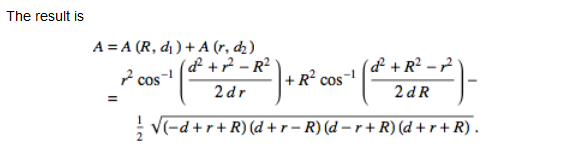
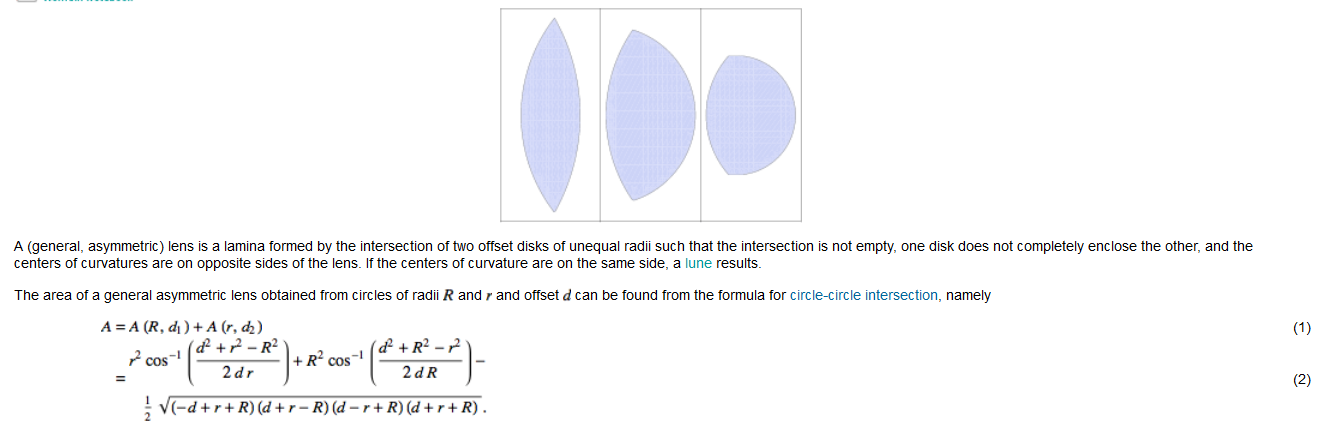
Documentation for Thesis work



<http://mathworld.wolfram.com/Circle-CircleIntersection.html>



<http://mathworld.wolfram.com/Lens.html>

<https://www.xarg.org/2016/07/calculate-the-intersection-area-of-two-circles/>

-> I have used this formula to calculate the area of the intersection of 2 circles.

-> Generalized IoU

<https://arxiv.org/abs/1902.09630>

-> Library for making shapes and manipulating them in the cartesian plane

<https://github.com/Toblerity/Shapely>

-> ellipse ellipse overlap

<https://stackoverflow.com/questions/48808941/overlap-area-of-2-ellipses-using-matplotlib>

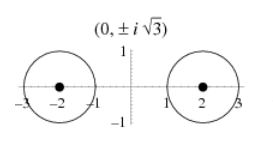
-> I used shapely to generate built-in functions. Then I used built-in methods to calculate the overlapping area and find the IoU.

For script *bounding box smaller circle.py*

* This script calculates the IoU of two overlapping circles by finding the overlapping area between the 2 circles.
* There are 2 special cases where the original formula does not provide any solution. These are solved by me using concepts from geometry

1. When the 2 circles do not overlap each other:

According to the theory provided by [wolfram alpha](http://mathworld.wolfram.com/Circle-CircleIntersection.html), they intersect on an imaginary point



Due to this, I used the following logic:

If distance between the 2 circles is greater than the sum of their radii, then it shows that the circle do not overlap at all

if distance > (radius\_circle1 + radius\_circle2):

1. When 1 circle lies inside the other circle:

This happens when the 2 circles are of unequal length. To solve this case, I use the following logic:

If the distance between the 2 circles + the smaller circle is greater than the radius of the large circle, then this means that the overlapping area would be equal to the area of the smaller circle.

if (distance + min(radius\_circle1, radius\_circle2)) < max(radius\_circle1, radius\_circle2):

For script *inscribed\_circle.py*

* This script inscribes a cirlce inside a square.
* This is useful for converting the datasets which are already annotated. Otherwise we would need to manually label each dataset again.
* We can also inscribe different shapes inside the square to compare against other shapes.

For script *ellipse ellipse overlap.py*

* This script calculates the overlapping area between 2 ellipses.

**Final Version for Thesis:**

**Abstract:**

This research explores the use of non-square bounding box for object detection. Specifically, we want to find the most interesting Region of Interest (ROI) by optimizing the Intersection over Union (IoU). This evaluation metric is mostly used to find the accuracy of an object detection network.

**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 an 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, food items 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.

**Problem Statement:**

In order to solve the above mentioned problem, this research investigates the potential use of non-square bounding box e.g Ellipse, diamond, etc. There are two benefits for using this approach:

1. Since most objects are not rectangular, therefore using non-square bounding boxes will help us get more accurate representation when performing object detection
2. Since the area enclosed by the circle is less than the area enclosed by a rectangle, therefore the IoU will get decreased.

**Challenges:**

Some of the challenges that are present at the moment are listed below:

1. Converting the existing datasets’s annotated bounding box to the geometrical shape of choice
2. Making the existing networks learn non-square bounding boxes as well as generating them.

**Outcome:**

The outcome 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.

Introvduction:

**Abstract:**

* This research explores the use of non-square bounding box for object detection. Specifically, we want to find the most interesting Region of Interest (ROI) by optimizing the Intersection over Union (IoU). This evaluation metric is mostly used to find the accuracy of an object detection network.
* If this works, this can be a step towards semi object segmentation. This means that we can also have some element of object segmentation, depending upon the nature of the bounding box.
* This work can be integrated into object detection.
* Future work could include more complex convex bounding boxes which completely encapsulate the relevant object in an image.
* Current methods use square bounding boxes to label the ground truth of an object in an image.
* Some objects in real life are not rectangular. Therefore using rectangular bound boxes to label is not the most optimal approach.
* Some objects are based on different geometrical structures. Example: Wheels of a car, round traffic signs, human face, etc.
* Therefore it makes sense to encode them using the geometrical shape that best describes(encapsulate) them.
* Can be included in the Tensorflow object detection API.
* The idea is that since the circle has smaller area as compared to a square, therefore, the IoU will be decreased.

**Related work**

0

1000

800

400

200

Kosten in GE/ZE

10

2000

1800

1600

1400

1200

5

3

25

2

15