YOLOrs: Object Detection in Multimodal Remote Sensing Imagery

Manish Sharma, *Student Member, IEEE*, Mayur Dhanaraj, *Student Member, IEEE*, Srivallabha Karnam, Dimitris G. Chachlakis, *Member, IEEE*, Raymond Ptucha, *Senior Member, IEEE*, Panos P. Markopoulos, *Member, IEEE*, and Eli Saber, *Senior Member, IEEE*

***Abstract*—Deep-learning object detection methods that are designed for computer vision applications tend to underperform when applied to remote sensing data. This is because contrary to computer vision, in remote sensing, training data are harder to collect and targets can be very small, occupying only a few pixels in the entire image, and exhibit arbitrary perspective transformations. Detection performance can improve by fusing data from multiple remote sensing modalities, including red, green, blue, infrared, hyperspectral, multispectral, synthetic aperture radar, and light detection and ranging, to name a few. In this article, we propose YOLOrs: a new convolutional neural network, specifically designed for real-time object detection in multimodal remote sensing imagery. YOLOrs can detect objects at multiple scales, with smaller receptive fields to account for small targets, as well as predict target orientations. In addition, YOLOrs introduces a novel mid-level fusion architecture that renders it applicable to multimodal aerial imagery. Our experimental studies compare YOLOrs with contemporary alternatives and corroborate its merits.**

***Index Terms*—Aerial imagery, fusion, multimodal, object detection, remote sensing (RS).**

I. Introduction

OBJECT detection is a fundamental task in computer vision and remote sensing (RS) with a plethora of civilian and military applications, including medical diagnosis, autonomous-vehicle navigation, surveillance, and search-and-rescue operations, to name a few [1], [2]. An object detection algorithm strives to jointly identify the location of each object within an image and the class to which this object belongs. Traditional machine learning (ML) approaches rely on extracting various features (e.g., edges, color histogram, and corners) from each image. These features are then given as input to a learning algorithm, which, in turn, performs classification of the objects, as shown in [3] and [4]. Popular ML approaches that are still widely used are the Haar-features-based classification [5] and the support-vector-machine (SVM) classifier using histogram of oriented gradient features, among others [6]. In contrast, deep learning (DL) methods perform classification and localization of objects jointly [7]–[10]. In order to determine whether a standard ML or a DL approach is more appropriate for a particular problem, one has to consider the amount of available training data and computational power. In general, DL approaches tend to exhibit superior performance when there is an abundance of training data and sufficient computational power so that the model can train in a reasonable amount of time.

In RS, images are acquired by satellite, aircraft, and more recently, drone sensor technologies. Therefore, in contrast to computer vision, RS training data are harder to collect. Red, green, and blue (RGB) cameras, light detection and ranging, synthetic aperture radar, and infrared (IR) are typical sensor technologies that are widely used in RS. Due to the way in which remote images are captured, a few major challenges arise. First, objects can be very small with respect to the size of the image, offering little feature information. For example, an object may occupy just few tens of pixels in a multimillion pixel image. Second, objects are oriented arbitrarily and object detection algorithms need to learn rotation-invariant features in order to attain higher detection performance [11]. Moreover, the scale of objects within and across images can vary significantly. For example, the scale of a car will greatly differ from the scale of an airplane [12]–[17]. Finally, noise signals, occlusions, and compression artifacts are also challenges that need to be addressed by object detection algorithms.

In this work, we propose YOLOrs: a new convolutional neural network, specifically designed for real-time object detection in multimodal RS imagery. The proposed YOLOrs model is capable of detecting rotated bounding boxes and performs detection at a larger scale compared to YOLOv3 [18]. Therefore, YOLOrs is able to better detect rotated and closely spaced small objects in an aerial imagery setting. In addition, motivated by the use of multimodal data, as explained in Section II-B, we present

Manuscript received July 31, 2020; revised September 28, 2020 and November 11, 2020; accepted November 13, 2020. Date of publication November 30, 2020; date of current version January 8, 2021. This work was supported in part by an academic grant from the National Geospatial-Intelligence Agency under Award # HM0476-19-1-2014, Project Title: Target Detection/Tracking, and Activity Recognition from Multimodal Data, in part by the National Science Foundation under Grant OAC-1 808 582, and in part by the Air Force Office of Scientific Research (Young Investigator Program) under Grant FA9550-20-1-0039. *(Corresponding author: Panos P. Markopoulos.)*

Manish Sharma is with the Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: ms8515@rit.edu).

Mayur Dhanaraj, Dimitris G. Chachlakis, and Panos P. Markopoulos are with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: mxd6023@rit.edu; dimitris@mail.rit.edu; pxmeee@rit.edu).

Srivallabha Karnam and Raymond Ptucha are with the Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: sk3715@rit.edu; rwpeec@rit.edu).

Eli Saber is with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY 14623 USA, and also with the Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: esseee@rit.edu).

Digital Object Identifier 10.1109/JSTARS.2020.3041316

an extended version of YOLOrs with the ability to conduct mid-level fusion and combine data from multiple RS modalities. The proposed fusion further improves the detection performance of YOLOrs, as it is verified in a series of experimental studies.

II. Background

A. Object Detection With DL

DL architectures can be classified as two-stage, single-shot, and anchor-free. Two-stage architectures first generate a large number of proposed region candidates (e.g., by means of the selective search algorithm [19]) and then perform classification at each region. Region-based convolutional neural network (R-CNN) [20] is one of the first successful two-stage methods. In R-CNN, candidate regions, in the form of bounding boxes, are given as input to a CNN that extracts class features and, in turn, passes them to a SVM classifier. Arguably, the computational efficiency of R-CNN is limited due to its dependence on heavy region proposal algorithms. Fast-R-CNN [21] addresses this problem by performing feature extraction over the image before proposing regions and replacing the SVM classifier by a softmax layer, which extends the CNN for predictions instead of a separate model. Faster-R-CNN [22] introduces a CNN-based region proposal network, omitting the use of a selective search algorithm, which further improves the inference speed, making it suitable for real-time applications. These approaches exhibit high detection performance, but tend to be computationally demanding.

In contrast, single-shot approaches combine the detection and classification steps by jointly predicting the class of an object and its bounding box. You only look once (YOLO) [23], YOLOv2 [24], YOLOv3 [18], and single-shot multibox detector (SSD) [25] are successful single-shot architectures with real-time processing capabilities. YOLO trains a single end-to-end CNN that jointly predicts bounding boxes and object class labels, significantly increasing processing speed, compared to its predecessors. YOLOv2 relies on fully convolutional layers and tuned priors on bounding boxes instead of predicting heights and widths. YOLOv3 makes the bounding box predictions at three different scales making it suitable to identify objects of multiple spatial resolutions. YOLOv4 [26] is able to further increase the detection performance and computation speed of YOLOv3 by employing heavy data augmentation, evolved activation functions, and improved IoU loss metrics. SSD is similar to the YOLO architectures but performs independent detection using multiscale feature maps. YOLOv4 is the latest iteration in the YOLO series and explores different single-shot detectors tend to exhibit moderate-to-high detection performance with faster detection speeds.

Both two-stage and single-shot approaches rely on horizontally aligned bounding boxes, which fall short in providing tight bounding-box predictions. Oriented bounding boxes approximate object shapes more tightly compared to horizontally aligned ones. Liao *et al.* [27] introduced a single-shot oriented text detector that can predict bounding boxes with arbitrary rotations, varying sizes, and different aspect ratios. Similarly, Nosaka *et al.* [28] used a regression branch to extract rotation-sensitive features by actively rotating the conventional filters. Liu *et al.* [29] introduced RR-CNN that is built on R-CNN and utilizes rotated-region-of-interest pooling layers, an auxiliary structure to extract features of rotated regions. Lie *et al.* [30] modified YOLOv3 to predict oriented bounding boxes for RS applications. Finally, Terrial and Jurie [31] introduced faster RER-CNN, which is based upon faster R-CNN and is able to regress oriented bounding boxes.

In contrast to two-stage and single-shot architectures that rely on anchors for the localization of objects by introducing a lot of hyperparameters in the model, anchor-free architectures omit the process of anchor-based sliding window and perform detection in a pixelwise fashion, similar to semantic segmentation. CornerNet [32] and fully convolutional one-stage object detection [33] are early examples of anchor-free architectures. These architectures aim to reduce the number of hyperparameters and, thus, the excess training time. Moreover, anchor-free architectures can simplify object detection and lead to training speed enhancements.

Similar to general object detection, both ML and DL approaches have been proposed for object detection in aerial imagery [34], [35]. Single-shot networks that employ multiple detection scale sizes and consider multiple target orientations have been shown to achieve a promising tradeoff between execution time and detection performance, rendering them appropriate for real-time applications.

B. Object Detection in Multimodal Data

Fusion of data collected across multiple modalities (multimodal data) can allow for enhanced inference [36]–[39]. Similarly, the performance of object detection in RS can further improve by leveraging multimodal aerial imagery. For example, IR modality captures longer thermal wavelengths and, thus, it can enable detection of objects in varying weather conditions, expanding on the capabilities of RGB [40].

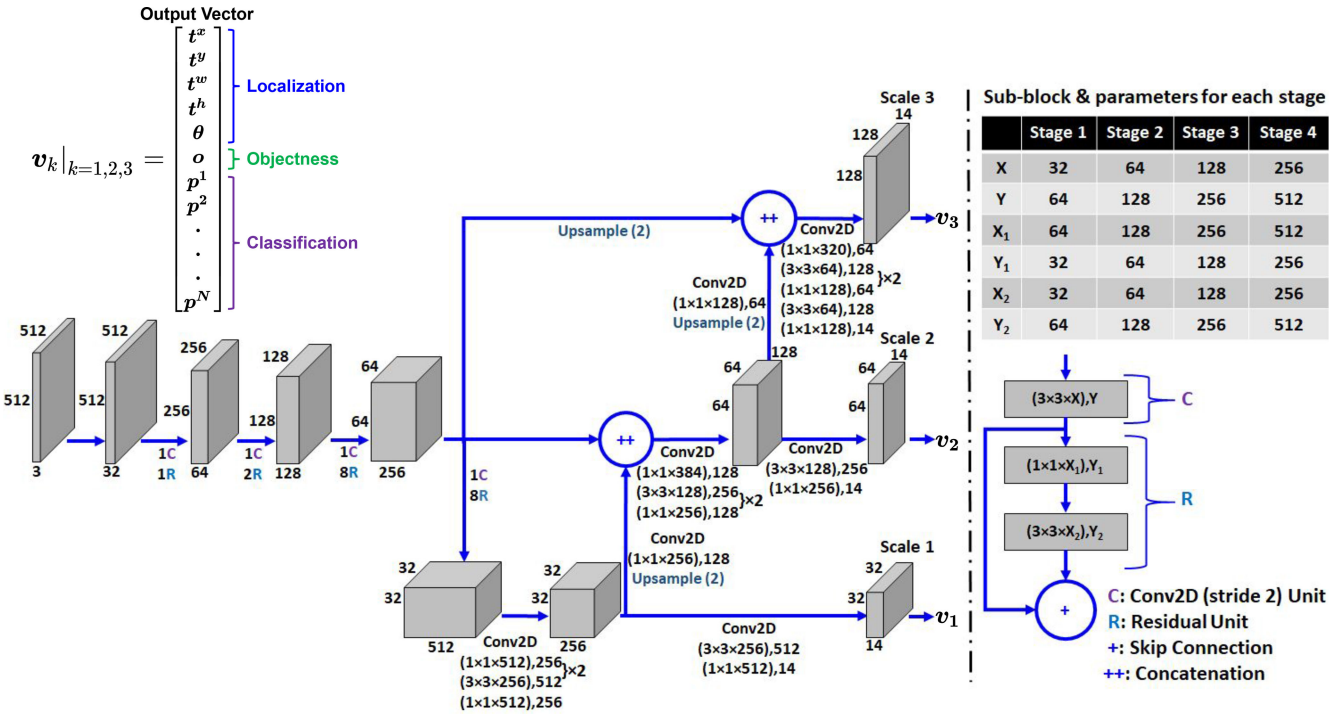
C. Challenges

The state-of-the-art object detection frameworks discussed earlier are designed and optimized for general computer vision tasks and tend to underperform in RS applications where objects cover only few pixels in the entire image and exhibit arbitrary perspective transformations. In this article, we address these challenges and introduce a real-time convolutional framework designed and optimized for object detection in multimodal RS aerial images. In addition, the proposed architecture is also capable to conduct mid-level fusion of multiple RS modalities.

III. Proposed Architecture: YOLOrs

A. Network Architecture

The proposed YOLOrs ("rs" stands for "remote sensing") framework builds upon the modular residual blocks from ResNet [41] and the multihead detection approach of YOLOv3 [18]. The network utilizes 66 *convolutional layers* along with 30 more layers that consist of the following.



	Stage 1	Stage 2	Stage 3	Stage 4
X	32	64	128	256
Y	64	128	256	512
X_1	64	128	256	512
Y_1	32	64	128	256
X_2	32	64	128	256
Y_2	64	128	256	512

Fig. 1. Schematic of the proposed YOLOrs architecture. The table on right presents the filter depths (X, X_1, X_2) along with the number of filters (Y, Y_1, Y_2) for each downsampling stage in the network. E.g., for stage (1C, 1R), 1C denotes 1 convolutional layer with 64 filters of size $(3 \times 3 \times 32)$ and 1R denotes 1 residual block with 2 convolutional layers with 32 filters of size $(1 \times 1 \times 64)$, followed by 64 filters of size $(3 \times 3 \times 32)$.

1) *Shortcut layers* that bring the output of the third-layer backward and add it to the output of the previous layer (see bottom right subblock in Fig. 1).
2) *Upsample layers* that upsample the output of the previous layer by a factor of stride using bilinear upsampling.
3) *Route layers* that return the depthwise concatenated outputs of the listed intermediate layers.
4) *YOLO layers* that correspond to the detection head layers, as shown by Scale 1, Scale 2, and Scale 3 layers in Fig. 1.

The combination of convolutional layers along with shortcut, upsample, route, and YOLO layers achieve strides of 16, 8, and 4 for the three detection heads at layers 70, 82, and 96, respectively. This architecture allows for differentiation between objects as close as four pixels apart. For an input image of dimensions 512×512, YOLOrs offers detection head granularity of 32×32, 64×64, and 128×128, for the three detection heads, respectively. The importance of detection-head granularity in resolving densely spaced objects in aerial imagery is illustrated in Fig. 2. Despite downsampling at four different stages in the intermediate feature maps by a stride of two pixels, the network achieves the aforementioned resolutions due to upsampling and concatenation (CO) of features from longer skip-connections from the semantically rich intermediate feature extraction layers. This shallower architecture offers YOLOrs double detection heads granularity with much fewer parameters (20 132 106) compared to YOLOv3 (61 561 429). The network utilizes batch normalization as the input normalizer for feature maps and leaky-ReLU as an activation function between convolutional layers.

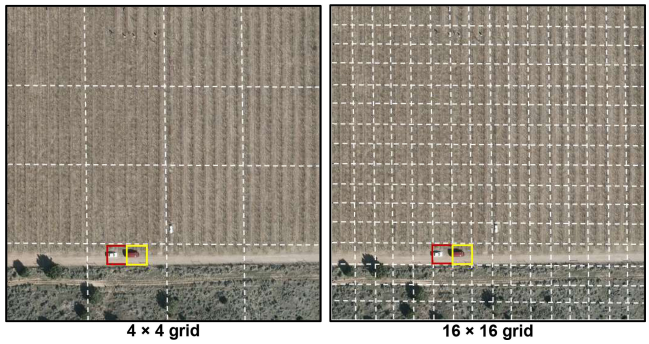


Fig. 2. Representation of objects in an aerial RGB image divided into 4×4 (left) and 16×16 (right) grids.

B. Multimodal Fusion

In YOLOrs, longer skip-connections start after the third downsampling block. This allows for enough depth (up to the 37th layer) for fusing data from multiple modalities at various stages in the feature extraction region of the network, as shown in Fig. 1. For example, if fusion occurs after the nth layer, we create two streams of the first n layers of the YOLOrs network, one for each modality, and fuse their outputs at the end of the $(n+1)$th (convolutional) layer with filter size 1×1. This last convolutional layer is used to adjust the weighted feature map depth of each modality so that after fusion, the resultant feature map meets the input requirements of the first layer of the remaining joint YOLOrs network. In this work, we consider two

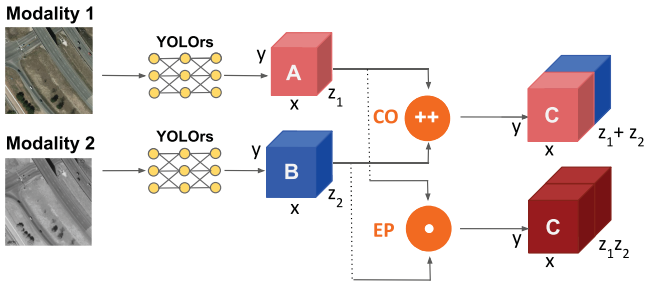


Fig. 3. Representation of multimodal fusion via CO and elementwise cross product (EP) methods.

specific fusion methods, as shown in Fig. 3. The proposed fusion methods are presented below for two modalities, for the sake of simplicity, but can straightforwardly extend to any number of modalities.

Concatenation: We start with two individual YOLOrs streams. We denote by $\boldsymbol{A}$ and $\boldsymbol{B}$ the feature maps at layer n for the first and second modality, respectively. The depth of $\boldsymbol{A}$ is z_1 and the depth of $\boldsymbol{B}$ is z_2. In this fusion approach, we concatenate $\boldsymbol{A}$ and $\boldsymbol{B}$ along the depth dimension and obtain the fused feature map $\boldsymbol{C}$, with depth $z_1 + z_2$ as

$$\boldsymbol{C}_{:,:,i} = \begin{cases} \boldsymbol{A}_{:,:,i}, & i \leq z_1 \\ \boldsymbol{B}_{:,:,i}, & i > z_1 \end{cases}. \tag{1}$$

The CO offers to the network an increased number of feature maps. Then, the network learns how to combine them (minimizing overall loss), effectively performing fusion of the modalities.

Elementwise Cross Product: In the same setup as above, we built the fused feature map $\boldsymbol{C}$ by elementwise multiplication of every feature map in $\boldsymbol{A}$ with the every feature map in $\boldsymbol{B}$. That is, for every pair of $i \in \{1, \ldots, z_1\}$ and $j \in \{1, \ldots, z_2\}$, we form

$$\boldsymbol{C}_{:,:,ij} = \boldsymbol{A}_{:,:,i} \circ \boldsymbol{B}_{:,:,j}. \tag{2}$$

The depth of the resulting fused feature map $\boldsymbol{C}$ is $z_1 z_2$. Compared to CO, EP offers the network an even larger collection of already combined feature maps, effectively enforcing fusion. Then, the network can learn how to further combine these maps in order to minimize the loss.

In addition, YOLOrs allows both CO and EP fusion to be carried out in weighted manner. For instance, for two modalities, we can set the *bias* hyperparameter to *left*, *right*, or *center* and thus, respectively, place more emphasis at the first modality, the second modality, or none of the two (equal emphasis). The proposed YOLOrs network along with multimodal fusion elements is easy to interpret and tune and it is end-to-end trainable. For example, fusion via balanced CO between feature maps for RGB and IR modalities at 38th layer is shown in Fig. 4.

C. Network Output

The proposed method partitions the entire image in a grid of cells within which we search for objects individually. The output feature map at each detection head is a 3-D tensor [42]: mode-1 corresponds to the width-index of each cell, mode-2 corresponds to the height-index of each cell, and mode-3 corresponds to the extracted features.

For the ith grid cell, YOLOrs predicts n_A bounding boxes, one for each anchor box. For each predicted bounding box, our method returns an output feature vector of the form

$$\boldsymbol{v_k} = \left[t^x \ , \ t^y \ , \ t^w \ , \ t^h \ , \ \theta \ , \ o \ , \ p^1 \ , \ \cdots \ , \ p^N \right]. \tag{3}$$

Output vector $\mathbf{v}_k$ is also shown in Fig. 1. Entries (t^x, t^y) are the center coordinates of the bounding box, relative to the top-left corner of the cell. Entries t^w and t^h are the log-transformed width and height, respectively, of the box. θ is the orientation angle of the box, relative to the positive x-axis. o is the "objectness" score, taking values from 0 to 1—high value signifies high probability of this box to contain an object. p^n, for $n = 1, 2, \ldots, N$, is the score for class n, taking values from 0 to 1—high p^n indicates high probability for the bounding box to contain an object from class n.

We consider n_A anchor boxes per detection head and, accordingly, n_A bounding boxes and n_A corresponding output vectors per grid cell. This leads to a detection head of depth $(6 + N)n_A$ for each grid cell.

Next, we want to identify the center, width, and height of any predicted bounding box. Consider bounding box corresponding to the output vector in (3) and placed within the ith grid cell. Denote the width and height indices of the cell (counting from the top-left corner of the image) by g_i^x and g_i^y, respectively. Also, denote the width and height of the anchor box by a_i^w and a_i^h, respectively. Then, for this bounding box, we calculate: the center coordinates relative to the top-left corner of the image $c^x = t^x + g_i^x$ and $c^y = t^y + g_i^y$; the width $w = a_i^w \exp(t^w)$; and the height $h = a_i^h \exp(t^h)$.

D. Loss Function

Similar to YOLO [23], we want a single bounding box to be responsible for each object detection. Therefore, we want to keep exactly one out of the n_A bounding boxes that the network predicted for each cell.

To do that, we keep the bounding box that yields the highest Intersection over Union (IoU), calculated as shown in detail in the following section. Next, we denote by $\mathbf{v}_i$ the single predicted bounding box for the ith cell and by $\mathbf{v}_i^g$ the ground-truth vector for the same cell. Based on these, we evaluate the loss of the prediction as follows:

$$L = L_{\text{reg}} + L_{\text{conf}} + L_{\text{cls}}. \tag{4}$$

In (4)

$$\begin{aligned} L_{\text{reg}} = \lambda_{xy} \sum_{i=1}^{S} \sum_{j=1}^{n_A} 1_{i,j}^{o} [(t_i^{x,g} - t_i^x)^2 + (t_i^{y,g} - t_i^y)^2] \\ \times \lambda_{wh} \sum_{i=1}^{S} \sum_{j=1}^{n_A} 1_{i,j}^{o} [(t_i^{w,g} - t_i^w)^2 + (t_i^{h,g} - t_i^h)^2] \\ \times \lambda_{\theta} \sum_{i=1}^{S} \sum_{j=1}^{n_A} 1_{i,j}^{o} [(\theta_i^g - \theta_i)^2] \end{aligned} \tag{5}$$

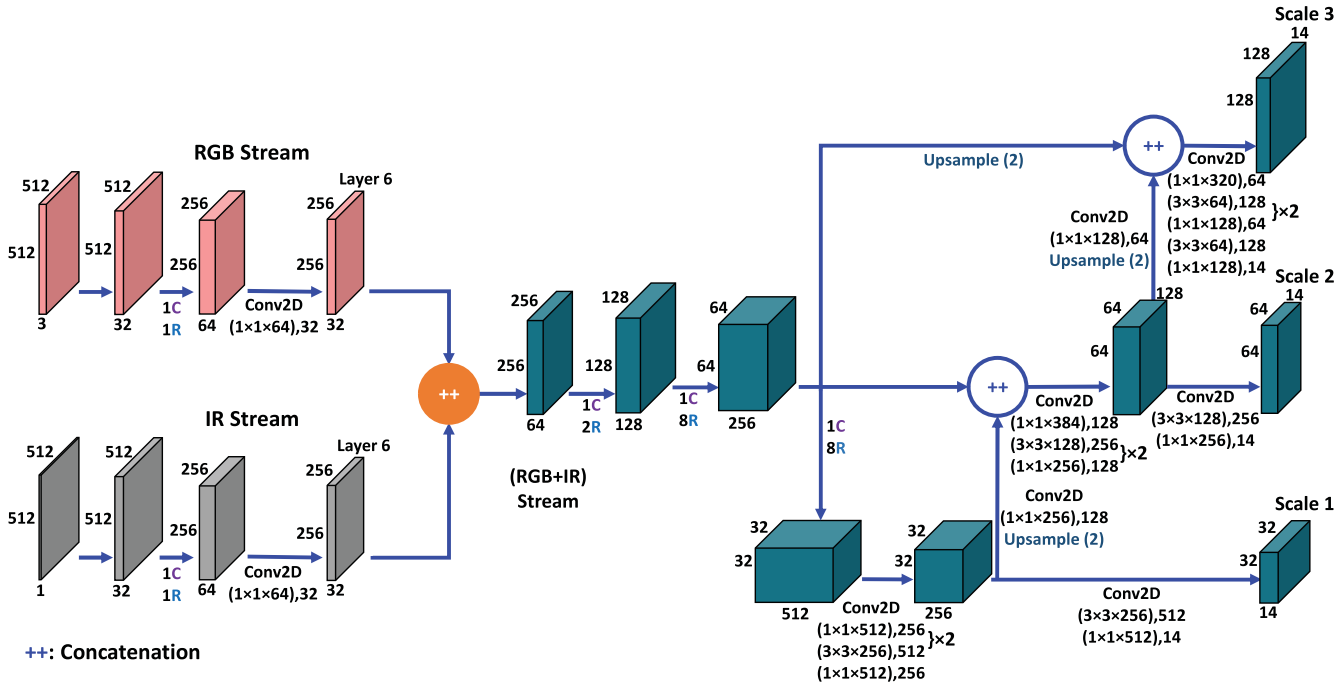


Fig. 4. Schematic of the mid-level CO fusion at sixth-layer model using YOLOrs as backbone network for RGB and IR modalities. R and C are same as in Fig. 1.

is the overall bounding-box regression loss, capturing loss due to location, width/height estimation, and orientation of the bounding box.

Moreover

$$L_{\text{conf}} = \lambda_o \sum_{i=1}^{S} \sum_{j=1}^{n_A} 1_{i,j}^{o} [-(1-o_i)^{\gamma} \log(o_i)] \times \lambda_{\text{no}} \sum_{i=1}^{S} \sum_{j=1}^{n_A} 1_{i,j}^{no} [-o_i^{\gamma} \log(1-o_i)] \tag{6}$$

is the object confidence loss (objectness and no-objectness) and

$$L_{\text{cls}} = \lambda_c \sum_{i=1}^{S} \sum_{n=1}^{N} 1_i^{o} [-\log(p_i^n)]. \tag{7}$$

is the object classification loss.

In (5)–(7), S is the total number of grid cells in the detection head. "g" in the superscripts denotes ground truth. 1_i^o is 1 if the object is present in the ith grid cell and 0 otherwise. $1_{i,j}^o$ is 1 if the jth bounding box is responsible for detection in the ith grid cell and 0 otherwise. Similarly, $1_{i,j}^{\text{no}}$ is 1 if the jth bounding box in the ith grid cell does not correspond to any detection and 0 otherwise. γ is the focal loss parameter [43] ($\gamma = 0$ corresponds to binary cross entropy loss). The weights λ_{xy}, λ_{wh}, λ_θ, λ_o, λ_{no}, and λ_c regulate error emphasis between box coordinates, box dimensions, box orientation, "objectness," "no-objectness," and classification. These weights can be set ad hoc. In this work, we consider $\lambda_{xy} = \lambda_{wh} = \lambda_o = \lambda_c =$ and $\lambda_{\text{no}} = 10$, to penalize false-positives since we expect in RS images low object to background ratio.

E. Intersection Over Union

To compute the IoU of a predicted bounding box in the ith grid cell with the corresponding ground-truth, we work as follows. First, we center the box at origin and compute the four vertices $\{(x_j, y_j)\}_{j=1,\ldots,4}$, using w and h. We repeat the same for the ground truth box. Next, we rotate the predicted and ground truth boxes by the predicted θ and ground truth θ_g, respectively. The rotations are counterclockwise with respect to the positive x-axis as

$$\begin{bmatrix} x'_j \\ y'_j \end{bmatrix} = \mathbf{R}(\phi) \begin{bmatrix} x_j \\ y_j \end{bmatrix}, \; j = 1, 2, 3, 4 \tag{8}$$

where

$$\mathbf{R}(\phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{9}$$

is the 2×2 rotation matrix for angle ϕ. Next, the rotated predicted bounding box is translated to its respective central co-ordinate (c_i^x, c_i^y) as

$$\begin{bmatrix} x''_j \\ y''_j \end{bmatrix} = \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} + \begin{bmatrix} c_i^x \\ c_i^y \end{bmatrix}, \; j = 1, 2, 3, 4. \tag{10}$$

We note that the region of intersection of two nonoriented bounding boxes is rectangular, leading to a simple IoU computation. However, in the case of oriented bounding boxes, the intersection area may not be rectangular, but a polygon with as many as eight sides. An example of intersection of two oriented bounding boxes ABCD and PQRS is shown in Fig. 6. The IoU is computed as the ratio of the intersection area PQKCM and

Fig. 5. Data augmentation using jumble-up technique to create 2×2 image grid of rectangular cells from original input image.

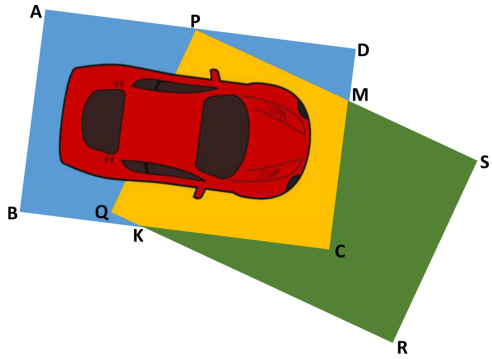


Fig. 6. Representation of the area of intersection and area of union of two rotated bounding boxes. ABCD and PQRS are the ground truth and predicted bounding boxes, respectively. The intersection area PQKCM is in yellow.

the union area ABKRSMD

$$\text{IoU} = \frac{\texttt{area}(\text{PQKCM})}{\texttt{area}(\text{ABKRSMD})}. \tag{11}$$

F. Data Augmentation

In view of the limited availability of labeled RS training datasets, to increase the robustness of the YOLOrs network against the variability in the input images, we employ the following data augmentation techniques: geometric augmentations, such as flipping (horizontal, vertical, and their combination) and rotation (90°, 180°, and 270°); photometric augmentations by altering gamma, brightness, contrast, and grayscale properties of an image. In addition, inspired by jigsaw puzzles [44], we introduce "jumble-up": a new geometric data augmentation method by which we split an image along any dimension into two arbitrary rectangular parts and swap them with each other. When this process is carried out horizontally and vertically, it leads to modified version of the input image as shown in Fig. 5. While splitting the image, we make sure that the split *does not* intersect with an object.

IV. Experimental Studies

In this section, we present the performance of the proposed YOLOrs architecture on the VEhicle Detection in Aerial Imagery (VEDAI) dataset [46]. The performance of YOLOrs is compared with that of unimodal YOLOv4 [26], EfficientDet (D0) [47], RetinaNet (with backbone ResNet 50) [43], and YOLOv3 [18] trained on RGB and IR images as well as their multimodal versions trained on concatenated RGB and IR images [45], as shown in Fig. 7. In addition, we present the performance of the proposed YOLOrs model on unimodal RGB, unimodal IR, and mid-level fusion of RGB and IR modalities. Before we proceed to the experimental results, we provide a brief overview of the VEDAI dataset.

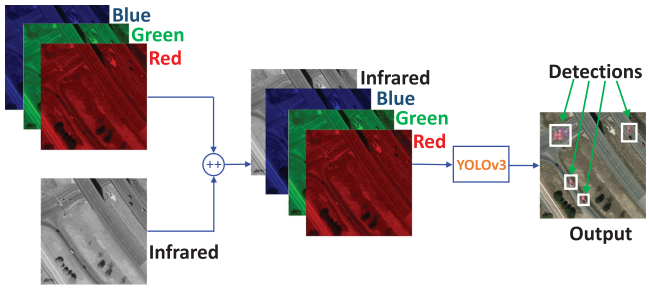


Fig. 7. YOLOv3 with early fusion [45].

TABLE I
Distribution of Available Class Instances in the VEDAI Dataset Across Ten Folds

Classes	Total Instances	Distribution Across 10 Folds
Car	1349	9 folds of 135; one fold of 134
Pickup	941	9 folds of 94; one fold of 95
Camping car	390	10 folds of 39
Truck	300	10 folds of 30
Other	200	10 folds of 20
Tractor	190	10 folds of 19
Boat	170	10 folds of 17
Van	100	10 folds of 10

A. VEDAI Dataset [46]

The VEDAI dataset contains cropped images that are taken from the much larger Utah Automated Geographic Reference Center (AGRC) dataset [48]. The size of each image in AGRC is about $16\,000 \times 16\,000$ pixels, with a resolution of about 12.5×12.5 cm per pixel. All images were captured from the same altitude. Each image is available in two modalities: RGB and IR. The images in both modalities capture the same scene and are registered with each other. VEDAI dataset includes 1246 smaller images, cropped from ARGC, in two resolutions, 1024×1024 and 512×512. These 1246 images were selected such that they include varying background, including grass, highway, mountains, and urban areas, among others. The VEDAI dataset contains 11 classes of vehicles. In this work, we operate on the 512×512 images of eight vehicle classes. The available instances per class are divided in ten folds, as shown in Table I. We do not consider classes with fewer than 50 instances in the dataset, such as plane, motorcycle, and bus. Every image is annotated and, for each object in the image, the annotations contain the co-ordinates of the center of the bounding box, the orientation of the object with respect to the positive x-axis, the four corners of the bounding box, the class ID, a binary flag that

identifies if an object is occluded, and another binary flag that identifies if an object is cropped.

B. Data Preprocessing

We convert the annotations of the VEDAI dataset to YOLOv3's Darknet format. First, we express the class ID for the 8 classes of interest 0, 1, ...,7. Then, we compute the normalized center co-ordinates of the bounding box ($x_{\text{center}}/512$, $y_{\text{center}}/512$. The normalized ground-truth width w^g and height h^g are obtained by first aligning the bounding boxes with the x-axis (to capture the exact width and height of the rotated object) to obtain the new corner co-ordinates $x_{g1}{}', y'_{g1}; x_{g2}{}', y'_{g2}; x_{g3}{}', y'_{g3}; x_{g4}{}', y'_{g4}$ using (8), with the transposed 2×2 rotation matrix, $\mathbf{R}^\top(\theta_g)$. Next, we compute $w_g = (\texttt{max}(x_{g1}{}', x_{g2}{}', x_{g3}{}', x'_{g4}) - \texttt{min}(x_{g1}{}', x_{g2}{}', x_{g3}{}', x'_{g4}))/512$ and $h_g = (\texttt{max}(y_{g1}{}', y_{g2}{}', y_{g3}{}', y'_{g4}) - \texttt{min}(y_{g1}{}', y_{g2}{}', y_{g3}{}', {}'y'_{g4}))/512$. Finally, we have the orientation θ_g of each object, scaled from $[-\pi, +\pi]$ to $[0, +\pi)$.

C. Results

We conduct performance evaluation by means of standard ten-fold cross-validation. Specifically, we define 10 train-test data splits (90:10) (the exact same data splits as in the VEDAI paper [46]). In each split, 1089 images are used for training and 121 images are used for testing, making sure that we never test on training data. By means of this process, we evaluate the average model performance for a range of hyperparameters and tune the model. Accordingly, we train all YOLOrs models for a total of 250 epochs, with a minibatch size of five images with gradient accumulation interval equal to two minibatch iterations. We employ the Adam optimizer [49] with a base learning rate of 10^{-3} as well as a weight decay parameter equal to 10^{-3} and use reduce on plateau scheme as the learning rate scheduler to decrease the learning rate by a factor of 0.1, if the difference in testing loss does not reduce by a certain threshold for a patience interval of 15 epochs. We evaluate the performance of the trained models on the testing data, at any training epoch, by computing the mean Average Precision (mAP), which is defined as $\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} P_{\text{avg}}(i)$, where $N = 8$ and $P_{\text{avg}}(i)$ is the average precision of class i computed as the area under the precision-recall curve [50]. We compute the final mAP scores averaged over ten cross-validation folds, by setting IoU threshold to 0.2, nonmaximum suppression (NMS) threshold to 0.1, and confidence threshold 0.7.

1) Number of Anchors: To demonstrate the effect of the number of anchors per detection head on the performance of YOLOrs, we perform a study by varying the number of anchors per detection head $nA \in \{1, 3\}$ for focal loss parameter $\gamma \in \{0, 3\}$ for unimodal RGB configuration. As per (6), $\gamma = 0$ means that we do not use focal loss and instead, use the standard binary cross-entropy loss; $\gamma = 3$ means we use focal loss instead of binary cross-entropy loss, with the focal loss parameter $\gamma = 3$.

We plot the resulting mAP versus training epoch index in Fig. 8 and observe that $nA = 3$ and $\gamma = 3$ demonstrate the best performance at lower epochs. However, the model with $nA = 1$ and $\gamma = 3$ catches up at higher epochs. We note that the model

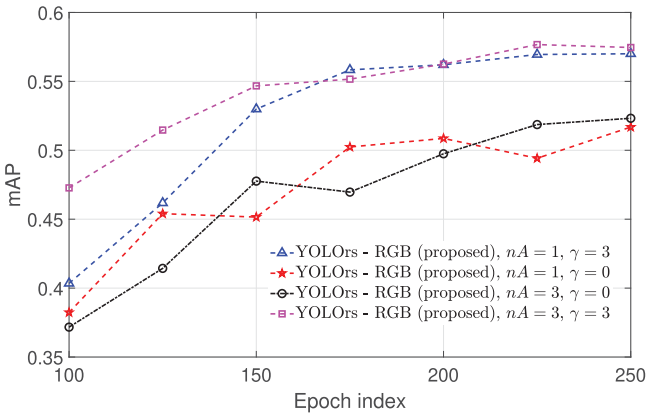


Fig. 8. Effect of the number of anchors $n_A \in \{1, 3\}$ per detection head for focal loss parameter $\gamma \in \{0, 3\}$ for proposed unimodal YOLOrs RGB model. The figure shows mAP on testing data versus training epoch index. IoU threshold, NMS threshold, and confidence threshold values are set to 0.2, 0.1, and 0.7, respectively.

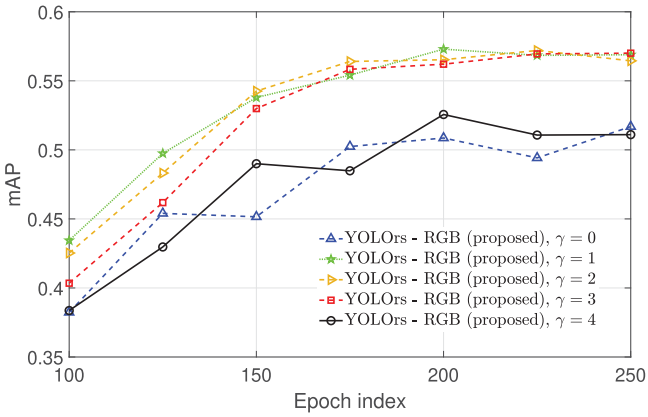


Fig. 9. Effect of the focal loss parameter $\gamma \in \{0, 1, 2, 3, 4\}$ for $n_A = 1$ anchors per detection head for the proposed YOLOrs on RGB. The figure shows mAP on testing data versus training epoch index. Threshold values are the same as in Fig. 8.

with $nA = 3$ takes considerably longer to train and test because of the use of three anchors per detection head, compared to the model with $nA = 1$, which uses only one anchor per detection head. Moreover, $nA = 1$ performs at least as high as $nA = 3$, making it a more efficient option.

2) Focal Loss: In addition, in Fig. 8, we observe that $\gamma = 3$ with both $nA = 1$ and $nA = 3$ improves the mAP performance significantly. We understand that this is because focal loss penalizes the incorrectly detected parts of the image much more compared to the correctly detected ones. Also, in Fig. 9, we performed study on other focal loss parameter γ values keeping $nA = 1$ and observed that $\gamma = 1, 2$, and 3 provide better performance than other values. Deeper insight on the classwise performance revealed that $\gamma = 3$ provides better balanced performance across all classes than $\gamma = 1$ and 2.

Overall, the aforementioned study justify that one anchor per detection head and focal loss parameter $\gamma = 3$ are preferred options.

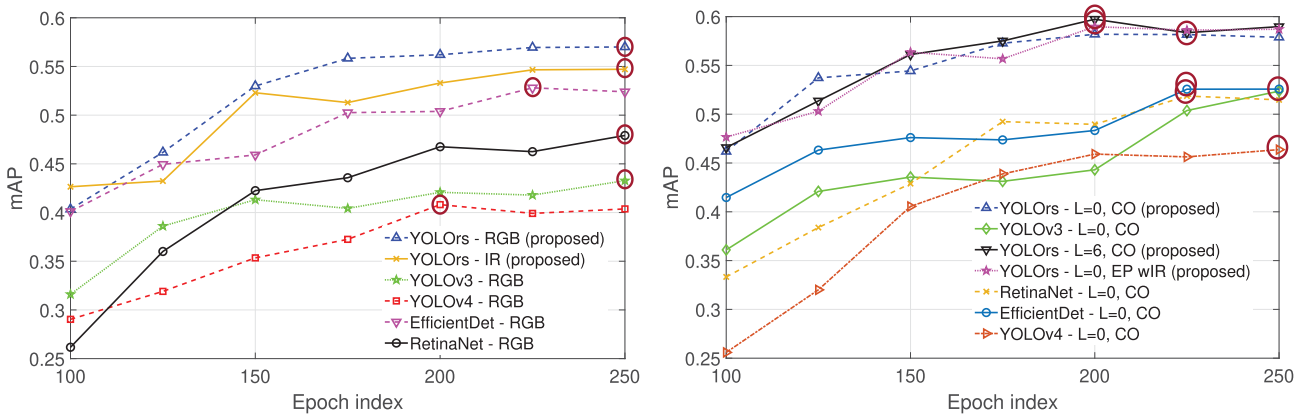


Fig. 10. mAP on testing data versus training epoch index. Threshold values are same as in Fig. 8. We use a maroon circle to mark the top performance of each method across all epochs. (a) Unimodal approaches. (b) Multimodal approaches.

TABLE II
STUDY OF THE INFLUENCE OF DIFFERENT DATA AUGMENTATION METHODS FOR YOLORS

No Aug	Flips	Rotate	Photometric	Jumble-up	mAP
✓					0.3502
	✓				0.4796
		✓			0.5153
			✓		0.3412
				✓	**0.5398**
	✓	✓	✓	✓	0.5700

The mAP score in bold highlights the best data augmentation technique in isolation.

3) Data Augmentation: We also study the influence of different data augmentation methods with proposed YOLOrs, as shown in Table II. The proposed jumble-up augmentation technique provides a significant boost in mAP scores over standard geometric and photometric augmentations. We infer that the jumble-up data augmentation motivates the network to learn more meaningful semantic features than the standard augmentation techniques. The combination of geometric, photometric, and jumble-up augmentation yields the best mAP scores.

To further evaluate YOLOrs configurations against contemporary methods, we use the same set of input hyperparameters as listed for YOLOrs, except for the initial learning rate and weight decay parameter of the optimizer. We also use the same set of augmentation techniques for all the networks and train them from scratch to allow for a fair comparison.

4) Unimodal Data: In Fig. 10(a), we plot the mAP values of unimodal approaches and observe that the proposed YOLOrs unimodal RGB model outperforms the corresponding unimodal IR model. Moreover, both the proposed YOLOrs unimodal RGB and IR models perform better than the unimodal RGB model configurations of YOLOv3 [18], RetinaNet [43], EfficientDet (D0) [47], and YOLOv4 [26].

5) Multimodal Data: To evaluate the performance of the proposed mid-level fusion, we fuse RGB and IR after layer $L = 6$, using feature CO and EP across feature maps, as explained in Section III-B. In Fig. 10(b), we plot the mAP values of multimodal approaches versus training epoch index and observe that the proposed YOLOrs outperforms YOLOv3 [45], RetinaNet, EfficientDet (D0) [47], and YOLOv4 [26] with early fusion approaches.

In Fig. 10(b), we notice that the performance of the proposed YOLOrs early fusion ($L = 0$) and mid-level fusion ($L = 6$) perform better than the unimodal YOLOrs RGB and IR models in Fig. 10(a), demonstrating a clear benefit of multimodal data fusion in aerial object detection. Moreover, we observe that YOLOrs for ($L = 6$, CO) demonstrates the best performance at larger training epochs. At the 250th epoch, the next best performing model is YOLOrs for [$L = 6$, EP (wIR)], i.e., EP fusion using 16 and 4 feature maps from IR and RGB streams, respectively. Early fusion ($L = 0$, CO) for the proposed YOLOrs model outperforms the corresponding RetinaNet, EfficientDet (D0), and YOLOv4 models.

Among the proposed YOLOrs fusion techniques, fusion using ($L = 6$, CO), performs better, arguably because the features from each individual modality are learned independently of the other, before fusing them using feature CO. Interestingly, we observe that the [$L = 6$, EP (wIR)] configuration performs better than the ($L = 6$, CO) configuration at smaller training epoch values (<110) and we understand this is because EP fusion method allows the model to learn cross-modality interactions, thereby unveiling richer semantic information in fewer training epochs. However, if the model is trained further, the simple CO fusion method allows the network the freedom to weigh the fused feature map, containing the same number of unaltered feature maps from individual modalities, in a way that leads to superior performance.

In summary, we observe that fusion methods perform better than the corresponding individual unimodal methods and we understand that this is because the fusion models are able to leverage any complimentary information available in different modalities, by creating fused feature maps that contain richer latent information.

TABLE III
CLASSWISE AVERAGE PRECISION (BEST ACROSS ALL EPOCHS) FOR PROPOSED YOLORS, YOLOV4, RETINANET (BACKBONE: RESNET 50), AND EFFICIENTDET (D0), UNIMODAL AND MULTIMODAL CONFIGURATIONS CORRESPONDING TO EPOCH INDEX OF THE BEST ACHIEVED MAP SCORE BY EACH MODEL

	Proposed YOLOrs					**YOLOv4**			**RetinaNet (50)**			**EfficientDet (D0)**		
	Unimodal		**Multimodal**			**Unimodal**		**Multimodal**	**Unimodal**		**Multimodal**	**Unimodal**		**Multimodal**
Class	**IR**	**RGB**	**L=0, CO**	**L=6, CO**	**L=6, EP (wIR)**	**IR**	**RGB**	**L=0, CO**	**IR**	**RGB**	**L=0, CO**	**IR**	**RGB**	**L=0, CO**
Car	0.8203	**0.8525**	0.8303	**0.8415**	0.8348	0.5650	0.6727	0.7450	0.5966	0.7087	0.7127	0.6723	0.7026	0.7296
Pickup	0.7392	0.7293	0.7651	**0.7827**	**0.7696**	0.4725	0.6029	0.6849	0.5409	0.6298	0.6383	0.6302	0.6715	0.6936
Camping car	0.6380	**0.7031**	0.6778	**0.6881**	0.6569	0.4473	0.6618	0.6239	0.5338	0.5936	0.6126	0.5951	0.6846	0.6214
Truck	0.5421	0.5065	0.5291	0.5260	0.5351	0.3355	0.4939	0.5272	0.5281	0.5474	**0.6238**	0.4960	0.6055	**0.6223**
Other	0.4399	0.4267	**0.4748**	**0.4675**	0.4388	0.1451	0.2690	0.2884	0.2489	0.1978	0.2610	0.2450	0.3865	0.3485
Tractor	0.5439	**0.7677**	**0.7048**	0.6788	0.6907	0.2658	0.3127	0.4203	0.3530	0.5047	0.5968	0.4844	0.6391	0.6275
Boat	**0.2197**	0.1865	0.2067	0.2147	**0.2228**	0.0205	0.0329	0.0578	0.0957	0.1766	0.1577	0.0741	0.1234	0.0867
Van	0.4338	0.3892	0.4642	**0.5791**	**0.5688**	0.1665	0.2190	0.1985	0.4229	0.4745	0.5461	0.4452	0.4113	0.4761
Mean	0.5471	0.5700	0.5816	**0.5973**	**0.5897**	0.3023	0.4081	0.4432	0.4150	0.4791	0.5186	0.4553	0.5281	0.5257

Note: The top two performances per class are in bold.

TABLE IV
PRECISION AND RECALL FOR PROPOSED YOLORS MODELS CORRESPONDING TO EPOCH INDEX OF THE BEST ACHIEVED MAP SCORE BY EACH MODEL

Model	**Precision**	**Recall**
IR	0.4162	0.7240
RGB	0.4664	0.7426
L=0, CO	0.4177	0.7577
L=6, CO	0.4138	07656
L=6, EP (wIR)	0.4184	0.7535

TABLE V
MAP PERFORMANCE OF UNIMODAL AND MULTIMODAL YOLORS FOR DIFFERENT COMBINATIONS OF THRESHOLD VALUES

Thresholds			**mAP**	
Confidence	**NMS**	**IoU**	**YOLOrs - RGB**	**YOLOrs - L=6, CO**
0.7	0.1	0.2	0.5728	0.5986
0.7	0.1	0.3	0.4976	0.5288
0.7	0.1	0.5	0.4084	0.4309
0.7	0.3	0.2	0.5595	0.5904
0.7	0.3	0.3	0.4971	0.5258
0.7	0.3	0.5	0.4067	0.4318
0.7	0.5	0.2	0.5420	0.5686
0.7	0.5	0.3	0.4847	0.5125
0.7	0.5	0.5	0.4009	0.4228
0.85	0.1	0.2	0.3548	0.4055
0.85	0.1	0.3	0.3132	0.3589
0.85	0.1	0.5	0.2607	0.3042
0.85	0.3	0.2	0.3544	0.4056
0.85	0.3	0.3	0.3126	0.3585
0.85	0.3	0.5	0.2607	0.3039
0.85	0.5	0.2	0.3520	0.4030
0.85	0.5	0.3	0.3110	0.3574
0.85	0.5	0.5	0.2596	0.3023

6) Classwise Performance: In Table III, we compare the classwise results of the proposed YOLOrs models with YOLOv4, RetinaNet, and EfficientDet (D0) models. The performances of the proposed multimodal YOLOrs methods are among the two best for every class (except for the truck). Also, the mid-fusion YOLOrs—($L = 6$, CO) and ($L = 6$, EP wIR)—attain the top two mean performances. Also, we notice that, overall, the mAP scores of both unimodal and multimodal YOLOrs are significantly higher than that of YOLOv4, RetinaNet, and EfficientDet (D0) for most of the classes. Furthermore, we notice that we achieve top performance for the classes "car" and "pickup," for which we also have the most training instances (see Table I).

7) Precision and Recall: In Table IV, we compare the performance of the proposed YOLOrs models in terms of precision and recall. We notice that the multimodal approaches have increased recall, which contributes to their superior mAP performance shown in the studies above.

8) mAP Versus Thresholds: In Table V, we compare the performances of the best unimodal (RGB) and multimodal ($L = 6$, CO) configurations of YOLOrs for confidence threshold in $\{0.7, 0.85\}$, NMS threshold in $\{0.1, 0.3, 0.5\}$, and IoU threshold in $\{0.2, 0.3, 0.5\}$. We notice that we achieve top performance for confidence 0.7, NMS 0.1, and IoU 0.2. Regarding the confidence threshold, lower values would increase undesired background detections, whereas higher values could lead to more false negatives. Regarding the NMS threshold, we notice that, due to minimal object overlap, different detection boxes generally capture the same object. A lower NMS threshold reduces the number of detections per ground truth object, leading to improved performance. Regarding the IoU threshold, we notice that low values allow us to consider positive detections even when there is some rotation misalignment between the bounding and predicted boxes. Also, we notice that multimodal YOLOrs outperforms unimodal YOLOrs for every threshold configuration.

D. Detection Results

In Fig. 11, we show the detection results obtained by the proposed unimodal YOLOrs RGB and the mid-fusion ($L = 6$, CO) configurations. The first column corresponds to the input image, whereas the second and third columns display the detection results obtained by the proposed unimodal RGB YOLOrs

Fig. 11. Object detection using various networks. Column 1 shows the input images, columns 2 and 3 shows detections obtained by the proposed YOLOrs network in baseline RGB and fusion at layer 6 via CO configurations, respectively. Black bounding boxes and labels show the ground truth, whereas colored bounding boxes and labels show the detected objects.

method and the proposed mid-fusion ($L = 6$, CO) configurations, respectively. Black bounding boxes along with black labels represent the ground truth, whereas colored bounding boxes and labels represent detections. In the examples of rows 1 and 4, unimodal and mid-fusion YOLOrs perform similarly. In example of the second row, mid-fusion method outperforms unimodal YOLOrs, detecting correctly the car. Also, in the example of the third row, the mid-fusion method again outperforms unimodal YOLOrs avoiding the two false-positive car detections.

V. Conclusion

We proposed YOLOrs: a fully convolutional single-shot network, designed and optimized for real-time object detection

in RS aerial imagery. YOLOrs has the capability to leverage data from multiple sensing modalities through a novel mid-level fusion scheme. Our extensive experimental studies corroborate the effectiveness of YOLOrs, which is shown to outperform state-of-the-art alternatives.

REFERENCES

[1] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Bombay, India, Jun. 1998, pp. 555–562.

[2] A. d'Acremont, R. Fablet, A. Baussard, and G. Quin, "CNN-based target recognition and identification for infrared imaging in defense systems," *J. Sens.*, vol. 19, no. 9, pp. 1–16, 2019.

[3] D. K. Prasad, "Survey of the problem of object detection in real images," *Int. J. Image Process.*, vol. 6, no. 6, pp. 441–466, 2012.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[5] P. Viola and M. Jones, "Robust real-time object detection," *Int. J. Comput. Vis.*, vol. 57, pp. 137–154, 2001.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, vol. 1, pp. 886–893.

[7] L. Liu *et al.*, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, 2020.

[8] Z.-Q. Zhao, P. Zheng, S.-T. 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.

[9] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: A survey," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 84–100, Jan. 2018.

[10] A. M. N. Taufique, B. Minnehan, and A. Savakis, "Benchmarking deep trackers on aerial videos," *Sensors*, vol. 20, no. 2, 2020, Art. no. 547.

[11] K. Li, G. Cheng, S. Bu, and X. You, "Rotation-insensitive and context-augmented object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2337–2348, Apr. 2018.

[12] T.-Y. 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.*, Honolulu, HI, USA, Jun. 2017, pp. 2117–2125.

[13] M. Ju, H. Luo, Z. Wang, B. Hui, and Z. Chang, "The application of improved YOLO V3 in multi-scale target detection," *J. Appl. Sci.*, vol. 9, no. 18, pp. 3774–3375, Jan. 2019.

[14] Y. Ren, C. Zhu, and S. Xiao, "Small object detection in optical remote sensing images via modified Faster R-CNN," *J. Appl. Sci.*, vol. 8, no. 5, pp. 813:1–813:11, 2018.

[15] M. Mandal, M. Shah, P. Meena, S. Devi, and S. K. Vipparthi, "AVDNet: A small-sized vehicle detection network for aerial visual data," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 3, pp. 494–498, Mar. 2020.

[16] M. Mandal, M. Shah, P. Meena, and S. K. Vipparthi, "SSSDET: Simple short and shallow network for resource efficient vehicle detection in aerial scenes," in *Proc. IEEE Int. Conf. Image Process.*, Taipei, Taiwan, Sep. 2019, pp. 3098–3102.

[17] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 3–22, 2018.

[18] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, *arXiv:1804.02767*.

[19] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[20] 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.*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[21] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Boston, MA, USA, Jun. 2015, pp. 1440–1448.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 91–99.

[23] 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.*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[24] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jun. 2017, pp. 7263–7271.

[25] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.

[26] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," Apr. 2020, *arXiv:2004.10934*.

[27] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[28] R. Nosaka, H. Ujiie, and T. Kurokawa, "Orientation-aware regression for oriented bounding box estimation," in *Proc. IEEE Int. Conf. Adv. Video Sig. Based Surveill.*, Auckland, New Zealand, Nov. 2018, pp. 1–6.

[29] Z. Liu, J. Hu, L. Weng, and Y. Yang, "Rotated region based CNN for ship detection," in *Proc. IEEE Int. Conf. Image Process.*, Beijing, China, Sep. 2017, pp. 900–904.

[30] J. Lei, C. Gao, J. Hu, C. Gao, and N. Sang, "Orientation adaptive YOLOv3 for object detection in remote sensing images," in *Proc. Chin. Conf. Pattern Recognit. Comput. Vis.*, Xi'an, China, Nov. 2019, pp. 586–597.

[31] J. O. d. Terrail and F. Jurie, "Faster RER-CNN: Application to the detection of vehicles in aerial images," Sep. 2020, *arXiv:1809.07628v2*.

[32] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, Sep. 2018, pp. 734–750.

[33] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Seoul, South Korea, Oct./Nov. 2019, pp. 9627–9636.

[34] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, "A review of deep learning methods and applications for unmanned aerial vehicles," *J. Sensor*, vol. 2017, Aug. 2017, Art. no. 3296874.

[35] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[36] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: An overview of methods, challenges, and prospects," *Proc. IEEE*, vol. 103, no. 9, pp. 1449–1477, Sep. 2015.

[37] Y. Zheng and E. Blasch, "Multispectral image fusion for vehicle identification and threat analysis," in *Proc. SPIE Commerical Sci. Sens. Imag.*, Baltimore, MD, USA, 2016, Art. no. 98710G.

[38] R. Niu, P. Zulch, M. Distasio, E. Blasch, D. Shen, and G. Chen, "Joint sparsity based heterogeneous data-level fusion for target detection and estimation," in *Proc. SPIE Defense Secur.*, Anaheim, CA, USA, May 2017, Art. no. 101960E.

[39] D. Yang, X. Liu, H. He, and Y. Li, "Air-to-ground multimodal object detection algorithm based on feature association learning," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 3, pp. 1–9, 2019.

[40] M. Dhanaraj *et al.*, "Vehicle detection from multi-modal aerial imagery using YOLOv3 with mid-level fusion," in *Proc. SPIE Defense Commercial Sens.*, Anaheim, CA, USA, May 2020, pp. 1139506:1–1139506:11.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[42] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 2980–2988.

[44] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 69–84.

[45] V. Knyaz, "Multimodal data fusion for object recognition," in *Proc. SPIE Multimodal Sens.: Technol. Appl.*, Munich, Germany, Jun. 2019, vol. 11059, pp. 110590P–1:110590P–12.

[46] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *J. Vis. Commun. Image Representation*, vol. 34, pp. 187–203, 2016.

[47] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10781–10790.

[48] Utah AGRC, 2012, Accessed: Jun. 9, 2020. [Online]. Available: https://gis.utah.gov/s

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Jan. 2017, *arXiv:1412.6980*.

[50] W. Su, Y. Yuan, and M. Zhu, "A relationship between the average precision and the area under the ROC curve," in *Proc. Int. Conf. Theory Inf. Retrieval*, Northampton, MA, USA, Sep. 2015, pp. 349–352.

Manish Sharma (Student Member, IEEE) was born in India, in 1994. He received the B.Sc. degree in physics from the University of Delhi, New Delhi, India, in 2014, and the M.Sc. degree in physics from the Indian Institute of Technology Delhi, New Delhi, India, in 2016. He is currently working toward the Ph.D. degree in imaging science with the Rochester Institute of Technology, Rochester, NY, USA.

His research interests include computer vision, deep learning, tensor processing for neural networks, and multimodal data fusion.

Mayur Dhanaraj (Student Member, IEEE) was born in Bengaluru, India, in 1994. He received the B.S. degree in electronics and communications engineering from the Bangalore Institute of Technology, Bengaluru, India (affiliated to Visvesvaraya Technological University, Belgaum, India), in 2016, and the M.S. degree in electrical and microelectronic engineering in 2018 from the Rochester Institute of Technology (RIT), Rochester, NY, USA, where he is currently working toward the Ph.D. degree in algorithms for reliable and dynamic tensor decomposition and CNN compression via tensor processing at the Kate Gleason College of Engineering.

He is currently a Research Assistant at the Kate Gleason College of Engineering, RIT. His research interests include signal processing, data analysis, and machine learning.

Srivallabha Karnam was born in Karnataka, India, in 1990. He received the Diploma degree (three years) from the Department of Technical Education, Govt. of Karnataka, India, in 2006, the bachelor's degree in electrical and electronics engineering from Visvesvaraya Technological University, in 2012, and the M.Sc. degree in computer engineering from the Rochester Institute of Technology (RIT), Rochester, NY, USA, in 2020, with a specialization in machine learning and computer vision.

After his bachelor's degree, he was with the semiconductor industry for six years as an R&D Engineer and Application Engineer working on embedded product development, power electronics, and high-speed serial standards. He has been recognized as a Computer Engineering Graduate Delegate for the class of 2020 with RIT. His research interests include machine learning and computer vision with an emphasis on self-supervised learning, network compression for edge devices, pattern recognition, audio processing, and 3-D point processing (unstructured data).

Dimitris G. Chachlakis (Member, IEEE) was born in Athens, Greece, in 1991. He received the Diploma (five-year program) degree in electronic and computer engineering from the Technical University of Crete, Chania, Greece, in 2016. He is currently working toward the Ph.D. degree in engineering with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY, USA.

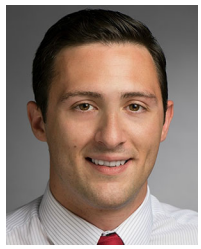
He is currently a Research Assistant with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, where he conducts research in the areas of machine learning, signal processing, and data analysis, with a focus on robustness and tensor methods.

Mr. Chachlakis is a Graduate Student Member of the Society for Industrial and Applied Mathematics. He has served as a Reviewer for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE SIGNAL PROCESSING LETTERS, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE ACCESS, IEEE PHOTONICS JOURNAL, and Elsevier *Digital Signal Processing*. He was a recipient of the 2018 Gerondelis Foundation Graduate Student Scholarship Award and the 2019 A. G. Leventis Foundation Graduate Student Scholarship Award.

Raymond Ptucha (Senior Member, IEEE) received the B.S. degree in computer science and the B.S. degree in electrical engineering from the University at Buffalo, Buffalo, NY, USA, in 1988 and 1989, respectively, the M.S. degree in image science from Rochester Institute of Technology (RIT), Rochester, NY, USA, in 2002, and the Ph.D. degree in computer science from RIT in 2013.

He is currently a Computational Display Technology Leader with the Visual Experience Group, Apple, Cupertino, CS, USA, where he is responsible for machine learning and algorithms in display products. He was an Associate Professor in computer engineering and the Director of the Machine Intelligence Laboratory, RIT, where he coauthored more than 100 publications, including topics in machine learning, computer vision, and robotics, with a specialization in deep learning. Prior to RIT, he was a Research Scientist with Eastman Kodak Company where he worked on computational imaging algorithms and was awarded 33 U.S. Patents.

Dr. Ptucha was the recipient of an NSF Graduate Research Fellowship, in 2010 and his Ph.D. research earned the 2014 Best RIT Doctoral Dissertation Award. He is a passionate supporter of STEM education, an NVIDIA certified Deep Learning Institute Instructor, the Chair of the Rochester Area IEEE Signal Processing Society, and an active member of his local IEEE Chapter and FIRST robotics organizations.



Panos P. Markopoulos (Member, IEEE) was born in Athens, Greece, in 1986. He received the Diploma (five-year program) and M.S. degrees in electronic and computer engineering from the Technical University of Crete, Chania, Greece, in 2010 and 2012, respectively, and the Ph.D. degree in electrical engineering from The State University of New York at Buffalo, Buffalo, NY, USA, in 2015.

Since August 2015, he has been an Assistant Professor of electrical engineering with the Rochester Institute of Technology (RIT), Rochester, NY, USA, where he has established and directs the Machine Learning Optimization and Signal Processing Laboratory. He is also a core faculty member of the RIT Center for Human-Aware Artificial Intelligence. In 2018 and 2020, he was a Summer Visiting Research Faculty with the U.S. Air Force Research Laboratory, Information Directorate, in Rome, NY, USA. He has coauthored more than 65 journal and conference articles in his research interests, which include statistical signal processing, machine learning, data analysis, and optimization, with a current focus on tensor methods, robustness, Lp-norm formulations, and dynamic learning.

Dr. Markopoulos has been the Principal Investigator of multiple research projects funded by the U.S. National Science Foundation, the U.S. National Geospatial-Intelligence Agency, the U.S. Air Force Office of Scientific Research, and the U.S. Air Force Research Laboratory. In 2020, he was the recipient of the prestigious AFOSR Young Investigator Award. He is a member of the IEEE Signal Processing, Computer, and Communications Societies, with high service activity that includes the organization of multiple conference events, such as the 2019 IEEE International Workshop on Machine Learning for Signal Processing, the 2019–2021 versions of the SPIE DCS Conference on Big Data: Learning Analytics and Applications, and the 2017–2020 versions of the IEEE International Workshop on Wireless Communications and Networking in Extreme Environments.

Eli Saber (Senior Member, IEEE) received the B.S. degree in electrical and computer engineering from the University at Buffalo, Buffalo, NY, USA, in 1988, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Rochester, Rochester, NY, USA, in 1992 and 1996, respectively.

He is currently a Professor with the Department of Electrical and Microelectronic Engineering, Kate Gleason College of Engineering, Rochester Institute of Technology, Rochester, NY, USA. Prior to that, he was with Xerox Corporation from 1988 to 2004. He holds more than 100 peer-reviewed publications and is the coauthor of a textbook titled *Advanced Linear Algebra for Engineers With MATLAB* (CRC Press, 2017). His research interests include digital image and video processing, remote sensing, and computer vision.