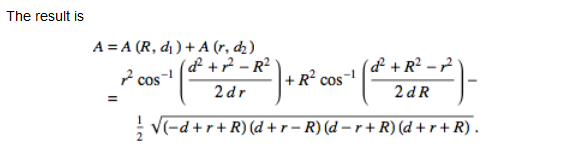
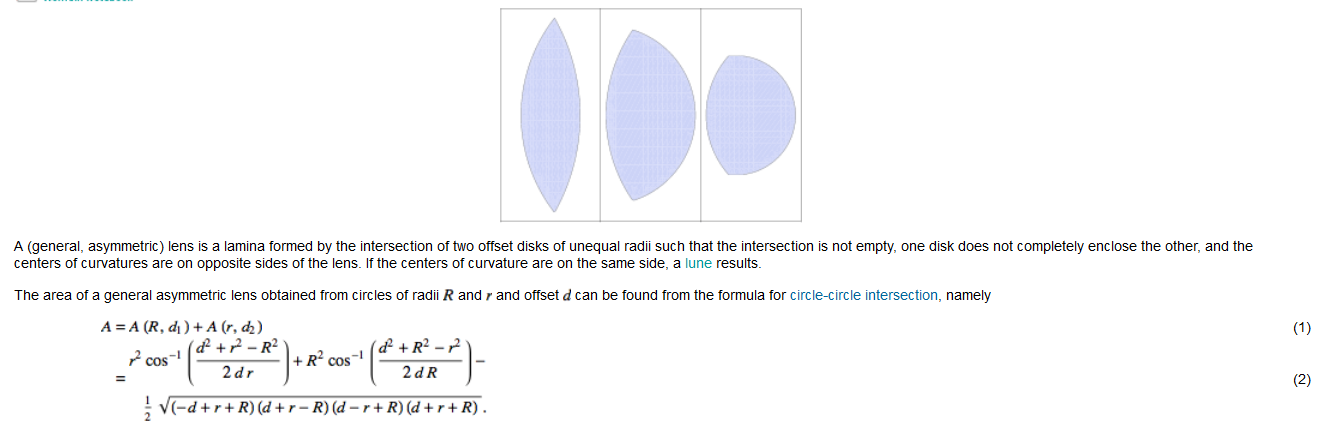
Documentation for Thesis work



<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>

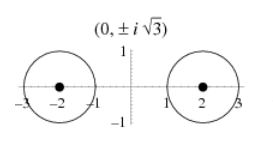
-> I used shapely to generate built-in functions. Then I used built-in methods to calculate the overlapping area and find the IoU.

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.

**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.
* There are two challenges that are present at the moment.
  + Converting the existing datasets’s annotated bounding box to the geometrical shape of choice
  + Making the existing networks learn non-square bounding boxes as well as generating them.
* 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.

A

A

A

A

A

A

A

A

A

0

1000

800

600

400

200

Kosten in GE/ZE

10

2000

1800

1600

1400

1200

Gesamtkosten

Rüstkosten - - - - - - - - -

Lagerkosten ..................

5

30

25

20

15