AGRICULTURE AND GREENHOUSE DETECTION FROM UAV IMAGES USING YOLOv5

Katakam Koushik[1] , S. Venkata Suryanarayana[2]

[1]PG Scholar, CVR College of Engineering/IT Department, Hyderabad, India

Email: 20B81DB004@cvr.ac.in

[2]Professor, CVR College of Engineering/IT Department, Hyderabad, India

Email: suryahcu@gmail.com

Abstract

Effective greenhouse mapping approaches are crucial for implementing sustainable farming practises, managing natural resources, and fostering sustainable urban and rural development. For monitoring and mapping greenhouses, remote sensing photography offers a lot of possibilities with various spatial and spectral resolutions. The traditional methods for greenhouse mapping are time- and money-consuming. A wide range of image processing techniques, including classification techniques using pixel or object-based classification, and remote sensing indices, have been used for greenhouse mapping. The growth and development of AG's essential for higher productivity. The proposed method utilizes the Agriculture Greenhouse data set containing two classes (agriculture, greenhouse) to train and test the model through YOLOv5 Object Detection algorithm. We compare the performance of YOLOv3, YOLOv5 Family while training them on a Agriculture Greenhouse dataset. This information will be helpful for practitioners to select the best technique based for the Agriculture and Greenhouse dataset.

Keywords: YOLOv5l, YOLOv5, YOLOv3, object detection, YOLOv5l6, YOLOv5x6.

INTRODUCTION

Agricultural greenhouses (AGs) are crucial used by modern agriculture to provide the market with farm goods. Many areas, including Europe, North Africa, and the Middle East, have seen a dramatic change in the agricultural environment as a result of the rapid expansion of AGs. However, the quick development and growth of AGs has raised some challenges with land management, such as occupied prime farmland, damaged soil, contamination from plastic wastes, etc. To grow AGs equitably and protect the farmland, they need a reliable detection system to keep track of their geographic distribution. The viewing environment, which includes the atmosphere, sensor quality, solar light, and surrounds, can influence how remotely sensed pictures seem and behave. In addition, AGs were covered with a variety of materials, which made them appear very fragmented and heterogeneous in remote sensing photographs. The thickness, transmission, and reflectivity of these materials varied. Therefore, under government management, the majority of AG detection was done by visual interpretation of remote sensing images.

On the basis of remote sensing images with high spatial resolution, recent developments in pattern recognition and machine learning, in particular, provide tremendous promise for autonomous information extraction from massive data sets. With the usage of deep learning models, which uses multi-layer neural networks to hierarchically characterise the most representative and discriminative properties. Deep learning has significantly advanced the disciplines of computer vision and natural language processing during the last few decades. The geoscience and remote sensing sectors have now enthusiastically embraced CNNs for applications including target extraction, terrain categorization, and object recognition.

Traditional Methods

Due to the long processing time, traditional approaches to object identification are not used real-time. Furthermore, the accuracy isn't up to par for the execution of actual applications. Non-neural networks require feature extraction, followed by classification using an SVM classifier, employing approaches such as the Viola-Jones object identification framework based on Histogram of Oriented Gradients(HOG) features.

Deep Learning Based Methods

Convolutional Neural Networks (CNN) for image processing were used in deep learning-based systems to eliminate these needless phases of feature engineering. CNN extracts features without the need for human involvement by automatically training and updating network parameters, resulting in increased speed, accuracy and performance. Deep learning-based approaches frequently have several parameter issues, making them computationally complicated and expensive, with a slow convergence rate. Traditional approaches are not only computationally expensive but they also perform poorly in terms of detection. Sparse representation-based approaches have been developed to address this issue. The basic functionality of Traditional and deep learning approaches for object detection is as shown in figure 1.

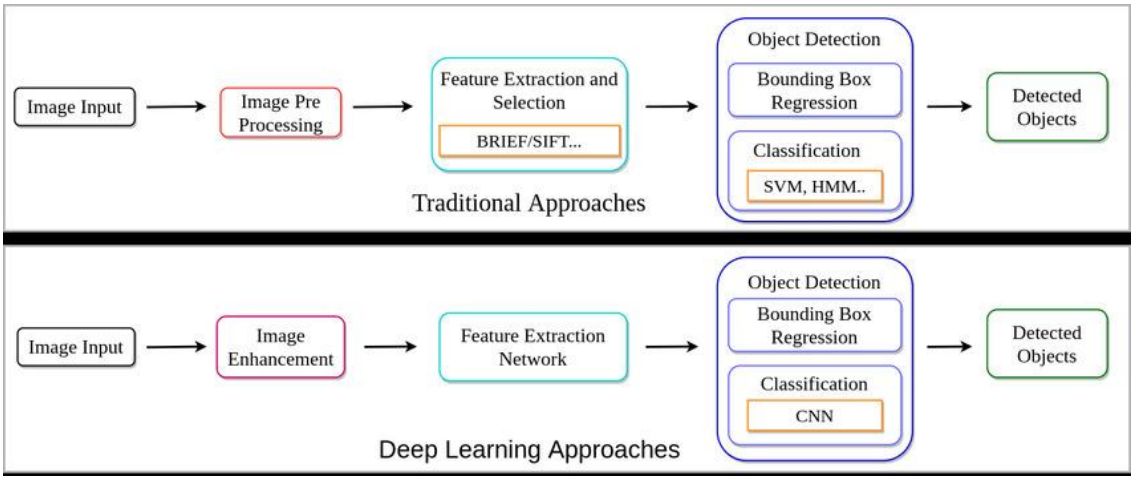

Figure 1 : Traditional and deep learning approaches

LITERATURE SURVEY

Kaiming He.et.al, [1] developed Mask R-CNN where Image Segmentation knowledge is required for finding the mask of an image. Mask R-CNN takes care of two forms of picture segmentation those are Semantic Segmentation and Instance Segmentation. Semantic segmentation is concerned with the pixel-level detection and grouping of similar entities into a single class. Because it emphasis the image's contents rather than the backdrop, semantic segmentation is also known as foreground segmentation. The goal of instance segmentation, also known as instance recognition, which is to accurately recognize and segment all objects in an image. Instance Segmentation recognizes each person as a separate entity, despite the fact that all things are persons. They observed that Mask R-CNN can only examine temporal information about the item of interest because it operates on still pictures. Mask R-CNN often fails to distinguish motion blurred objects in low resolution images.

In [2] the authors developed YOLOv1 witch is a real-time video processing neural network with a latency of less than 25 milliseconds. During training and evaluation, it scans the entire image and collects contextual information about courses as well as their look. The YOLOv1 architecture takes a picture and reduces it to 448*448. The photograph is subsequently sent to the CNN network. YOLOv1 last layer forecasts a cuboidal output. This is accomplished by altering the final completely linked layer (1, 1470). (7, 7, 30). The architecture was influenced by the GoogLeNet image classification model. The training and assessment criteria are the same for both the Fast and Slow YOLO. This YOLO model uses PASCAL VOC to test our network after pretraining it on the ImageNet 1000-class competition dataset. YOLOv1 has a lesser recall and a larger localization error when compared to Faster R-CNN. Because each grid can only propose two boundary boundaries, finding close things in detecting small images is a challenge in this model.

Joseph Redmon.et.al, [3] developed YOLOv2 which is trained on different architectures such as VGG-16, GoogleNet and darknet. Darknet was selected over alternative designs because it consumes less processing power 5.58 FLOPS. They introduced batch normalization into the design to increase the model's convergence, resulting in quicker training. Normalization methods like dropout is not used in this model because there is no overfitting. When compared to ordinary YOLO, they observed that merely adding batch normalization raises mAP by 2%.

In [4] the author demonstrated YOLO9000 was based on the YOLOv2 architecture and could identify over 9000 different classes.YOLO9000 proposed the Hierarchical Classification technique. They devised a hierarchical tree-based framework to represent classes and subclasses for categorizing the data. The classification network of YOLOv2 was finetuned for a total of ten epochs at full 448x448 resolution. WordTree employs conditional probability at every node level to do categorization. By multiplying the conditional probabilities of all its parents, the conditional probability of a leaf node (particular class) can be calculated. The architecture assumes Pr(Physical Object) =1 while defining "Physical object" as its root node. They computed mAP of YOLO9000 is 19.7%, with a 16 percent mAP on classes not included in the detection dataset. They also observed that YOLO9000 outperforms the COCO dataset when it comes to new animal species. The mAP is higher than the DPM model's calculation.The key benefit of YOLO9000 is that it can forecast more than 9000 classes in real time (9418 to be exact). By training a large-scale detector, they merged the COCO detection dataset with the top 9000 classes from the entire ImageNet release to produce a combined dataset. There are 9418 classes in the WordTree for ImageNet dataset. To keep the output size small, they employed the fundamental YOLOv2 architecture with only three boxes instead of five.

Joseph Redmon.et.al, [5] developed YOLOv3 which is a neural network that uses dimension clusters as anchor boxes to forecast bounding boxes. For class predictions, they applied binary cross-entropy loss during training and feature map from the two preceding layers is then upsampled by 2x. They extracted more significant semantic information from the upsampled features using this strategy. In general, COCO's average mean AP measure is comparable to SSD versions, but it is three times quicker. As the IOU

threshold rises, YOLO's performance suffers, showing that the boxes aren't completely aligned with the object. YOLOv3 predicts three bounding boxes per cell (versus five in YOLO v2), but at three different scales, totaling nine anchor boxes. YOLOv3 can predict at three distinct sizes due to its architectural uniqueness, using feature maps from layers 82, 94, and 106. On recognizing characteristics on three separate scales, YOLOv3 compensates for the shortcomings of YOLOv2 and YOLOv1. Finding close items is difficult since each grid can only offer three border boxes.

In [6] the authors developed YOLOv4 which is a 10% upgrade over YOLOv3, with a 12% increase in mean average accuracy (mAP) and a 12% increase in frames per second. Spatial attention module(SAM) [7] increases ResNet50-SE [8] top-1 accuracy on the ImageNet image classification job by 0.5 percent with just 0.1 percent more calculation in YOLOv4. The nice aspect is that it has no impact on inference performance on the GPU. They applied Bag of freebies, Bag of specials for backbone and detector. Bag of freebies(BOG) for backbone are cutmix, cutouts, dropblock, mosaic data augmentation and class label smoothing. Random patches were clipped and pasted between the training pictures. In YOLOv4, Mosaic is the first new data augmentation approach. The model could then be able to recognize things that are smaller than usual. It may also be used in training since it eliminates the requirement for a huge mini-batch size. Labels are often thought of as hard, binary assignments when doing image classification tasks. Bag of specials used for backbone and detector are mish activation [9] , cross stage partial connection (CSP), spatial pyramid pooling (SPP) [10] , spatial attention module [7]. Mish has considerably higher precision, reduced total loss and is smoother.

In [11] the authors developed YOLOv4-tiny which is the compressed version of YOLOv4. YOLOv4-tiny used to decrease parameters and simplify the network architecture using YOLOv4 so that this can be developed on devices. YOLOv4-tiny has nearly eight times the frame rate of YOLOv4. The YOLOv4-tiny model achieves 22.0 percent AP at 443 frames per second on the RTX 2080Ti, whereas the YOLOv4-tiny model achieves 1774 frames per second with TensorRT, batch size = 4 and FP16-precision. In a real-time object identification inference time is more important than precision or accuracy which can be acquired by YOLOv4-tiny.

Ross Girshick.et.al, [12] developed a R-CNN which is one of the first large-scale, successful convolutional neural network to many application areas like object identification, segmentation and localization. To circumvent the difficulty of picking a huge number of regions. They used a selective search technique is utilized, which extracts just 2000 parts from the image. A convolutional neural network is used to square the 2000 likely area predictions and feed them into a 4096-dimensional feature vector. The CNN acts as a feature extractor and the collected features are sent into an SVM [13], which categorizes the item's presence inside the candidate region suggestion. This approach predicts not only the presence of an object within a specific zone but also four offset values to improve bounding box precision. Although the computer may have predicted the presence of a person based on a region proposal, the face of that person may have been halved inside that area proposition. The major drawback in their approach is no learning happen due to the single algorithm is used that is selective search algorithm.

Ross Girshick.et.al, [14] developed a Fast R-CNN for detecting objects which is a more difficult process that necessitates the use of more complicated approaches. Models are generally trained in sluggish and inelegant multi-stage pipelines. They used a single-stage training method for categorizing and refining item suggestions at the same time. To classify each item proposition, SPPnet [15] employs a feature vector from the common feature map. They employed a revolutionary training strategy that improves the system's speed and accuracy while correcting errors. They also compared their Fast R-CNN performance with the use of VGG 16 which cuts training and testing times by 9 hours, from 84 to 9.5 hours and is 10 times quicker than SPPnet. Updating layers from conv3 and above was all that was required for VGG16 (9 of the 13 conv layers). Average Recall (AR) is the gold standard for evaluating object proposal quality. For various R-CNN proposal approaches, AR correlates well with mAP.

3 METHODOLOGY

Agriculture greenhouses (AGs) are a crucial building for the advancement of contemporary agriculture. For the strategic planning of contemporary agriculture, accurate and efficient AG detection is required.. We use YOLOv5 model to detect agriculture and greenhouse areas because YOLOv5 model has a focus layer witch cable to detect low level features accurately. The Workflow of YOLOv5 for the Crop and Weed dataset is as shown in Figure II.

YOLOv5

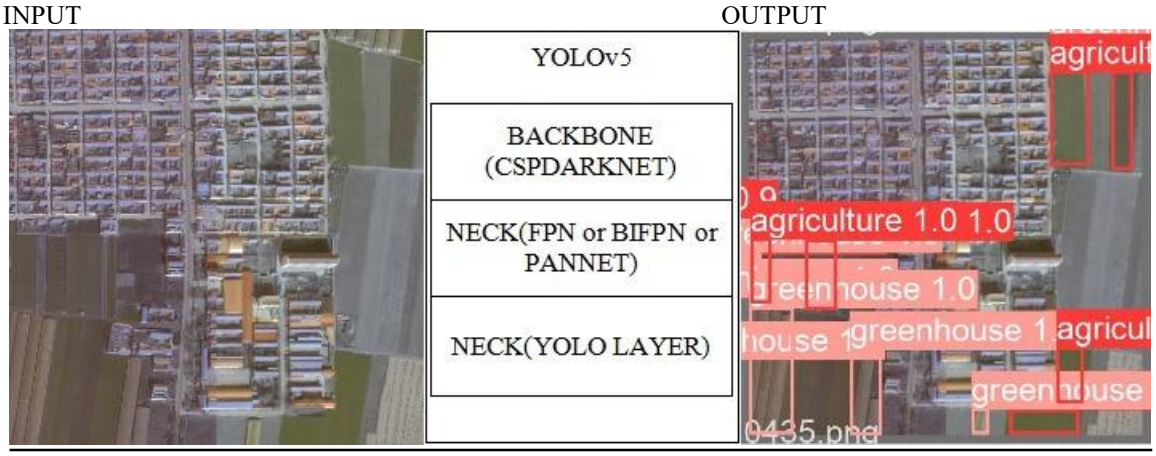

Figure 2 : Workflow of YOLOv5

YOLOv5 which uses the MS COCO[16] AP50.95 and AP50 is a cutting-edge detector that is both more accurate and faster (FPS) than any other detector in the market. They adopted substantial network enhancements to speed up the network's performance in terms of mean average precision (mAP). In FPN[17] top-down augmentation path is used and in PAN bottom-up data augmentation path is used. YOLOv5 comes with 30 unique training hyper parameters . It is possible to think of the learning rate as a step size that keeps the expense of each repetition to a bare minimum. To prevent overfitting, the learning rate needs to be carefully chosen. How many pictures will be transmitted to the network in a single transmission depends on the batch size. Thus, using a bigger batch size will speed up training. The potential for poor generalization when employing larger batch sizes must also be taken into account. Since the image size relates to the size of the input network, each image is reduced to 416x416 before being given to the network (in our example, 416 pixels).

The different versions of YOLOv5 based on p5 and p6 models as shown in Table 1. The YOLOv5 architecture comprises the backbone (CSPDarknet), the neck (PANet), and the head (YOLO Layer), as seen in Figure 3. Model Backbone's primary function is to draw out crucial details from an input image witch uses the CSP (Cross Stage Partial Networks) architecture for this purpose. Model's Neck is used to arrange features into pyramids. FPN or BIFPN or PANET can be used as Model's Neck. The model Head is mostly in charge of the final detecting step. With the use of anchor boxes, bounding boxes, objectness scores, and class probabilities, it creates final output vectors. The Architecture of YOLOv5 is as shown in Figure 3.

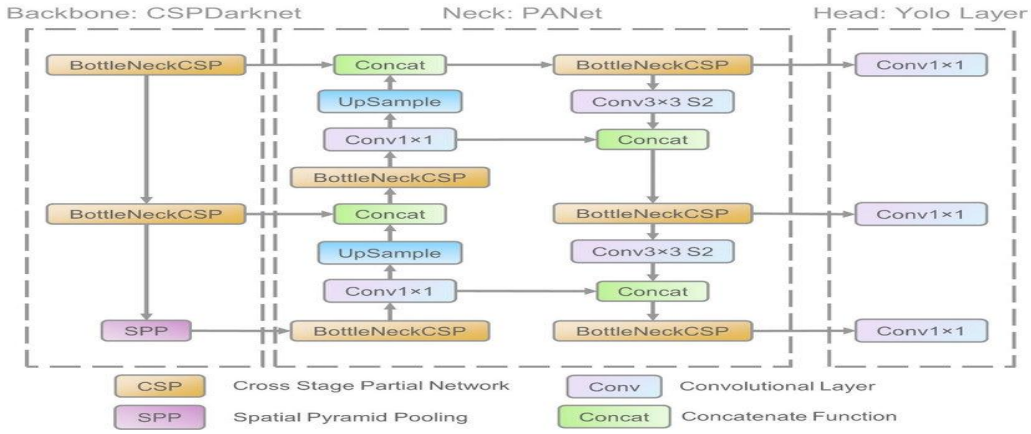

Figure 3: YOLOv5 Architecture

Table 1: Various types of YOLOv5 models

S.no	YOLOv5p5 Models	YOLOv5p6 Models
1	YOLOv5n	YOLOv5n6
2	YOLOv5s	YOLOv5s6
3	YOLOv5m	YOLOv5m6
4	YOLOv5l	YOLOv5l6
5	YOLOv5x	YOLOv5x6

RESULTS:

Experimental Setup

To Implement YOLOv5 model we require the following Hardware and Software support.
Hardware requirements: Processor: i3 or better,RAM: 8GB or More,Storage: 120GB or More,Nvidia GPU Recommended are used.
Software requirements:,OS: Windows,Python3.7 or later,Pytorch,OpenCVOther necessary Python Modules are used.
Precision, Recall, F1 score, map 0.5, map 0.5:0.95 are used to evaluate the performance of the model.

DATASET:(Agriculture and Greenhouse)

The Agriculture and Greenhouse dataset contain images from agriculture lands collected using UAV's.The Agriculture and Greenhouse detection dataset contains 701 images across two classes namely agriculture and greenhouse. The whole dataset is divided into train/test sets by ratio 70% and 30% within each event class so 501 used for training and 200 images used for testing. The dataset collected from https://github.com/thealejandroperilla/ProtectedAgriculture-GEE.git website.
Sample output of agriculture and greenhouse for YOLOv5x6 shown in Figure 4 with probability values.

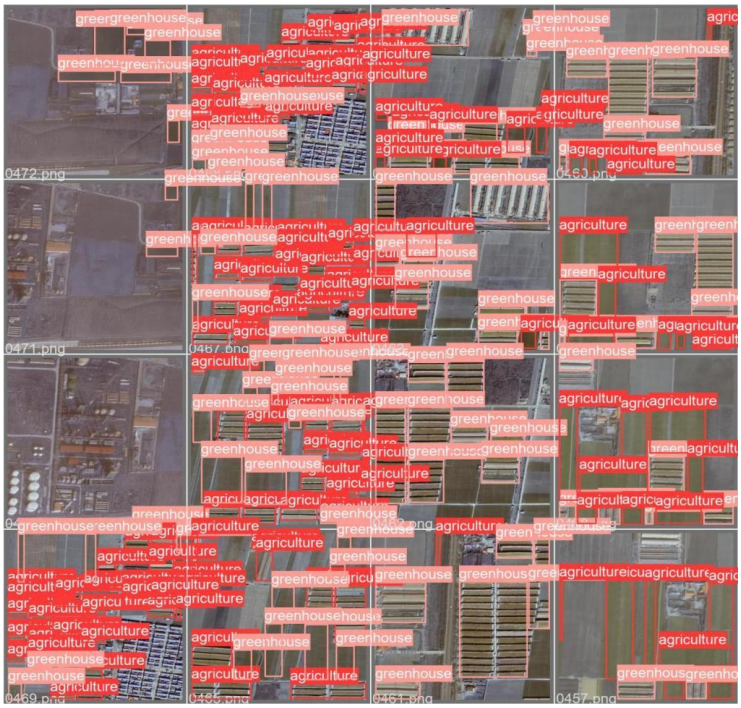

Figure 4: sample output of crop and weed for YOLOv5x6

The performance of YOLOv3 and YOLOv5 family for the given image dataset is listed in the Table 2, Table 3 and Table 4. We also compared the accuracy of all models in the form of bar chart as shown in Figure 5.

Table 2: Precision,Recall,F1 score,map 0.5 and map 0.5 to 0.95 values for agriculture

YOLOv3 and YOLOv5 models	agriculture				
	map 0.5	map 0.5:0.95	Precision	Recall	F1 score
YOLOv3-tiny	94.6	62.6	0.854	0.898	0.8754
YOLOv3	99.5	94.5	0.992	0.997	0.9944
YOLOv3-spp	99.5	93.8	0.993	0.995	0.99399
Yolov5n	97.9	70	0.938	0.947	0.94247
Yolov5s	99.4	84	0.981	0.989	0.98498
Yolov5m	99.5	92.6	0.99	0.995	0.99249
Yolov5l	99.5	95.8	0.994	0.995	0.99449
Yolov5x	99.5	97.2	0.996	0.993	0.99449
Yolov5n6	98.9	79.5	0.962	0.965	0.96349
Yolov5s6	99.5	90.8	0.987	0.993	0.98999
Yolov5m6	99.5	95.7	0.991	0.997	0.99399
Yolov5l6	99.5	97.2	0.992	0.998	0.99499
Yolov5x6	99.5	98.2	0.994	0.996	0.99499

Table 3: Precision,Recall,F1 score,map 0.5 and map 0.5 to 0.95 values for greenhouse

YOLOv3 and YOLOv5 models	greenhouse				
	map 0.5	map 0.5:0.95	Precision	Recall	F1 score
YOLOv3-tiny	96.8	67	0.9.6	0.926	0.91589
YOLOv3	99.5	93.4	0.992	0.992	0.992
YOLOv3-spp	99.5	92.4	0.99	0.994	0.99199
Yolov5n	98.7	73	0.955	0.96	0.95749
Yolov5s	99.4	83.7	0.985	0.983	0.98399
Yolov5m	99.5	91.4	0.99	0.992	0.99099
Yolov5l	99.5	94.6	0.993	0.991	0.99199
Yolov5x	99.5	96.1	0.993	0.992	0.992499

Yolov5n6	99.1	79.9	0.972	0.967	0.96949
Yolov5s6	99.5	89.4	0.985	0.991	0.98799
Yolov5m6	99.5	94.4	0.99	0.995	0.99249
Yolov5l6	99.5	96	0.989	0.997	0.99749
Yolov5x6	99.5	97	0.991	0.994	0.99249

Table 4: Precision,Recall,F1 score,map 0.5 and map 0.5 to 0.95 values for all(agriculture and greenhouse)

YOLOv3 and YOLOv5 models	All(agriculture & greenhouse)				
	map 0.5	map 0.5:0.95	Precision	Recall	F1 score
YOLOv3-tiny	95.7	64.8	0.88	0.912	0.8957
YOLOv3	99.5	94	0.992	0.994	0.99299
YOLOv3-spp	99.5	93.1	0.991	0.995	0.99299
Yolov5n	98.3	71.5	0.947	0.953	0.94999
Yolov5s	99.4	83.9	0.983	0.986	0.98449
Yolov5m	99.5	92	0.99	0.994	0.99199
Yolov5l	99.5	95.2	0.993	0.993	0.993
Yolov5x	99.5	96.6	0.995	0.992	0.99349
Yolov5n6	99	79.7	0.967	0.966	0.9664
Yolov5s6	99.5	90.1	0.986	0.992	0.98899
Yolov5m6	99.5	95.1	0.99	0.996	0.99299
Yolov5l6	99.5	96.6	0.99	0.998	0.99398
Yolov5x6	99.5	97.6	0.992	0.995	0.99349

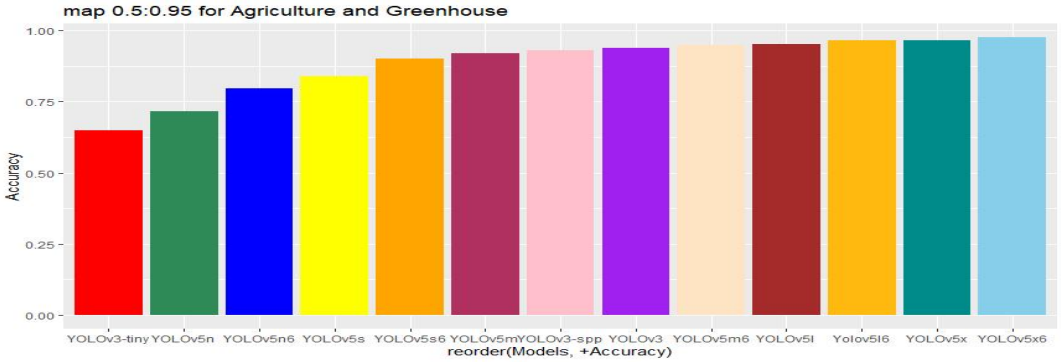

Figure 5 : Accuracy comparisons of all models for All(agriculture and greenhouse) category

Conclusion

In most computer and robot vision systems, object detection is crucial. Despite recent improvements and the implementation of some current approaches into various consumer devices or assistance driving systems. This study introduces a real-time Agriculture Greenhouse detection using advanced image processing to achieve higher productivity. Accuracy (map 0.5), Accuracy (map 0..5 to 0.95), Precision, Recall and F1 score for YOLOv5l6 model are 99.5, 96.6, 0.99, 0.998, 0.99398 respectively. Accuracy (map 0.5), Accuracy (map 0..5 to 0.95), Precision, Recall and F1 score for YOLOv5x6 model are 99.5, 97.6, 0.992, 0.995, 0.99349 respectively. Therefore, we can affirm that YOLOv5x6 can be preferred than YOLOv3 family and other models of YOLOv5 family like YOLOv5l6 for the considered AG's dataset for Agriculture Greenhouse detection with appropriate speed and accuracy.

References

[1] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2017.322

[2] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.91

[3] Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., & Cai, B. (2018). An improved YOLOv2 for vehicle detection. Sensors, 18(12), 4272. https://doi.org/10.3390/s18124272

[4] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2017.690

[5] Du, K., Song, J., Wang, X., Li, X., & Lin, J. (2020). A multi-object grasping detection based on the improvement of YOLOv3 algorithm. 2020 Chinese Control And Decision Conference (CCDC). https://doi.org/10.1109/ccdc49329.2020.9164792

[6] Wang, C., Bochkovskiy, A., & Liao, H. M. (2021). Scaled-yolov4: Scaling cross stage partial network. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr46437.2021.01283
[7] Woo, S., Park, J., Lee, J., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. Computer Vision – ECCV 2018, 3-19. https://doi.org/10.1007/978-3-030-01234-2_1
[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.90
[9] Mercioni, M. A., & Holban, S. (2021). Soft clipping Mish - A novel activation function for deep learning. 2021 4th International Conference on Information and Computer Technologies (ICICT). https://doi.org/10.1109/icict52872.2021.00010
[10] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep Convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904-1916. https://doi.org/10.1109/tpami.2015.2389824
[11] Pal, J. B. (2019). Real time object detection Canbe embedded on low powere devices.https://doi.org/10.31219/osf.io/t3gdy
[12] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2014.81
[13] Mu, n., & qiao, d. (2019). image classification based on convolutional neural network and support vector machine. 2019 6th international conference on information, cybernetics, and computational social systems (iccss). https://doi.org/10.1109/iccss48103.2019.9115443
[14] Girshick, R. (2015). Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2015.169 He, K., Zhang, X., Ren, S., & Sun, J. (2015).
[15] Spatial pyramid pooling in deep Convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904-1916. https://doi.org/10.1109/tpami.2015.2389824
[16] Ning, Z., Wu, X., Yang, J., & Yang, Y. (2021). MT-yolov5: Mobile terminal table detection model based on YOLOv5. Journal of Physics: Conference Series, 1978(1), 012010. https://doi.org/10.1088/1742-6596/1978/1/012010
[17] Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2017.106