficient convolutional neural network for mobile devices,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 6848–6856.

[28] T. Liu, S. Wang, Y. Liu, W. Quan, and L. Zhang, ‘‘A lightweight neural network framework using linear grouped convolution for human activity recognition on mobile devices,’’ J. Supercomput., vol. 78, no. 5, pp. 6696–6716, Oct. 2021.

[29] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, ‘‘Feature pyramid networks for object detection,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, Jul. 2017, pp. 936–944.

[30] G. Li, M. Zhang, J. Li, F. Lv, and G. Tong, ‘‘Efficient densely connected convolutional neural networks,’’ Pattern Recognit., vol. 109, Jan. 2021, Art. no. 107610.

[31] M. Awais, M. T. B. Iqbal, and S. Bae, ‘‘Revisiting internal covariate shift for batch normalization,’’ IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 11, pp. 5082–5092, Nov. 2021.

[32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, ‘‘Grad-CAM: Visual explanations from deep networks via gradient-based localization,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Venice, Italy, Oct. 2017, pp. 618–626.

[33] S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster R-CNN: Towards realtime object detection with region proposal networks,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[34] Y. Mu, R. Feng, R. Ni, J. Li, T. Luo, T. Liu, X. Li, H. Gong, Y. Guo, Y. Sun, Y. Bao, S. Li, Y. Wang, and T. Hu, ''A faster R-CNN-based model for the identification of weed seedling,'' Agronomy, vol. 12, no. 11, p. 2867, Nov. 2022.

[35] H. Lu, C. Li, W. Chen, and Z. Jiang, ''A single shot multibox detector based on welding operation method for biometrics recognition in smart cities,'' Pattern Recognit. Lett., vol. 140, pp. 295–302, Dec. 2020.

[36] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, ''YOLOX: Exceeding YOLO series in 2021,'' 2021, arXiv:2107.08430.

[37] Y. Zhang, W. Zhang, J. Yu, L. He, J. Chen, and Y. He, ''Complete and accurate holly fruits counting using YOLOX object detection,'' Comput. Electron. Agricult., vol. 198, Jul. 2022, Art. no. 107062.

[38] M. Tan, R. Pang, and Q. V. Le, ''EfficientDet: Scalable and efficient object detection,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, Jun. 2020, pp. 10778–10787.

Dataset Link:
https://roboflow.com/convert/labelbox-json-to-yolov5-pytorch-txt