Wirtschaftsinformatik
und Maschinelles Lernen
Stiftung Universität Hildesheim
Marienburger Platz 22
31141 Hildesheim
Prof. Dr. Dr. Lars Schmidt-Thieme
Mofassir ul Islam Arif, M.Sc.

Research Proposal

"Using non-square bounding boxes in object detection with improved IoU"

Syed Khalid Ahmed

276970
khalidahmed.nedian@gmail.com

10th May 2019

Abstract

Current architectures used in object detection networks rely on rectangular bounding boxes to perform object detection and localization. However, there are certain drawbacks to this approach in that a lot of background noise also gets incorporated in these regions. To solve this problem, this research explores the use of non-rectangular bounding box for optimizing the Intersection over Union (IoU) accuracy metric in an object detection network. Specifically, we want to find the most relevant Region of Interest (ROI) with reduced background noise. This research uses a pre-existing object detection network called Faster-RCNN and allows the network to generate non-rectangular bounding boxes to perform object localization and detection. The dataset used is PASCAL VOC. The bounding boxes in the annotated images are converted into non-rectangular shape to feed the network and learn from it. The outcome of this research will be an object detection network capable of generating non-rectangular bounding boxes with improved IoU scores on the dataset in consideration.

Contents

1 Introduction **4**

1.1 Problem Statement 5

2 Literature Review **5**

2.1 Research Question 5

2.2 Objectives 5

2.3 Related Work / State-of-the-Art 5

2.4 Definitions 6

3 Methodology **7**

3.1 Circle-Circle IoU 7

3.1.1 When circles do not overlap each other 8

3.1.2 When a circle is inscribed inside another circle 9

3.2 Ellipse-Ellipse IoU 10

3.3 Inscribed ellipse 10

4 Dataset **12**

1 Introduction

Object detection refers to the process of identifying certain objects in a given image or video. This allows us to develop a number of applications which leverage this technique and train systems based on it. This technique is based on supervised learning, which means that we need to supply the system with training data in order for it to learn from. This usually includes pictures which have a highlighted area which we want to predict. If we have a picture with 1 cat and 1 dog, then we need to properly label the cat and dog with bounding boxes. Bounding box basically represents the area which is of interest to us. This usually includes a rectangular box enclosing the relevant object. The network learns the regions of interest using these rectangular bounding boxes. It then performs prediction based on it. The metric used by the network to learn these object is usually called Intersection over union (IoU). It gives the ratio of the overlapping area between the actual bounding box and the predicted bounding box of the neural network. Since a rectangle encloses the most area, therefore it is the most commonly used shape to annotate the images. However, there are certain cases in daily life where the object of interest is not necessarily rectangular. Wheels of a car, Round traffic signs, human face, etc. are some prevalent examples in daily life. Enclosing them inside a rectangular bounding box is not the most optimal solution since it also introduces noise from the background. This can be best understood from the image below:

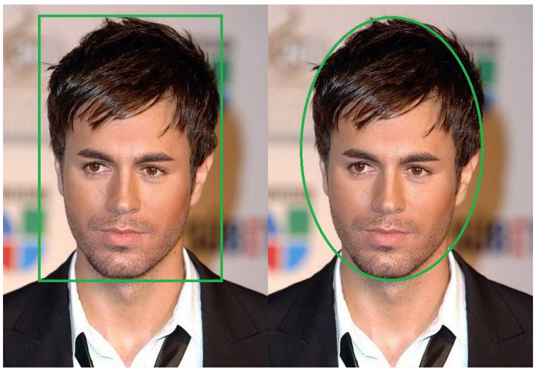

Figure 1: In the figure on the left side, the traditional rectangular bounding box is applied. As can be seen in the figure, the rectangle also introduces additional noise from the background. However when applying elliptical bounding box on the same image, the face is more accurately represented and there is less noise from the background.

1.1 Problem Statement

Given a non-rectangular object, we want to perform object detection with minimum background noise. Therefore, this research investigates the potential use of non-square bounding box e.g Ellipse, circle, diamond, etc. There are two benefits for using this approach:

- Since most objects are not rectangular, therefore using non-square bounding boxes will help us get a more accurate representation of the object when performing object detection.
- Since each geometrical shape has unique properties, this could help us in further refinement of the generated results.

2 Literature Review

2.1 Research Question

How would non-square bounding box improve the IoU so that the generated network can compete against state-of-the-art object detection networks.

2.2 Objectives

The objective of this research is an object detection network capable of generating non-rectangular bounding boxes with an improved IoU as compared to the state-of-the-art networks. This work can be integrated into the tensorflow object detection API.

2.3 Related Work / State-of-the-Art

Much work has been done in the field of object recognition. Uijlings et al. [2013] introduced the concept of region proposals in which multiple regions on an image are proposed to get a superficial view of the image. This saves computation as the network does not have to perform an exhaustive search on the whole image. Using this work, Girshick et al. [2014] proposed the method to find the candidate regions and classify them using a Convolutional Neural Network. It also predicts four coordinates for the bounding box in order to perform object localization. But R-CNN had drawbacks in that it was slow and very high training times. Therefore in order to optimize speed of the network, Girshick [2015] came up with the idea of passing the image first from a CNN without generating any regions beforehand. The CNN outputs a feature map, which shows the area of interest. Using these maps, region of proposals are generated and concatenated with other regions to generate an NxN image indicating Region of Interest (RoI). By using this technique, Fast R-CNN achieved significant speedups as compared to simple R-CNN. But the above algorithms used selective search for region proposals. To decouple the generation of region proposals, Ren et al. [2015] proposed a method in which a separate network learns to perform region proposals. These proposals are reshaped using the RoI pooling layer.

All of the algorithms described previously used region proposals to perform object localization. However in YOLO proposed by Redmon et al. [2016], we divide the whole image in an NxN grid and generate m bounding boxes in them. The network then predicts the probability value for each of these boxes.

2.4 Definitions

The following concepts will help us understand the research in more detail.

- **Intersection over Union (IoU):** It is also called Jaccard index. It is an accuracy metric used to measure the accuracy of an object detector.

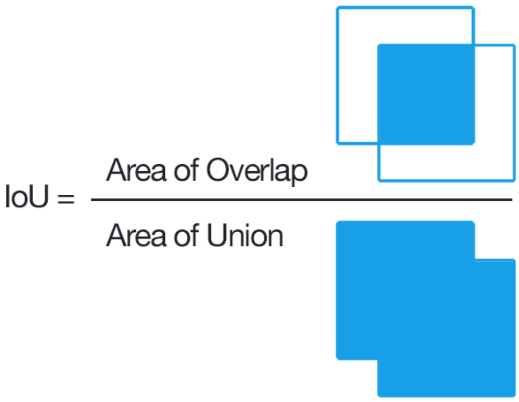

Figure 2: Graphical illustration of IoU. Source: IoU [2019]

- **Object Detection:** It deals with the detection of semantic objects belonging to a certain category (humans, buildings, cars, etc.) in digital images and videos. There are numerous real-world applications for this domain such as Video surveillance, car detection, etc. Source: Object Detection [2019]

- **Convolutional Neural Network (CNN):** They are a special type of deep neural network which is most commonly used for visual imagery analysis. They perform convolution operation on image pixels to extract important features from images. Activation functions are used to classify either the image or the objects present in the image. Source: CNN [2019]

3 Methodology

The current work establishes a proof-of-concept for the research topic. The fundamental functions are implemented which would later be applied to the object detection network to achieve the objective. The work done so far encompasses the following domains:

3.1 Circle-Circle IoU

The research started with the implementation of the circle-circle overlap and calculating the IoU of them. Consider two circles with the following naming conventions:

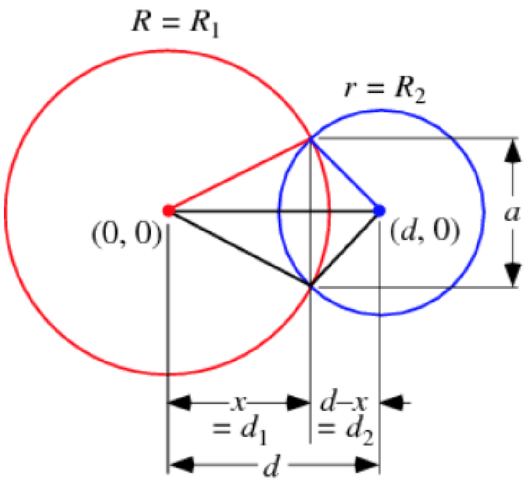

Figure 3: Attributes defined with appropriate variables. Source: Wolfram Alpha [2019]

Using the above representation, the following formula is used to calculate the overlapping area, which in turn is used to calculate the IoU:

$$A = r^2 \arccos\left(\frac{d^2 + r^2 - R^2}{2dr}\right) + R^2 \arccos\left(\frac{d^2 + R^2 - r^2}{2dR}\right) - \frac{1}{2}\sqrt{(-d + r + R)(d + r - R)(d - r + R)(d + r + R)} \tag{1}$$

This is the most simple geometrical shape from which the IoU can be calculated easily. The library used is Matplotlib [2019] to generate the circles. The following figure illustrates this concept:

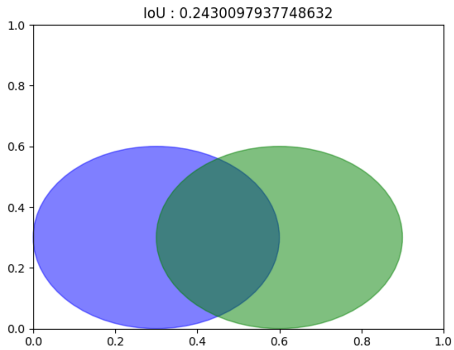

Figure 4: Calculating IoU of two circles using circle overlap formula.

There are two special cases in the circle-circle overlap in which the original formula does not provide any solution. These are mentioned below:

3.1.1 When circles do not overlap each other

If the circles do not overlap each other at all, then they intersect on an imaginary point. This is shown in the figure below:

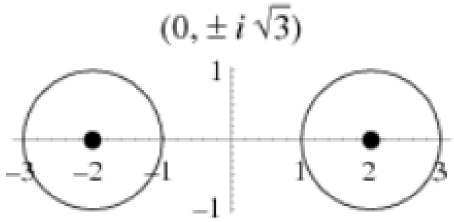

Figure 5: Two circles intersecting at imaginary points. Source Wolfram Alpha [2019]

In order to overcome this problem, we make the assumption that if the distance between the two circles is greater than the sum of their radii, then it shows that the circle do not overlap at all. This is shown in the formula below:

$$\textbf{if } \text{distance} \geq (R + r) \textbf{ then}$$
$$\text{IoU} = 0$$

The following figure illustrates this concept:

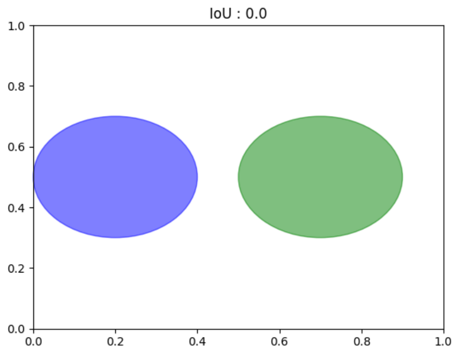

Figure 6: IoU for imaginary intersection.

3.1.2 When a circle is inscribed inside another circle

This case arises when the two circles have unequal radii and one circle lies inside the other circle. In order to solve this special case we make the assumption that if the sum of the distance between the two circle and the radius of the smaller circle is greater than the radius of the bigger circle, this means that the overlapping area is equal to the area of the smaller circle. This is shown in the formula below:

$$\textbf{if } \text{distance} + \min(R, r) \leq \max(R, r) \textbf{ then}$$
$$\text{IoU} = \textit{Area of smaller circle}$$

The following figure illustrates this concept:

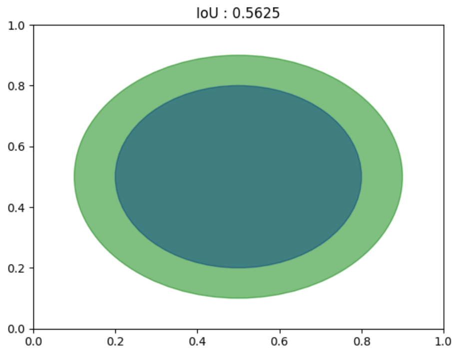

Figure 7: IoU for inscribed intersection.

3.2 Ellipse-Ellipse IoU

In order to find the IoU of two ellipses, we made a proof-of-concept study which uses the Shapely [2019] library to generate the ellipses to calculate the IoU between them. This library also allows the generation of different geometrical shapes, which can later be used in the research. The following figures illustrates the concept.

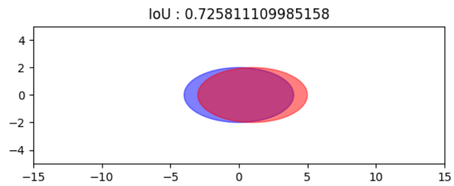

Figure 8: IoU for intersecting ellipses.

3.3 Inscribed ellipse

Since the dataset used has rectangular bounding boxes as the ground truth, therefore it becomes necessary to convert these to non-rectangular in order to feed the network and train on them. Our work initially starts with ellipse as the candidate for non-rectangular bounding boxes. Since the datasets that are currently used to train deep neural networks have rectangular bounding boxes, therefore there is need for a dataset which suits our purpose. Therefore, the approach taken in this research is to convert the existing ground truth annotations to ellipse so that we avoid the tedious task of annotating images manually. To achieve this objective, the initial work generates an inscribed ellipse inside a rectangle. The following figures illustrate this concept:

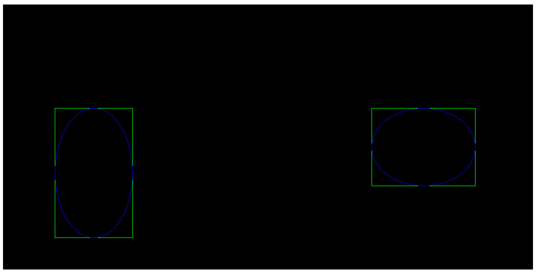

Figure 9: Ellipses inscribed inside a rectangle.

In order to obtain these inscribed ellipses, the following notation to denote the axis points of a rectangle are used:

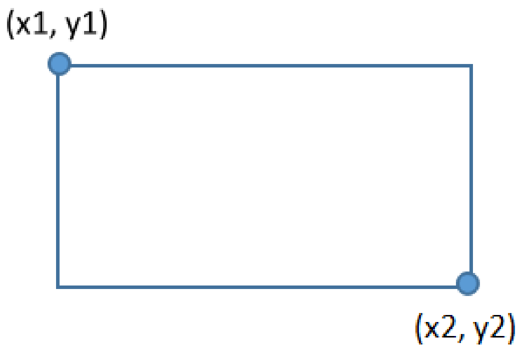

Figure 10: Graphical illustration of axis points.

To find the major axis, we use the following formula:

$$(x1 - x2) / 2$$

Similarly to find the minor axis:

$$(y1 - y2) / 2$$

To find the origin points, we provide the offset the above generated points:

$$x\_origin = x1 + (x1 - x2) / 2$$
$$y\_origin = y1 + (y1 - y2) / 2$$

4 Dataset

The dataset that will be used for the research is the Pascal VOC provided by Everingham et al. [2010].

References

CNN. Convolutional neural network — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=896271372`, 2019. [Online; accessed 9-May-2019].

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

IoU. Iou illustration. `https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/`, 2019. [Online; accessed 9-May-2019].

Matplotlib. Convolutional neural network — Wikipedia, the free encyclopedia. `https://matplotlib.org/`, 2019. [Online; accessed 9-May-2019].

Object Detection. Object detection — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Object_detection&oldid=878674134`, 2019. [Online; accessed 9-May-2019].

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

Shapely. Convolutional neural network — Wikipedia, the free encyclopedia. `https://shapely.readthedocs.io/en/stable/manual.html`, 2019. [Online; accessed 9-May-2019].

Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

Wolfram Alpha. Convolutional neural network — Wikipedia, the free encyclopedia. `http://mathworld.wolfram.com/Circle-CircleIntersection.html`, 2019. [Online; accessed 9-May-2019].