CA-YOLO: Ensembled Model Optimization for Remote Sensing Image Object Detection

G. Uma Devi, Assistant Professor, Department of CSE, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India.

kudevi@gmail.com

Barla Sai Meghana, Aratikatla Anusha, Duggineni Nikhila, Bokara Bhavani, Students, Department of CSE, Raghu Engineering College, Visakhapatnam, Andhra Pradesh, India.

20981a0515@raghuenggcollege.in, 21985a0502@raghuenggcollege.in, 20981a0541@raghuenggcollege.in, 20981a0520@raghuenggcollege.in

Abstract: Multiple object retrieval methods are solved by the CA-YOLO model to find elements in complex spatial images. This addresses weak multiscale feature learning and the delicate balance between model complexity and performance. CA-YOLO is based on YOLOv5 and uses a small coordinate attention module at the lower layer to recover selected features and reduce data. A fast spatial pyramid with tandem design modules at the bottom uses stochastic pooling to speed up thinking and merge features of different sizes. Anchor box interactions and missing features have been updated to make it easier to find items of different sizes and shapes. CA-YOLO outperforms YOLO in multiple object detection and average inference speed of 125 frames/sec. CA-YOLO is a great option in the same conditions and difficulties. The study also investigated V3 tiny, V4, V5s, V8s, CA Yolos, and V5x6 YOLO models. This suggests that YOLO V5x6 can achieve more than 95% mAP on remote sensing object identification datasets.

Index terms - Object detection, attention mechanism, coordinate attention, SPPF, SIoU loss.

1. INTRODUCTION

Satellite imagery is important for intelligent transportation, urban planning, agriculture, disaster relief, environmental tracking, military operations, and public safety [1]. Intelligent interpretation relies heavily on object identification, including location and classification. Image processing was further developed using convolutional neural networks (CNN). AlexNet won the 2012 ImageNet Competition for its feature representation and classification [2].

Since then, CNN-based object identification research has grown to improve feature extraction for detection and classification [3]. CNN-based object detection algorithms have two stages and one stage. Use classification and regression categories. R-CNN's two-step approach first selects a bounding box, then performs classification and regression. Running takes a long time. Many advances, such as SPPnet and more powerful R-CNN models, address these issues [4–8]. In this paper, we examine the accuracy-speed tradeoff of CNN-based object identification algorithms in two-stage and one-stage approaches over time. This study highlights the importance of CNNs as the basis for many object identification algorithms.

The one-step method combines classification and location regression. Used by SSD [9], RetinaNet [10], YOLO [11], [12], [13], etc. This one-step approach is fast, but less accurate than previous approaches.

Researchers investigated the use of regression-based algorithms to identify elements in remote sensing photographs. Although these approaches are faster than region proposal-based methods, they are less effective. CNN architecture is used for object detection. Remote sensing images are more sophisticated and feature many small objects of varying sizes spread over a large area. Therefore, speed and accuracy may be reduced.

CA-YOLO algorithm improves his one-step method. Uses the YOLOv5 backbone. The backbone of the YOLOv5 network module extracts features, and the head combines those features.

2. LITERATURE SURVEY

[1] This study reviews the existing literature and discusses the shortcomings of deep learning-based object detection in remote sensing images. A lot of work has been done in this area, but the information provided has problems such as insufficient sharing of images or objects, which hinders deep learning algorithms. This work explores recent advances in computer vision and the use of deep learning to identify objects in real-world observations. Due to problems with the available data, the authors propose a large scale called DIOR, which stands for "Optical Image Detection of Remote Sensing." The collection includes 23,463 images and 192,472 samples from 20 product categories. DIOR solves the main problem by offering a wide range of products in different ways such as weather, season and quality. The blueprint makes it easier for professionals to design and test data-driven methods. The DIOR dataset is used to test state-of-the-art techniques for object detection in optical imaging, laying the foundation for future research.

[2] This article specifically explains how to find objects that have reached their limits in the PASCAL VOC dataset. The authors proposed a new method, a numerical analysis that can increase the accuracy (mAP) from almost 30% of the highest value of VOC 2012 to 53.3%. The large convolutional neural networks (CNN) for the following regional recommendations provide preliminary training for individual activities as well as accuracy of regional objects and segmentation, which is a special asset, especially in the absence of this teaching material. Support. R-CNN (regions with CNN features) outperforms the CNN-based sliding window detector OverFeat on strict 200-class ILSVRC2013 detection data. This work offers a simple but effective way to solve the problem. It shows how regional recommendations and CNNs can improve accuracy and establish new benchmarks.

[3] This article presents image classification using deep convolutional neural networks (CNN). The proposed neural network outperforms previous models. Trained using 1.2 million high-resolution ImageNet LSVRC-2010 images. Using the model, the accuracy of image classification was improved to a top-1 error of 37.5% and a top-to-bottom error of 17.0%. A CNN has 600,000 neurons and 60 million parameters. It has three fully connected layers ending with a thousand-way softmax and five convolutional layers with a max-pooling layer. GPU applications that train faster and without overloading neurons are important innovations. The author uses a new method of "dropping", which is quite effective against overloading in the connection process. In the ILSVRC-2012 event, the first 5 test errors of this model were 15.3%, and this rate was 26.2% higher than the second model. This research reanalyzes image distribution and shows how deeply CNNs can transform visual mass.

[4] For visual images, the performance of spatial pyramid pooling network (SPP-net) is better than deep convolutional neural network (CNN). SPP-net uses spatial pyramid pooling to create a long-range model that can be used to solve the large-scale problem of CNNs regardless of image size or scale. This new method works on different objects and improves on ImageNet 2012's CNN design. SPP-net's classification score on the Pascal VOC 2007 and Caltech101 datasets (with the full image and not necessarily corrected) is good. This affects the information of the product. SPP-net computes and distributes maps 24-102 times faster than R-CNN while maintaining or exceeding Pascal VOC 2007 accuracy. Among 38 teams participating in the 2014 ImageNet Large-Scale Visual Recognition Competition (ILSVRC), the proposed method ranked second in object detection and third in image classification. SPP-net effectively improves visual performance.

[6] To create safety signs for drivers, this article discusses the use of images as a new technique to immediately locate objects. Faster region-based convolutional neural network (Faster R-CNN) is the best choice for such tasks as it is equally fast and accurate. Faster R-CNN combines the advantages of RPN and Fast-RCNN algorithms. This work improves video performance by using GPUs during training and testing. It runs at 15 fps at 3000 frames in 4 groups. The file includes images of three lighting levels and a stop sign. Faster R-CNN performs well in real-time target detection. This shows that technology can improve driving safety by recognizing traffic signs.

3. METHODOLOGY

i) Proposed Work:

The suggested method uses CA-YOLO, an improved model based on YOLOv5, to locate various items in satellite photos. A lightweight coordinate attention module in the shallow layer helps CA-YOLO handle complicated distant sensing images. This increases detailed feature extraction and reduces extraneous information. A spatial pyramid pooling-fast with tandem construction module is added to the lowest tier. This random pooling method combines crucial feature information from several scales and levels.

This reduces model factors and accelerates reasoning. The enhanced anchor box architecture and loss function help the model locate things of various sizes and shapes. V3-tiny, V4, V5s, V8s, and CA-Yolos form the basis of the proposed system. It also examines how YOLO V5x6 might boost speed. CA-YOLO excels in remote sensing object detection, achieving 94% mAP on RSOD datasets. More work with YOLO V5x6 should improve recognition accuracy over 95% mAP.

ii) System Architecture:

CA-YOLO, a superior single-stage approach based on YOLOv5, is added to the recommended system architecture to enhance remote sensing object detection. The design addresses complex distant sensing issues using novel approaches. A lightweight coordinate attention module in the bottom layer improves detailed feature extraction and reduces useless information. A tandem building module and spatial pyramid pooling-fast module are in the lowest tier. This strategic design combines essential feature data from multiple sizes across layers using stochastic pooling. This reduces model parameters and accelerates inference. Improvements to the anchor box method and loss function allow the model to locate things of many sizes and scales. This comprehensive design examines V3-tiny, V4, V5s, V8s, CA-Yolos, and YOLO V5x6 models using YOLOv5. This makes the system robust and versatile enough to locate things in tough distant sensing environments.

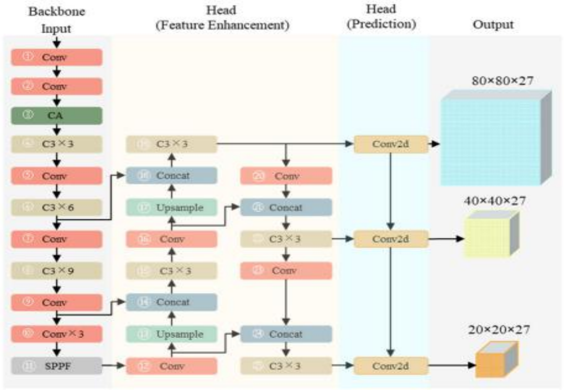


Fig 1 System Architecture

iii) Dataset Collection:

The dataset exploration includes three different datasets that get bigger over time: RSOD, NWPU VHR-10, and DOTA. RSOD has 976 pictures, with 40 images labeled as background and 936 images labeled as objects, including images of airplanes, oil tanks, overpasses, and playgrounds. A carefully calculated 6:2:2 split of the information is used to create training, validation, and test sets. NWPU VHR-10 has 800 pictures, with 650 labeled images of objects and 150 labeled images of backgrounds. The images are organized into 10 object groups. The large DOTA collection, on the other hand, has 2806 remote sensing pictures that have been carefully classified into 15 groups. This set of data comes from many different places, like Google Earth, the GF-2 and JL-1 satellites, and overhead photos from CycloMedia B.V. Notably, DOTA uses both RGB and sepia photos, which give a full picture of what might happen in real life. The dataset is very rich because it has a lot of different picture sources and has been carefully labeled. This makes it a great tool for training and testing object recognition models in the field of remote sensing.

iv) Image Processing:

Image Processing:

Converting to Blob Object: The first step in handling a picture is to change it into a blob object. This process changes the image's size to fit the network's input needs, evens out the values of the pixels, and rearranges the channels. The blob object that is made is an organized description of the picture that can be used to feed more information into a deep learning model.

Defining the Class and Declaring Bounding Box: Once the blob has been converted, classes are set up to find the items that are of interest. Bounding boxes are set up around these classes to define the edges of each object's space. This step sets the stage for the next step, which is object recognition. It gives us important data for training and testing models.

Convert the Array to a NumPy Array: The blob object is changed into a NumPy array to make working with it easier. NumPy arrays are fast and flexible, so they can be easily used with deep learning tools. This conversion makes it easy to work with and change picture data in later steps of processing.

Loading the Pre-trained Model:

Reading the Network Layers: Reading the network layers is a way to understand how a pre-trained model is put together before you load it. This step makes sure that the model's structure is compatible and easy to understand. This makes fine-tuning or feature extraction easier in later steps.

*Extracting the Output Layers:*The output layers are gotten after the model is loaded. Feature maps and class scores made during the forward pass are stored in these layers. To get forecasts and understand what the model knows about the original picture, you need to extract the output layers.

Image Processing (Continued):

Appending Image Annotation Files and Images: In this step, image annotation files that contain information about the real world are paired with the pictures that go with them. This combination makes a large sample that is needed to train and test the model. It also lets the program learn from cases that have been labeled.

Converting BGR to RGB: Because different libraries have different ways of representing colors, the picture needs to be changed from BGR to RGB. This orientation makes sure that the colors are shown the same way on all devices, so the picture is ready to be processed and shown later.

Creating the Mask and Resizing the Image: A mask is made to show important parts of the picture, which helps with the next step, which is feature extraction. At the same time, the picture is resized to a standard size, which makes sure that the model always gets inputs of the same size. This step is necessary to keep things consistent and reliable across different datasets and situations.

v) Data Augmentation:

Randomizing the Image: Adding more data to training datasets for machine learning models is a key part of making them more robust and diverse. Randomizing pictures by performing random changes is a basic method that adds variety. This includes changing the model's sharpness, contrast, and color strength to give it a wider range of viewing situations. Randomization prevents overfitting by showing the model different versions of the same object. This makes it better at adapting to new data.

Rotating the Image: Rotation is a key data enhancement technique that helps us understand object positions in the dataset more fully. By rotating pictures at random angles, the model learns to spot objects from different angles. This makes it better prepared for real-life situations where objects may show at different angles. This method for adding on helps keep the model from being too dependent on certain object positions that were in the original dataset. This makes it easier for the model to change to new cases.

Transforming the Image: Transformation, which includes scaling, splitting, and turning, changes the shapes of the pictures during enhancement. By modeling different spatial relationships between items, this method makes the collection more diverse. You can change the size by scaling, the shape by cutting, and the image by flipping it horizontally or vertically. The model becomes more resistant to changes in size, shape, and direction after being exposed to these changed cases. Overall, adding randomness, spin, and change to data strengthens machine learning models, making it easier for them to work with data they haven't seen before and making them better at using what they've learned in real life.

vi) Algorithms:

YOLO V3-tiny: YOLO V3-tiny is a small object recognition method that works best in real-time settings. Because it requires less complicated computing, it can be used effectively in places with limited resources. YOLO V3-tiny was chosen for our project because it strikes a good mix between speed and accuracy. This makes it a good choice for remote sensing picture analysis where finding multiple items quickly is very important.

YOLO V4: YOLO V4, the latest model in the YOLO line, has cutting-edge features that make it more accurate at finding objects. Its accuracy is improved by the use of modern building features. YOLO V4 was chosen for our project because it has cutting-edge features that make it both good at finding things in complex remote sensing situations and efficient at using computers.

YOLO V5s: Algorithm Definition: The YOLO V5s, which is part of the YOLOv5 series, is known for its simpler design and better speed. Because it works well, YOLO V5s meets the needs of our project's real-time object recognition. It can work with different types of remote sensing images and focuses on accuracy and speed, which are all things that the project needs.

YOLO V8s:A version of YOLO with more features, called YOLO V8s, finds a good mix between model complexity and computing speed. Its improved design makes it more accurate at finding objects at all sizes. YOLO V8s is used in our project to solve problems in remote sensing picture analysis, where finding items of all sizes and shapes accurately is very important.

CA-YOLOs: CA-YOLO is designed to find objects in pictures from remote sensing that are very complicated. It has a lightweight coordinate focus tool that makes feature extraction better and cuts down on unnecessary work. In our project, CA-YOLO was picked because it is more accurate, faster, and more flexible when finding multiple objects. It also solves some of the biggest problems that algorithms face in remote sensing apps.

Yolo V5x6: Yolo V5x6, which is an expanded version of Yolo V5, improves the ability to learn features on multiple scales. Because it works better at finding items of different sizes, it can be used in a variety of remote sensing situations. YOLO V5x6 was chosen for our project because it improves the model's ability to find objects in complex settings. This makes sure that object recognition works well and correctly in a range of picture situations.

4. EXPERIMENTAL RESULTS

Precision: Precision is the percentage of correctly classified cases or samples compared to those that were correctly classified as hits. So, here is the method to figure out the precision:

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

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

Recall: In machine learning, recall is a parameter that shows how well a model can find all the important cases of a certain class. It shows how well a model captures cases of a certain class. It is calculated by dividing the number of correctly predicted positive observations by the total number of real positives.

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

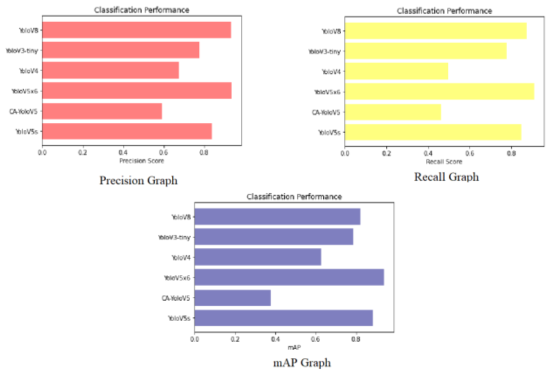
mAP: Mean Average Precision (MAP) is a way to measure quality and rank things. It looks at how many related suggestions there are and where they are on the list. To find MAP at K, take the average precision (AP) at K for all users or searches and multiply it by 100.

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

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

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

COMPARISON GRAPHS – RSOD DATASET



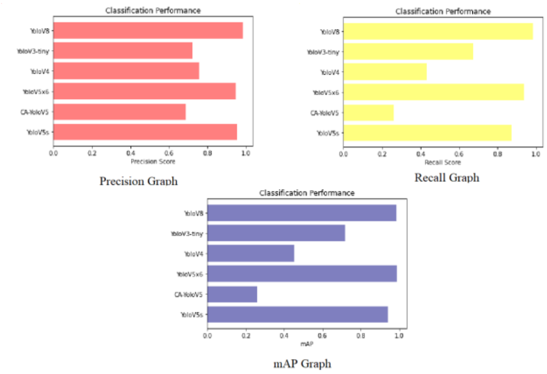
Precision Graph

Recall Graph

mAP Graph

Fig 2 Precision, Recall, mAP Comparison graph of RSOD dataset

COMPARISON GRAPHS – NWPU-VHR-10 DATASET



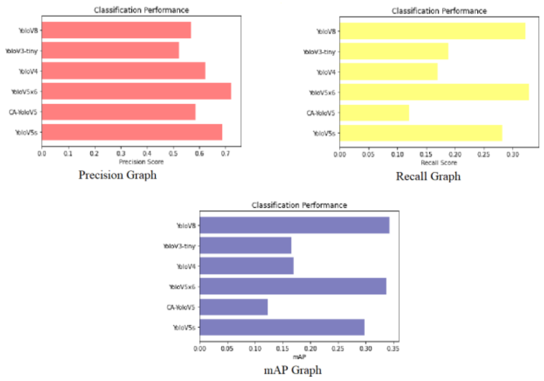
Precision Graph

Recall Graph

mAP Graph

Fig 3 Precision, Recall, mAP Comparison graph of NWPU-VHR-10 dataset

COMPARISON GRAPHS – DOTA DATASET

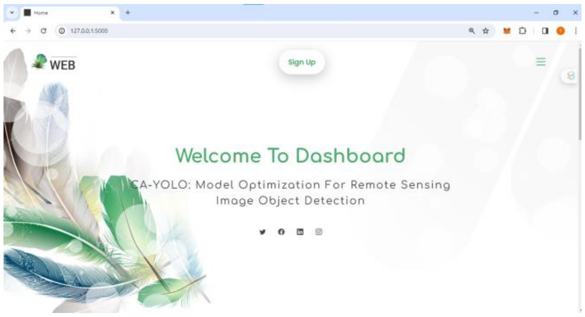


Precision Graph

Recall Graph

mAP Graph

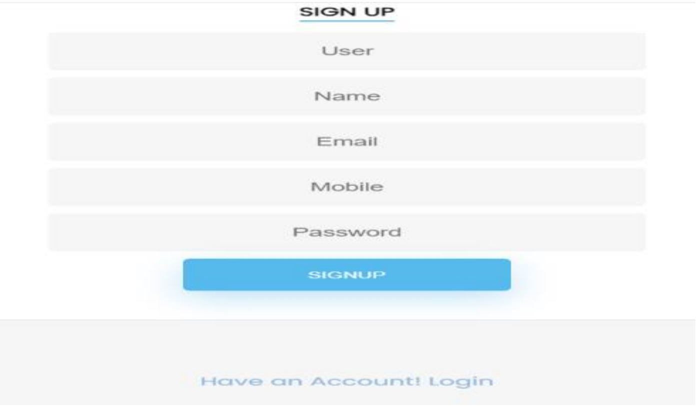
Fig 4 Precision, Recall, mAP Comparison graph of DOTA dataset



Welcome To Dashboard

CA-YOLO: Model Optimization For Remote Sensing Image Object Detection

Fig 5 Home page


SIGN UP

User

Name

Email

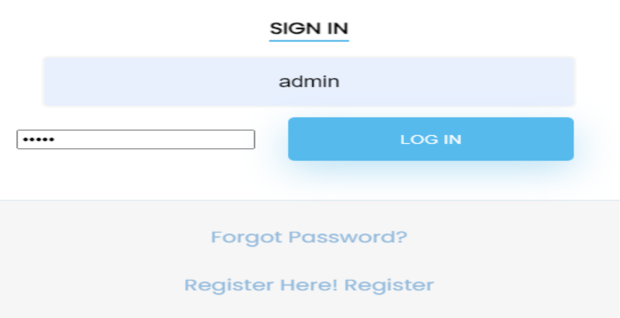
Mobile

Password

SIGNUP

Have an Account! Login

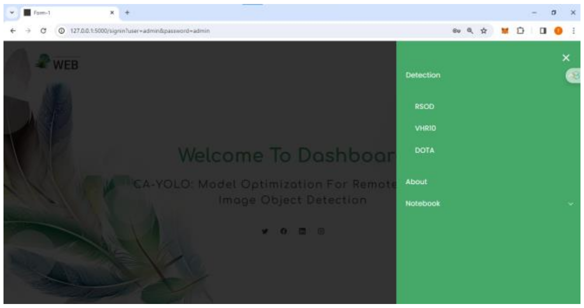
Fig 6 Registration page



admin

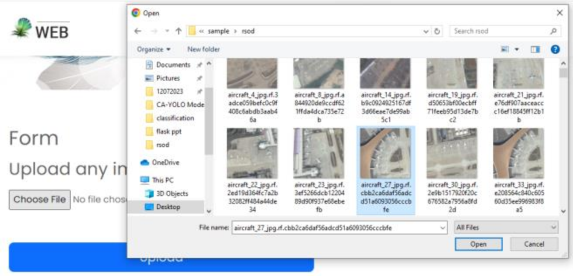
•••••

Fig 6 Login page



127.0.0.1:5000/signin?user=admin&password=admin

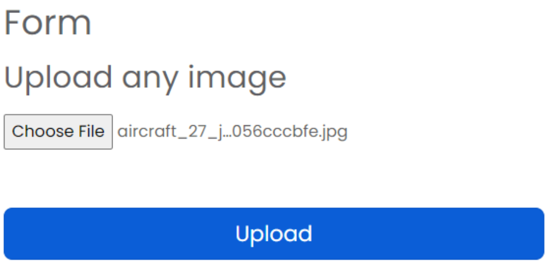
Fig 7 Main page



aircraft_4_jpg.rf.3adce059befc0c9f408c6abdb3aab46a
aircraft_8_jpg.rf.a844920de9ccdf621ffda4dca735e72b
aircraft_14_jpg.rf.b9c0924925167df3d66eae7de99ab5c1
aircraft_19_jpg.rf.d50653bf00ecbff71feeb95d13de7bc2
aircraft_21_jpg.rf.e76df907aaceaccc16ef18845ff12b1b
aircraft_22_jpg.rf.2ed19d364fc7a2b32082ff484a44de34
aircraft_23_jpg.rf.3ef5266dcb1220489d90f937e68ebefb
aircraft_27_jpg.rf.cbb2ca6daf56adcd51a6093056cccbfe
aircraft_30_jpg.rf.2e9b1517920f20c676582a7956a8fd2d
aircraft_33_jpg.rf.e208564c840c60560d35ee996983f8a5

File name: aircraft_27_jpg.rf.cbb2ca6daf56adcd51a6093056cccbfe

Fig 8 RSOD dataset input images folder



aircraft_27_j...056cccbfe.jpg

Fig 9 Upload input image

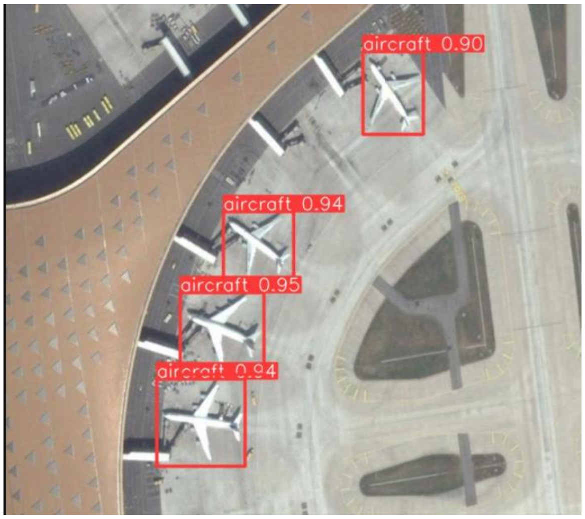
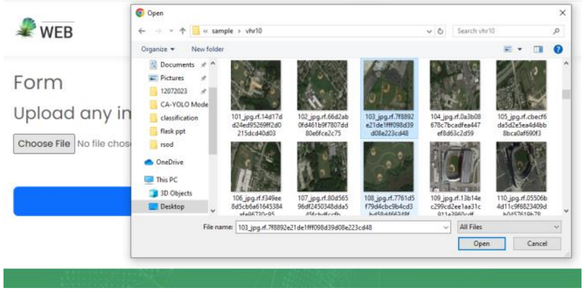


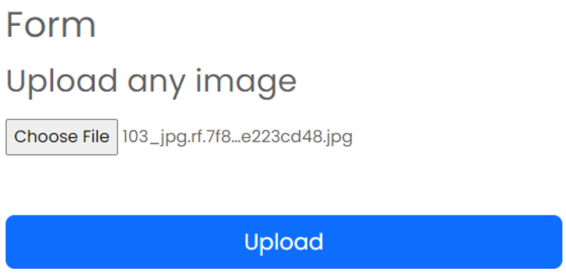
Fig 10 Predict result



101_jpg.rf.14d17dd24ed95269ff2d0215dcd40d03
102_jpg.rf.66d2ab0fd461b9f7807dd80e6fce2c75
103_jpg.rf.7f8892e21de1fff098d39d08e223cd48
104_jpg.rf.0a3b08678c7bcadfea447ef8d63c2d59
105_jpg.rf.cbecf6da5d2e5ea4d4bb8bca0af690f3
106_jpg.rf.f349ee8d5cb6a61645384
107_jpg.rf.80d56596df2450348dda5
108_jpg.rf.7761d5f79d4cbc9b4cd3
109_jpg.rf.13b14ec299cd2ee1aa31c
110_jpg.rf.05506b4d11c9f6823409d

File name: 103_jpg.rf.7f8892e21de1fff098d39d08e223cd48

Fig 11 NWPU-VHR-10 dataset input images folder



Form

Upload any image

103_jpg.rf.7f8...e223cd48.jpg

Fig 12 Upload input image

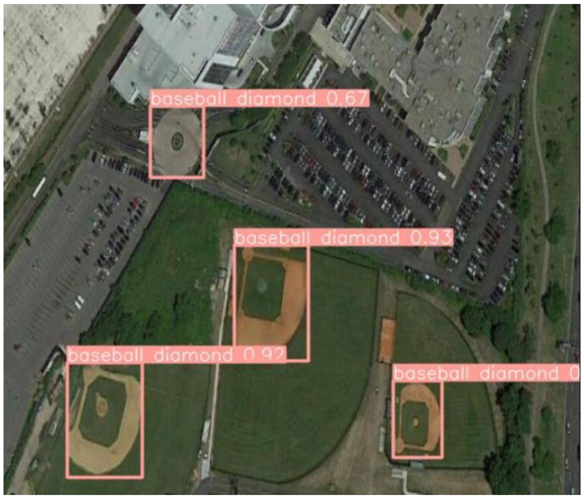
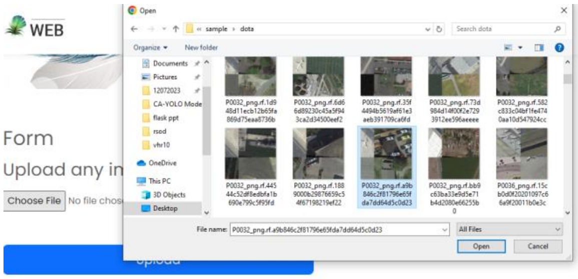


Fig 13 Final outcome


WEB
Open
sample › dota
Search dota
Organize
New folder
Documents
Pictures
12072023
CA-YOLO Mode
flask ppt
rsod
vhr10
OneDrive
This PC
3D Objects
Desktop
P0032_png.rf.1d948d11ecb12b65fa869d75eaa8736b
P0032_png.rf.6d66d89230c45a5f943ca2d34500eef2
P0032_png.rf.35f4494b5619af61e3aeb391709ca6fd
P0032_png.rf.73d984d14f00f2e7293912ee596aeeee
P0032_png.rf.582c833c04bf1fe4740aa10d547924cc
P0032_png.rf.44544c52df8edbfa1b690e799c5f95fd
P0032_png.rf.1889000b29876659c54f67198219ef22
P0032_png.rf.a9b846c2f81796e65fda7dd64d5c0d23
P0032_png.rf.bb9c63ba33e9d5e71b4d2080e66255b0
P0036_png.rf.15cb0d0f20201097c66a9f20011b0e3c
All Files
Open
Cancel
Choose File
Upload

Form

Upload any in

No file chos

File name: P0032_png.rf.a9b846c2f81796e65fda7dd64d5c0d23

Fig 14 DOTA dataset upload input images

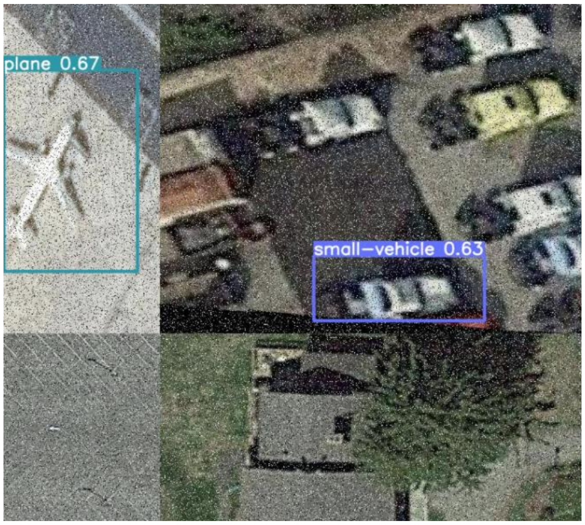


Fig 15 Predict result for given input

5. CONCLUSION

Finally, this paper presents a better CA-YOLO model that can successfully deal with problems in finding multiple objects of different sizes in remote sensing photos. The model improves feature extraction and reduces interference from duplicate information by adding a coordinate attention method to the YOLOv5 series. This helps solve problems related to low accuracy and weak generalization. Adding a tandem building module for Spatial Pyramid Pooling-Fast (SPPF) helps learn and combine features from different scales, which speeds up inference and improves the accuracy of identification. Using a mix of K-Means clustering and genetic algorithms to improve anchor boxes makes sure they are more aligned with goal sizes in the dataset.

The SIoU_loss loss function finds the best weight and makes target recognition work better. The CA-YOLO model works very well and is more accurate at finding things and putting them into groups than other YOLO-based algorithms. Notably, it gets an amazing 94% mAP for the RSOD dataset, which shows how much better it is. Further research into methods such as YOLO V5x6 could lead to even better spotting accuracy, possibly hitting 95%mPA or higher. This work confirms CA-YOLO as a reliable and effective method for remote-sensing picture analysis, achieving a good mix between accuracy, the ability to generalize, and the speed of inference compared to other models.

6. FUTURE SCOPE

The CA-YOLO model could be improved for real-time use and different weather situations through more study. Adding new technologies like cloud computing and AI-driven processes can make the model more useful in real life. Another thing that makes CA-YOLO a cutting-edge answer for new problems in remote-sensing picture analysis is that training techniques and dataset addition methods are always being tested and improved.

REFERENCES

[1] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, ‘‘Object detection in optical remote sensing images: A survey and a new benchmark,’’ ISPRS J. Photogramm. Remote Sens., vol. 159, pp. 296–307, Jan. 2020, doi: 10.1016/j.isprsjprs.2019.11.023.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ‘‘Rich feature hierarchies for accurate object detection and semantic segmentation,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classification with deep convolutional neural networks,’’ Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[4] K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Spatial pyramid pooling in deep convolutional networks for visual recognition,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[5] R. Girshick, ‘‘Fast R-CNN,’’ in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[6] R. Gavrilescu, C. Zet, C. Foşalău, M. Skoczylas, and D. Cotovanu, ‘‘Faster R-CNN: An approach to real-time object detection,’’ in Proc. Int. Conf. Expo. Electr. Power Eng. (EPE), Oct. 2018, pp. 165–168, doi: 10.1109/ICEPE.2018.8559776.

[7] Z. Cai and N. Vasconcelos, ‘‘Cascade R-CNN: Delving into high quality object detection,’’ in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6154–6162, doi: 10.1109/CVPR.2018.00644.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ‘‘Mask R-CNN,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, Feb. 2020, doi: 10.1109/TPAMI.2018.2844175.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ‘‘SSD: Single shot multibox detector,’’ in Proc. Eur. Conf. Comput. Vis., in Lecture Notes in Computer Science, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[10] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, ''Focal loss for dense object detection,'' in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ''You only look once: Unified, real-time object detection,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[12] J. Redmon and A. Farhadi, ''YOLO9000: Better, faster, stronger,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[13] J. Redmon and A. Farhadi, ''YOLOv3: An incremental improvement,'' 2018, arXiv:1804.02767.

[14] T. Kong, A. Yao, Y. Chen, and F. Sun, ''HyperNet: Towards accurate region proposal generation and joint object detection,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 845–853, doi: 10.1109/CVPR.2016.98.

[15] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, ''A unified multi-scale deep convolutional neural network for fast object detection,'' in Computer Vision—ECCV 2016 (Lecture Notes in Computer Science). Springer, 2016, pp. 354–370, doi: 10.1007/978-3-319-46493-0_22.