Edge detection-based boundary box construction algorithm for improving the precision of object detection in YOLOv3

National Institute of Technology Tiruchirappalli, India

Introduction

- Goal to improve the precision of the boundary box around the object by using edge detection and pixel values in an area.

- Object detection has been used in many fields
  - self-driving cars
  - pedestrian counting
  - face recognition
  - security systems

- Pretrained COCO dataset for object detection.

Importance of research

Precise boundary boxes => minimize hardware cost.

Self-driving cars a robotic system => hardware sensors

= Better object detection software to precisely locate the position of object

Previous works

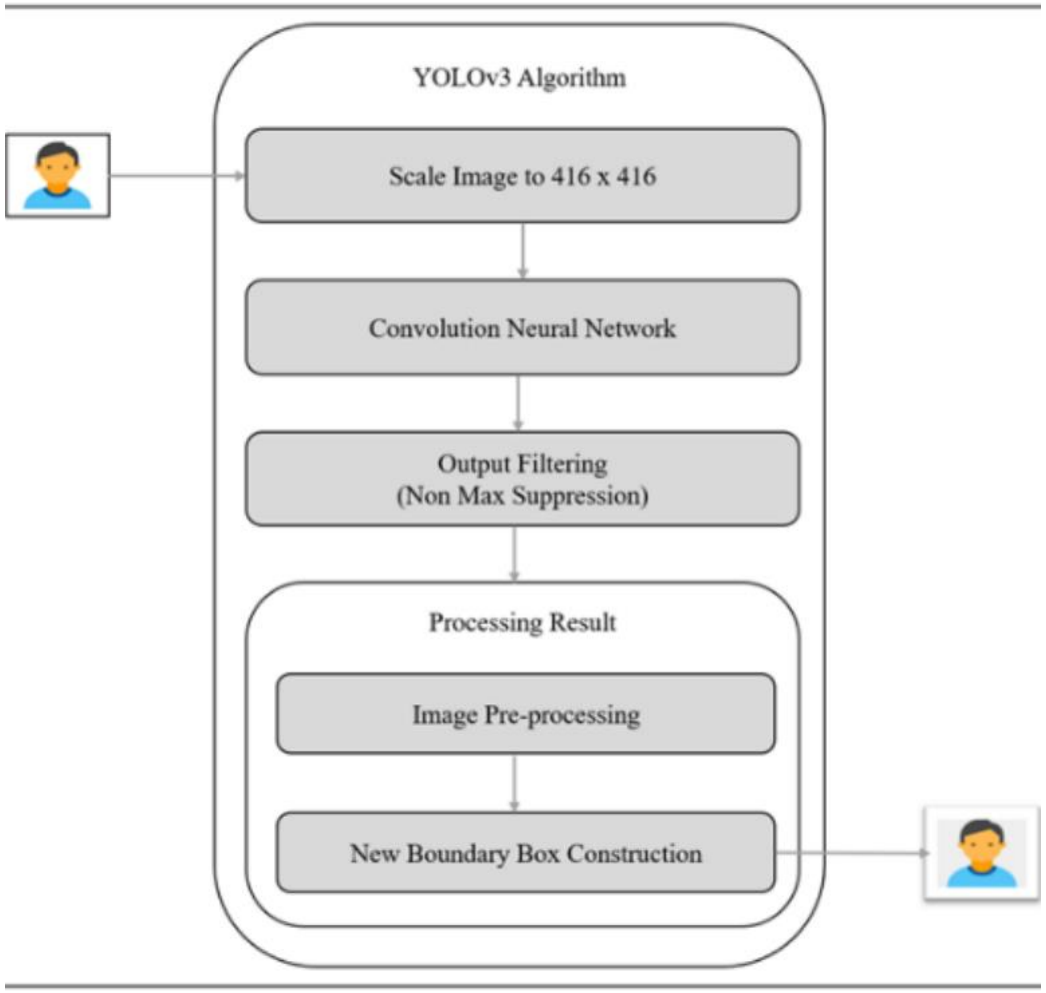
- Deformable Parts Model (based on score)
  - Dividing each part
  - Based on distance from the main image scoring likeliness
  - Also known as sliding window approach
- R-CNN (Regional proposal)
  - Selective search generates potential bonding boxes
  - SVM scores the box
  - Non-max suppression eliminates duplicate detection.

YOLO and Multibook

- Yolo adds SoftMax layer to multibook parallel to box regressor and box classifier layer to classify and object.
- Uses grid approach for boundary box detection
- Proposed = adding a new layer of edge detection + algorithms to improve precision of boundary boxes.

Proposed algorithm

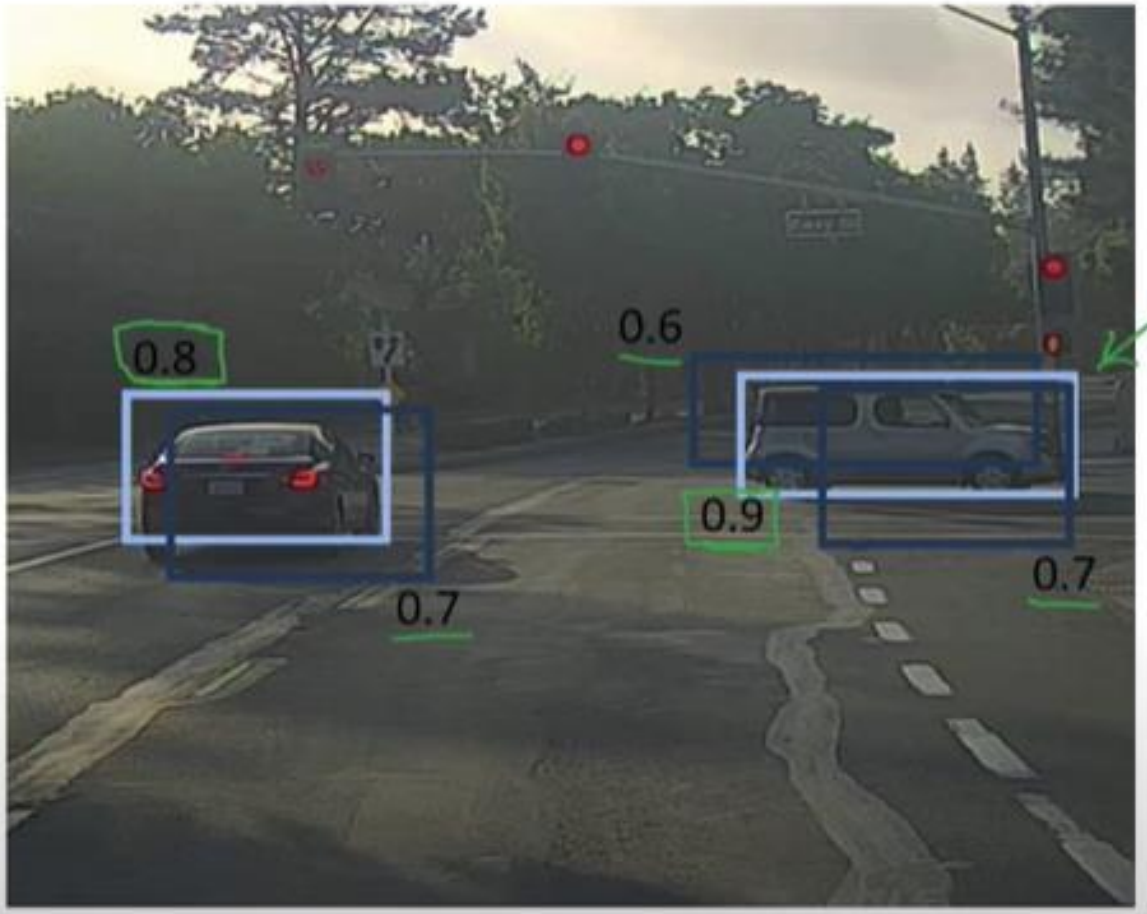
- YOLOv3 scales the image to 416x416
- CNN => Tensor



YOLOv3 Object Detection Algorithm

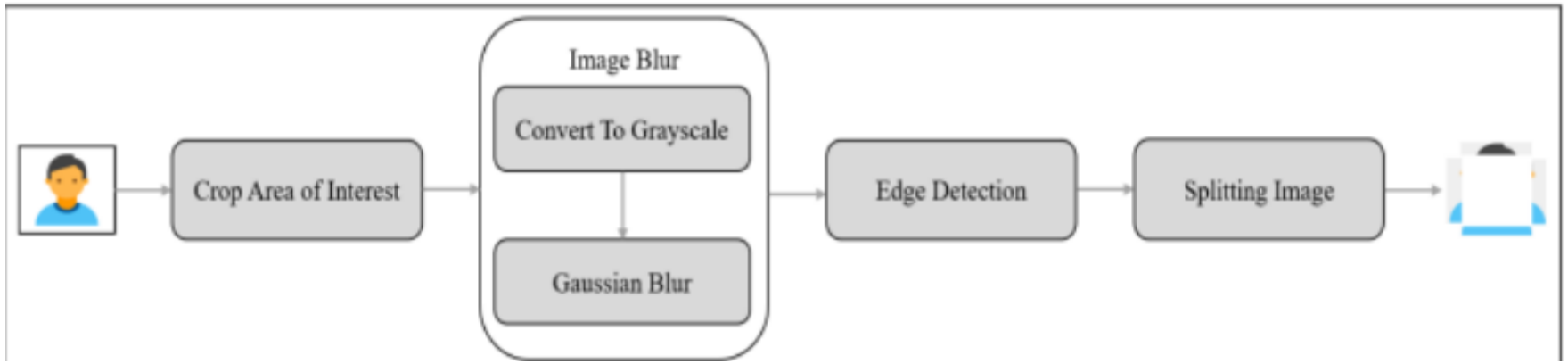
- 416x416, Neural Network 106 layers, consisting of residual layer and un-sampling layer and skip connection.
- CNN => a probability value that each boundary box contains an object and class
- Boundary box < .5 ignored
- Non max suppression algorithm filters out false predicted boundary boxes
- For training, pre-trained weights of COCO dataset for detection were used. These pre-trained weights are capable of detecting object in 80 different classes

Non-max suppression example



p_c

Image preprocessing



Pre-processing is applied to improve the performance and to obtain an image which contain edges of detected object.

1. Area of interest
2. Image Blur : converting to gray scale and blurring using Gaussian blur function. kernel size = 3 x 3,

Sigma value used in this work is 0.33

1. Design the kernel

1. The formula to design 2D Gaussian Kernel: $$\frac{1}{2\pi\sigma^2}.e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

2. Let us consider the standard deviation, sigma =0.6 and the Kernel size =3 X 3

3.

$$\frac{1}{2\pi\sigma^2} = \frac{1}{2\times 3.14\times 0.6\times 0.6} = \frac{1}{2.2619}$$

4. The width of the kernel is X = 3 and the height of the kernel is Y =3

i.e $$X = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$ and $$Y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\frac{-(x^2+y^2)}{2\sigma^2} = \begin{bmatrix} -2.7778 & -1.3889 & -2.7778 \\ -1.3889 & 0 & -1.3889 \\ -2.7778 & -1.3889 & -2.7778 \end{bmatrix}$$

Gaussian blur function

Edge Detection

- Canny edge detection algorithm
- $LowerThreshold = \max(0, (1 - sigma) * median)$
- $UpperThreshold = \min(255, (1 + sigma) * median)$

Image Partition

- Once edge is detected it divides the image further into four parts
- Next step is to pass these four images one by one to proposed algorithm.
- Each rectangle is analyzed one by one by moving in a particular direction

Threshold value

- Here x is used for number of rows and y is used for number of columns in an area.

$$sumofvalues = \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} pixelvalue_{(x,y)}$$

$$calculated\_Threshold = \frac{sum\ of\ values}{255 * Area}$$

$$Area = Number\ of\ rows\ (x) * Number\ of\ Colums\ (y)$$

Threshold that has been calculated gives a value in between 0 and 1,this tells us the density of white pixel in an area.

$$ThresholdValue = \begin{cases} 0.5, Top\ Boundary \\ 0.5, Bottom\ Boundary \\ 0.8, Left\ Boundary \\ 0.5, Right\ Boundary \end{cases}$$

If calculate threshold is greater than the threshold value then an error has occurred

(a) Input image

(b) Output of YOLOv3

(c) Area of Interest

(d) Converting to grayscale image and applying blur

(e) Edge Detection

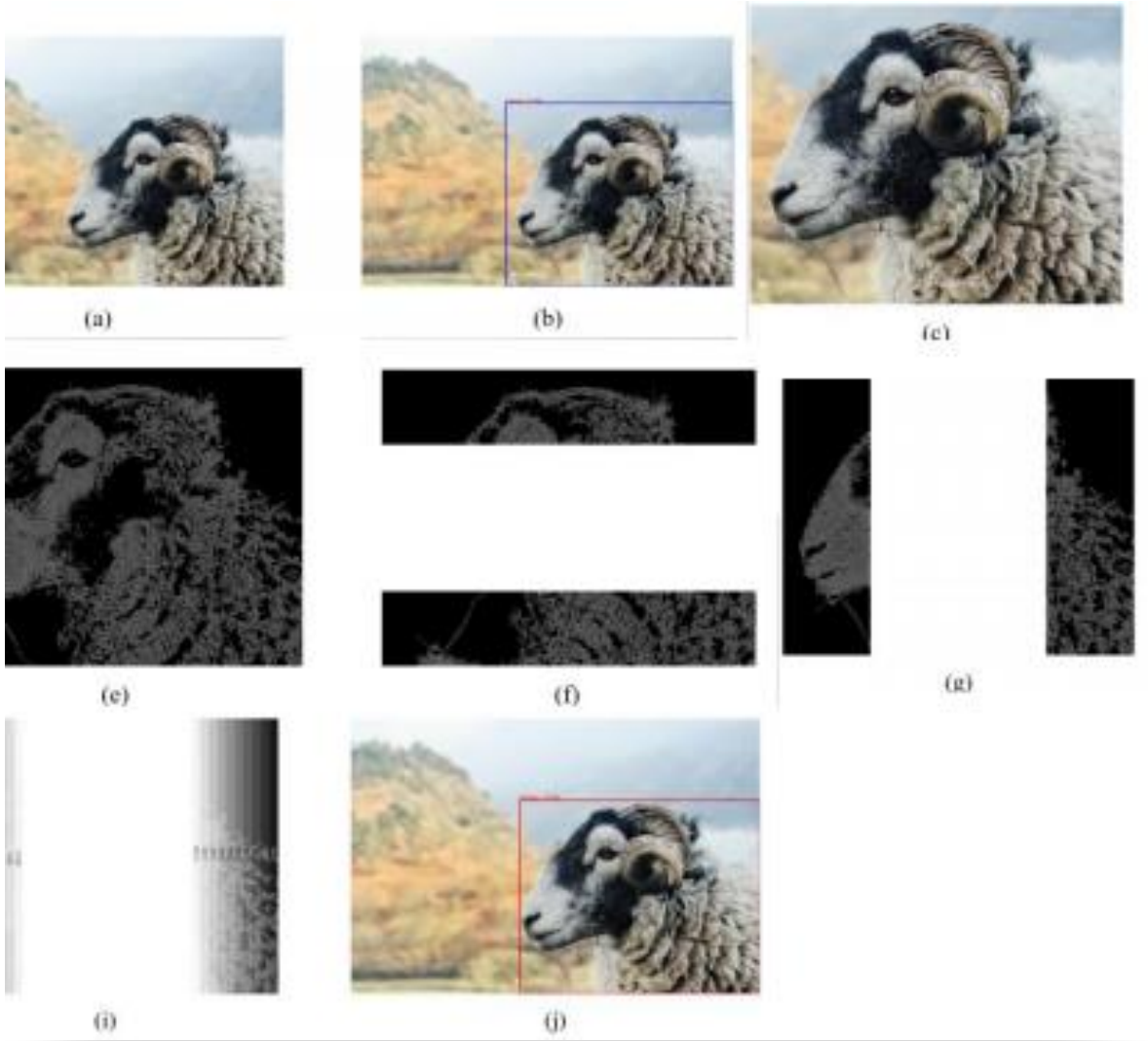
(f) Splitting top and bottom image

(g) Splitting left and right image

(h) Diving top and bottom image into rectangle of equal area

(i) (Dividing left and right image into rectangle of equal area (j)

(j) Output of proposed work



IOU Comparison Intersection over union

(IOU) is a parameter, which is used to test how precise boundary box is drawn with respect to ground truth. IOU is given by following formula.

IOU = Area of overlap/ Area of union

Comparison

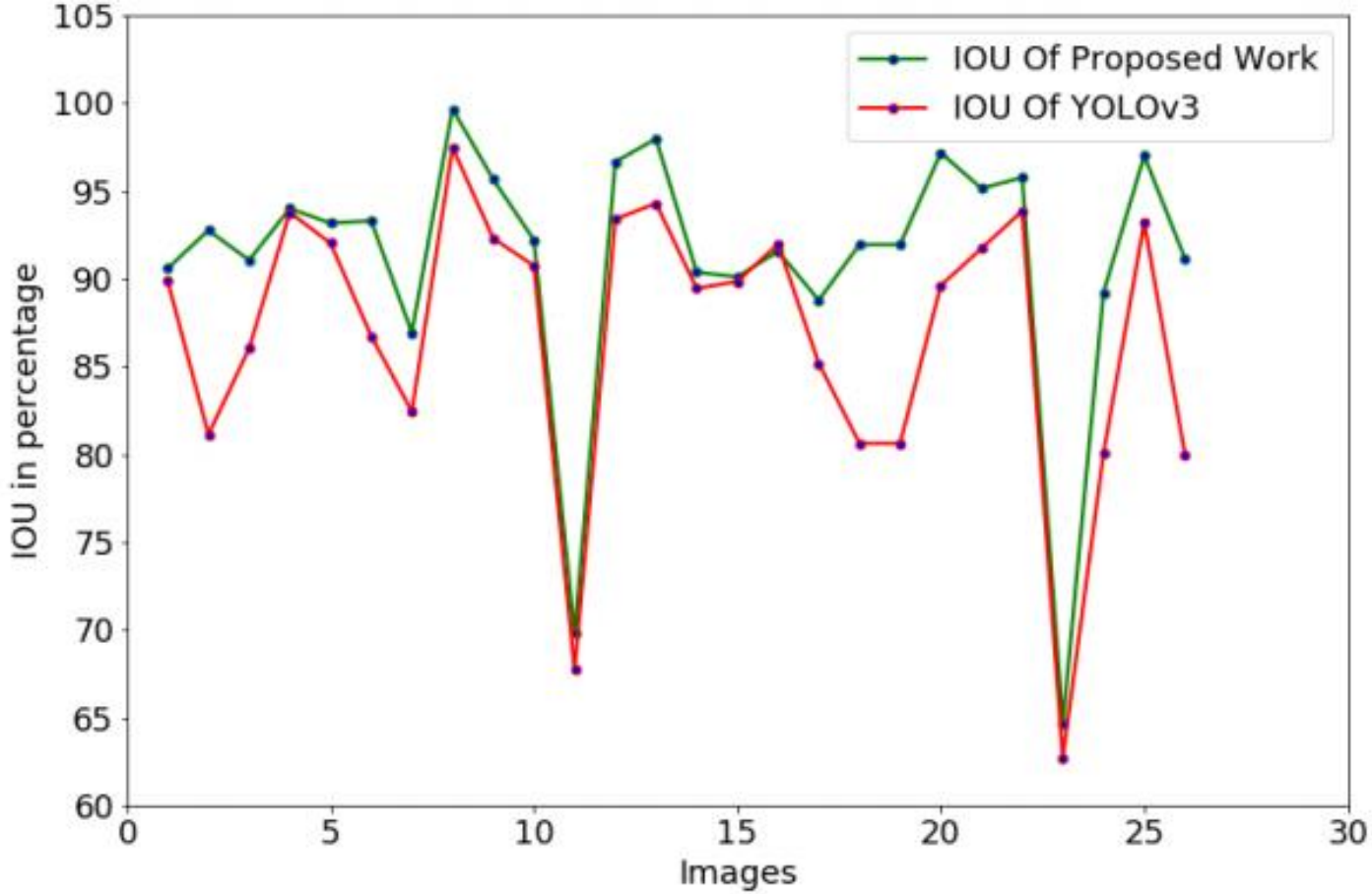
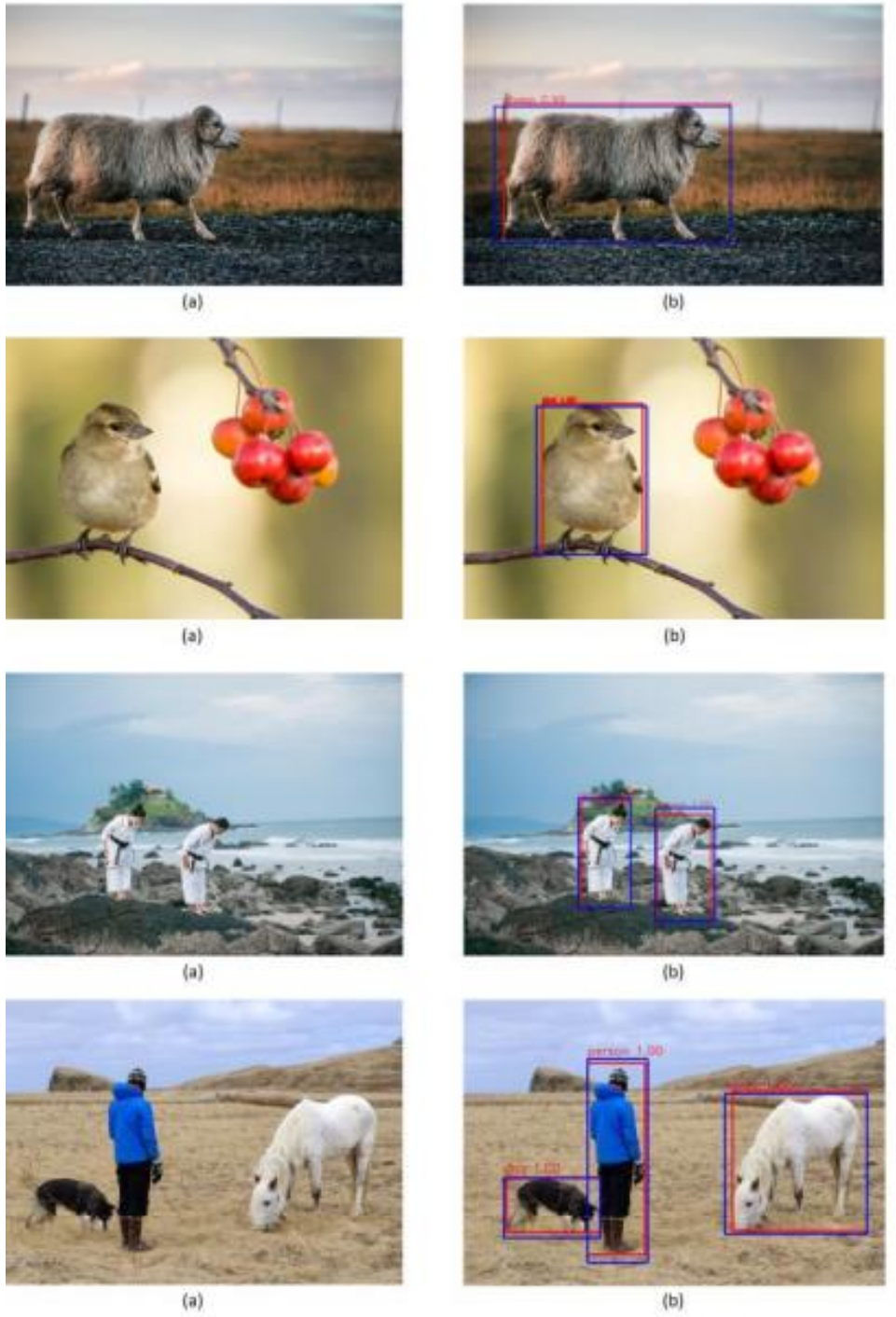


Fig. 6. A comparison graph for IOU of YOLOv3 [9] vs IOU of proposed

Red= Proposed Method

Blue= YOLOv3



Conclusion

- Proposed method has a better boundary box prediction
- Shortcoming of research images with high resolution and noise

Thank You ☺