GSoC 2023

Emory University organization

Project Idea: Auto-detect coverage bounding boxes for brain MRI images

Yasien Essam Mohamed
Faculty of Engineering – Cairo University
Cairo, Egypt
Biomedical Engineering department

Table of Contents

1. Abstract / Project Summary

This project is aimed at assessing the anatomical coverage of different brain regions through a series of MR images. The ultimate goal is to develop an automated approach to detect and measure the extent of anatomical coverage and compare it with pre-defined protocol constraints. The methodology will ensure that the acquired MR image data meets the specific anatomical coverage requirements set forth by the protocol, which will help us ensure the quality of the images is up to the standard and reduce the need for additional exams to show the whole desired area.

The project will mainly work with DICOM images provided by the PACS system. The module will be integrated into a comprehensive pipeline to evaluate the overall quality and compliance of the MRI modality. By developing this module, we can save time and increase the efficiency of the MRI process.

2. Contributor Information:

Name	Yasien Essam Mohamed
University	Cairo University
Faculty	Engineering
Department	Biomedical Engineering
Country	Egypt
E-mail	yasien.ghalwsh00@eng-st.cu.edu.eg
Github	https://github.com/Yasien99
LinkedIn	https://www.linkedin.com/in/yasien99/

3. Potential Mentor(s):

- **Puneet Sharma (puneet.sharma@emory.edu)**
- **Tony Pan (tony.pan@emory.edu)**

4. Personal Background (Brief CV):

- **Education:**

- Bachelor in Biomedical Engineering at Cairo University · Faculty of Engineering
- My GPA: 3.4
- Undergraduate Courses: Machine Learning, Computer Vision, Computer Graphics, Probability, Statistics, Calculus, Linear-Algebra, Digital Signal Processing, Data Structures, Algorithms, Databases, Embedded Systems, Image Processing, Deep Learning.

EXPERIENCE:

- Software Engineer Intern @Astute Imaging from Oct 2022 - May 2023
  - Develop a web-based Dicom viewer with the ability to segment breast cancer ultrasound images automatically.
  - Tools: Fast API, AWS, Image Processing, Deep Learning, PACs

- Machine Learning Engineer Intern @Omdena from Mar 2023 - May 2023
  - Identifying immature leukemic blasts from normal cells under the microscope is challenging due to morphological similarity using Computer Vision.
  - Tools: Image Processing, Tenserflow, open-cv, Deep Learning.

- Service Engineer Intern @Siemens Healthineers from Jul 2022 - Aug 2022
  - Know the main components and Basics of MRI, CT, PET/CT and medical laboratory equipment

- Deep Learning Summer Academy Student @Neuromatch from Jul 2022 to Aug 2022
  - 3-week intensive tutorial-based training program about Basics DL (Linear Deep Learning, Multi-Layer Perceptron, Optimization, Regularization), CNN, NLP and Reinforcement Learning.

Personal Projects:

- Diabetic Prediction
  - Predict whether the patient has diabetes or not using the PIDD dataset using different ML algorithms.
  - Tools: seaborn, numpy, matplotlib, pandas

- GI-Tract-Image-Segmentation
    - Used UNet to effectively segment the stomach and intestines in MRI scans to improve the cancer treatment to avoid high doses of radiation to healthy tissues.
    - Tools: Tensorflow, matplotlib, pandas, numpy

5. Project Goals / Major Contributions:

- Project Goals:
    1. Develop an automated approach for detecting and measuring the extent of anatomical coverage in brain MR images.
    2. Compare the measured anatomical coverage with pre-defined protocol constraints to ensure compliance.
    3. Integrate the developed module into a more extensive pipeline for assessing overall quality and compliance with MRI modalities.
    4. Enhance the accuracy and efficiency of the image assessment process.
    5. Reduce the time and resources required for evaluating the quality of MRI images.
- Major Contributions:
    1. Development of a novel automated approach for anatomical coverage detection and measurement.
    2. Data preparation and preprocessing: Collect and preprocess DICOM images to prepare them for machine learning models from the cancer archive or private data provided by the mentors.
    3. Feature extraction: Extract relevant features from the images, such as intensity, texture, shape, and location and grey level occurrence like GLCM and Haralick Descriptors from the images.
    4. Model training: Train a machine learning model, such as a support vector machine or a random forest, on the extracted features and labelled data to predict the coverage bounding boxes for brain regions.
    5. Model evaluation: Evaluate the model's performance on a held-out test set and fine-tune the hyperparameters to improve its accuracy.
    6. Integration into the pipeline: Integrate the trained model into the existing pipeline to automate the detection and measurement of anatomical coverage for MR images.

7. Improvement in the accuracy and consistency of MRI quality assessment.
8. Streamlining of the MRI quality assessment process, resulting in cost savings and increased efficiency.
9. Contribution to the field of biomedical engineering by publishing a paper on this topic, to share our findings and enable other researchers to build on our work,

6. Project Schedule

Phases	Estimated Start/End time	Tasks
Pre-Coding Phase	**4 April – 3 May**	● Get to know more about the pre-defined protocol constraints for the specific anatomical coverage requirements of brain regions. ● Identify and research appropriate image processing techniques and algorithms to auto-detect and measure the extent of anatomical coverage in MR images.
Community Bonding	**May 4- May 31**	● Get to know the mentor, review the project details and scope, and clarify any doubts or questions. ● Review the relevant literature on DICOM image processing, auto-detection of anatomical landmarks, and compliance assessment methods and rank the reviewed papers so I can apply their algorithms. ● Review the private data provided by the organisation and make sure that images should be labelled with ground-truth coverage bounding boxes or help with the annotation process if needed.
Development Phase	**June 1 - June 7**	● Preprocess the data to extract relevant features and ensure consistency in image resolution and orientation. ● Use feature selection techniques to identify the most important features for the detection of coverage bounding boxes.
	June 8 - June 14	● Evaluate various machine learning algorithms found from the literature review and test all of them to choose the best fit for our problem and data.

		● Test the model on the training data and fine-tune it to improve its accuracy.
Development Phase	**June 15 - June 21**	● Test the algorithm on the new data from different distributions from the training data to assess its robustness and generalizability. ● Make observations about model performance and what are the failing cases
	June 22 - June 28	● Ensure that I handle the corner cases of the model performance
	June 29 - July 5	● Create a pipeline for the model so it can be scaled up to handle larger datasets and more complex models, making it easier to merge it with Niffler.
Phase 1 Evaluation	**July 6 - July 12**	● Submit a mid-term progress report to my mentor and the GSoC community ● Reflect on the work done so far and identify any areas that need more attention or refinement
	July 13 - July 20	● Address any issues or bugs identified during the mid-term evaluation ● Start documenting the code and the algorithm for future reference and use. ● Began writing the paper on our model and exploring different conferences that will be interested to publish our work.
	July 21 - July 26	● Continue documenting the code and the algorithm ● Refine the training and testing datasets and run additional experiments to validate the system's accuracy and effectiveness ● Implement any additional features or enhancements based on feedback from my mentor and the community
	July 27 - August 2	● Continue documenting the code and the algorithm ● Refine the training and testing datasets and run additional experiments to validate the system's accuracy and effectiveness ● Implement any additional features or enhancements based on feedback from my mentors.

	August 3 - August 23	● Finalize the code and the algorithm and conduct comprehensive testing on a variety of brain MRI images ● Prepare final documentation and user manual for the system ● Work with my mentor to ensure that the code is well-documented and that the system is ready for deployment ● Finalize the paper and sent it to different conferences.
	August 24 - September 6	● After Having stable model performance we should begin integrating the model with Niffler ● Understand Niffler code base and run it on my local computer and understand the code design pattern
	September 7 - September 27	● Intergrate the code and build a module in Niffler and plan the module workflow ● Test the module performance and stability
	September 28 - October 4	● Add unit test for the module and try to containerize the application ● Ensure that the code is matching Niffler code base design and style and add comments and documentation
	October 5 - October 11	● Improve the documentation and user manuals for the code, including examples, screenshots, and tutorials, to facilitate its dissemination and reuse.
	October 12 - October 18	● Complete any remaining features or improvements to the module. ● Seek feedback and guidance from the mentor for the performance of the module.
Final evaluation	**October 19- November 1**	● Finalize any remaining work and ensure that the module is functioning correctly. ● Conduct final testing and validation of the module.
	November 2 - November 6	● Prepare for the final evaluation and submission ● Submit the final code and documentation to my mentor and the GSoC

	November 6 **-** **....**	●Continue contributing to the project by: ⇒ Solving more issues. ⇒ Adding more options to the project.

7. Planned GSoC work hours

Start/End date	Rate of hours/week
4 May – 24 May	25-35 hours/week
25 May – 20 June	10-14 hours/week
20 June – 15 July	25-35 hours/week
15 July – 21 July	10-14 hours/week
22 July - 4 Nov	35-45 hours/week

8. Planned absence/vacation days

- I will graduate this summer so I will need to take a break during my final exams from May 27 to 20 Jun.
- Also, I will defend my bachelor's thesis on Jul 21, so I will need to take a break for a week before it from Jul 15 to Jul 21.

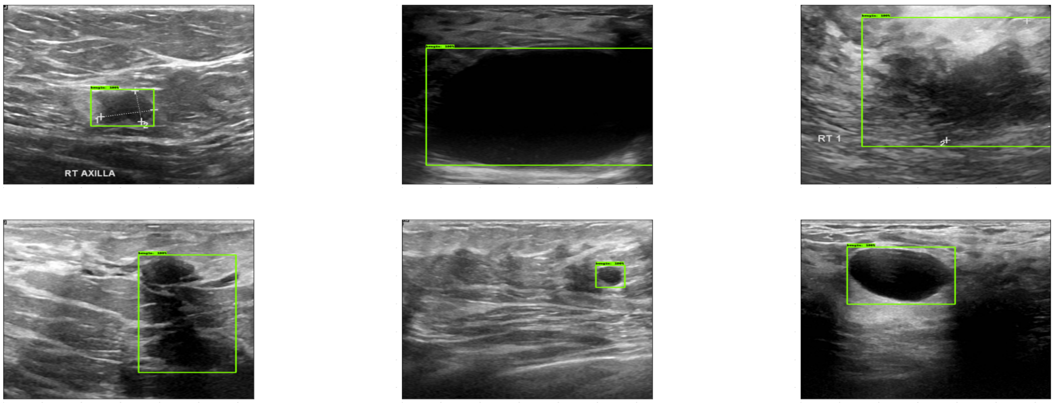
9. Skill set

- Strong programming skills in Python
- Familiarity with DICOM images and basic human anatomy during my major studies in college.
- Experience with machine learning libraries and applied my knowledge in many projects.
- Previously worked on medical imaging projects, including a web-based AI solution for breast ultrasound
- Completed the Code Challenge and included a link to the code sample in my proposal

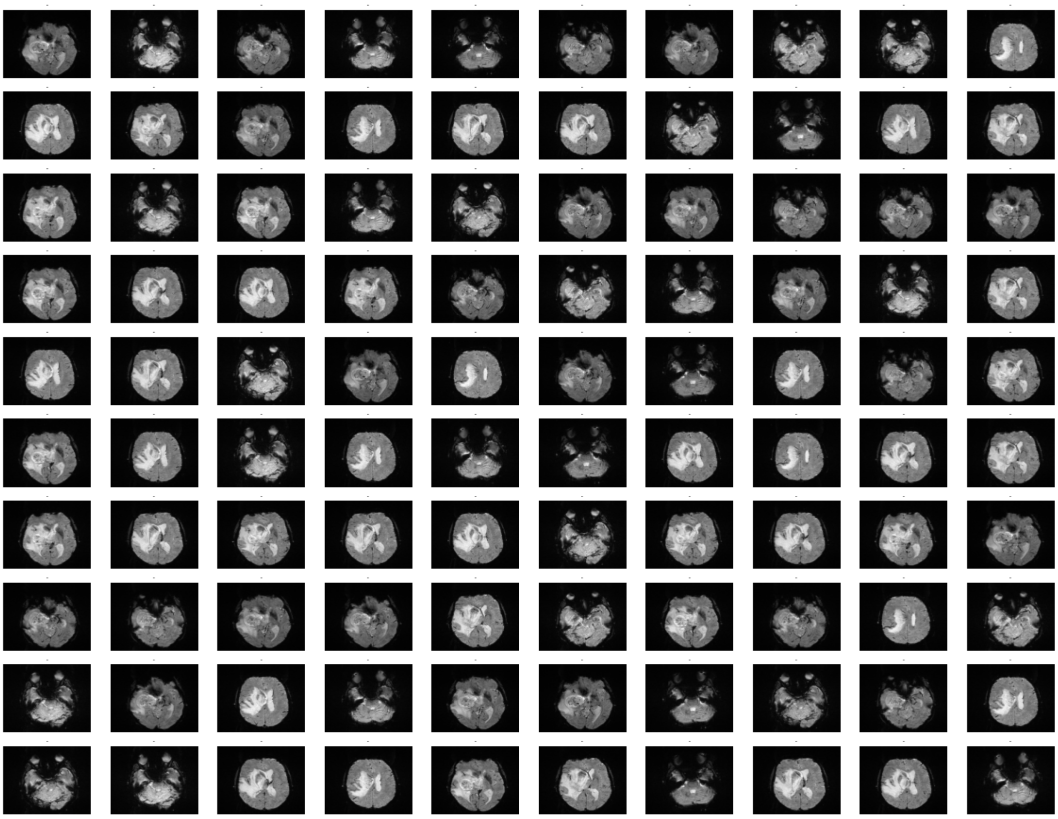
10. Code Challenge

For the Code Challenge, I had three phases:

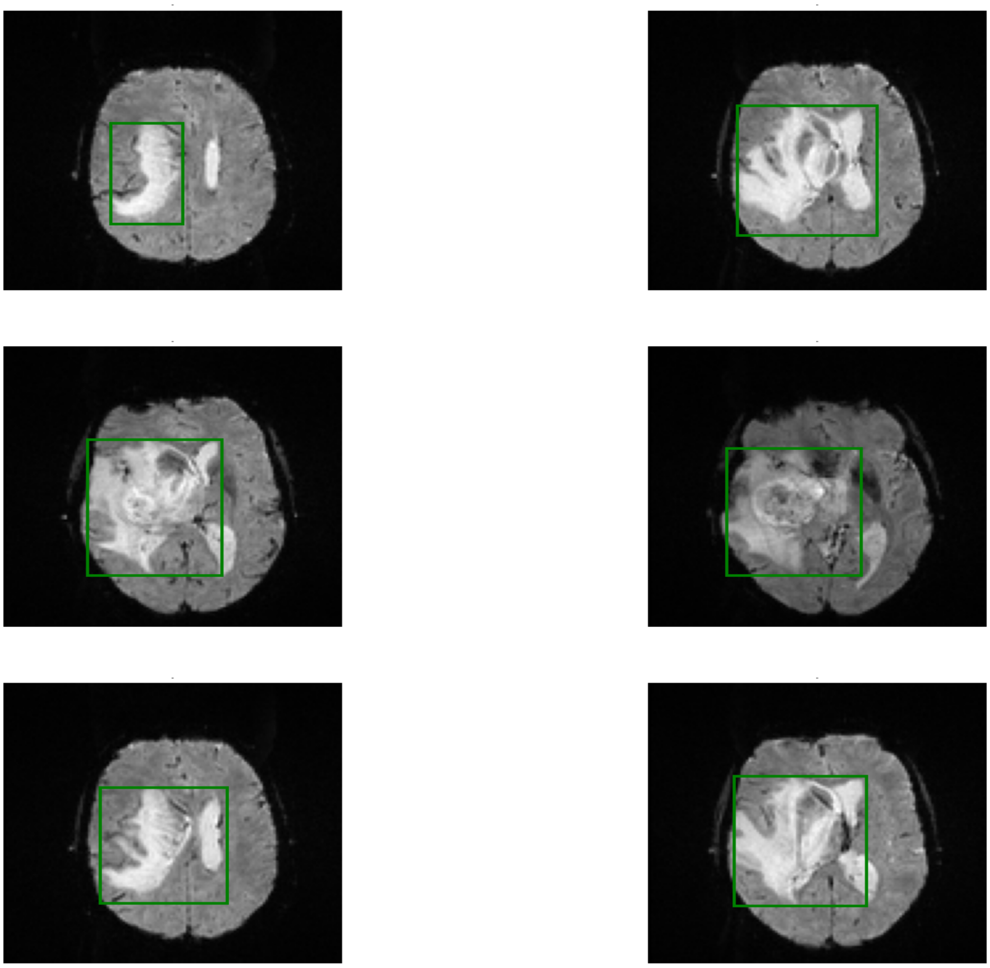
1. For my graduation project, I attempted to use my data on breast cancer to create boundary boxes around tumours using OpenCV. To get started, I utilized OpenCV to detect the contours of the images. You can find all of my code for this in this link to my notebook.[Bounding Box of the breast Cancer.ipynb]
    - After running my code, I was able to generate some output images that show the boundary boxes around the tumours. Here's an example:



2. For the second phase of my project, I downloaded the data from the Cancer Archive [Diffuse Glioma MRI (UCSF-PDGM)]. To get started, I created a visualization of 100 images from the dataset, which you can see in this notebook and the image below, [Automatic bounding boxes for brain cancer MRI images][Notebook].



- After doing some research, I came across an interesting approach for detecting tumours in the images and drawing a boundary box around them. I found a paper called "A Brain Tumor: Localization Using Bounding Box and Classification Using SVM " which provided some great insights. I also took a course on Data Camp called "Image Processing in Python." which helped me learn the necessary techniques.
- Using what I learned, I was able to apply the algorithm to images from the dataset. Here's an example of the output after running the algorithm:

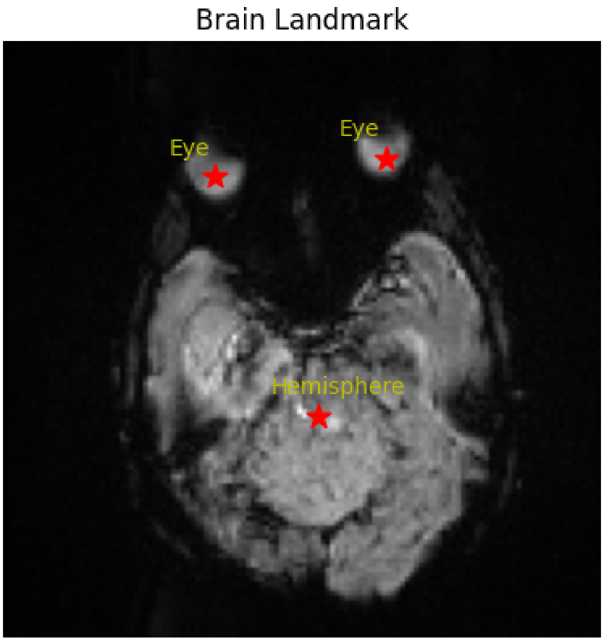


3. For the third phase of my project, I decided to download Niffler and extract PNG images from the DICOM files in the dataset. I did this by editing the config.json file and running the script. Once I had the images extracted, I started working on detecting landmarks in the images.

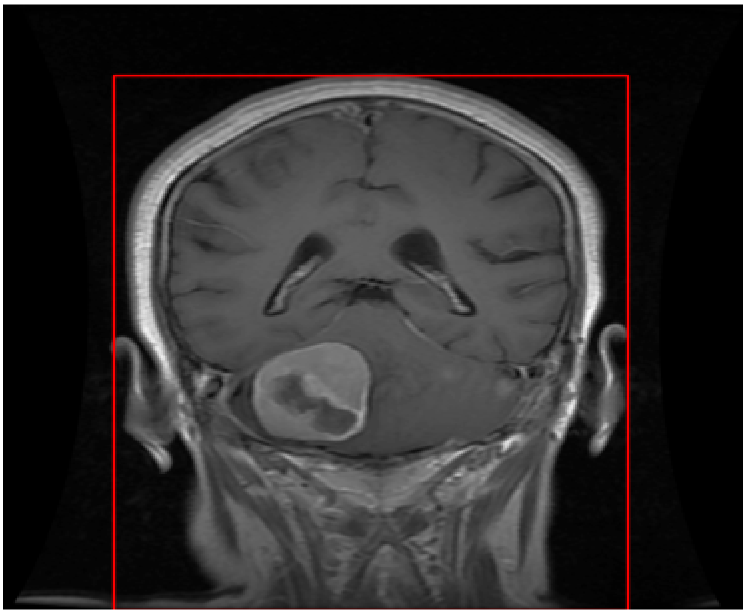
   And you can find all the code in this notebook, Automatic detection of human brain landmarks[Notebook].

- The plane I focused on for detecting landmarks was the axial plane. To do this, I applied a threshold to the image to generate contours. From there, I was able to extract the contours and draw a star on each landmark, labelling it with its corresponding name above it.

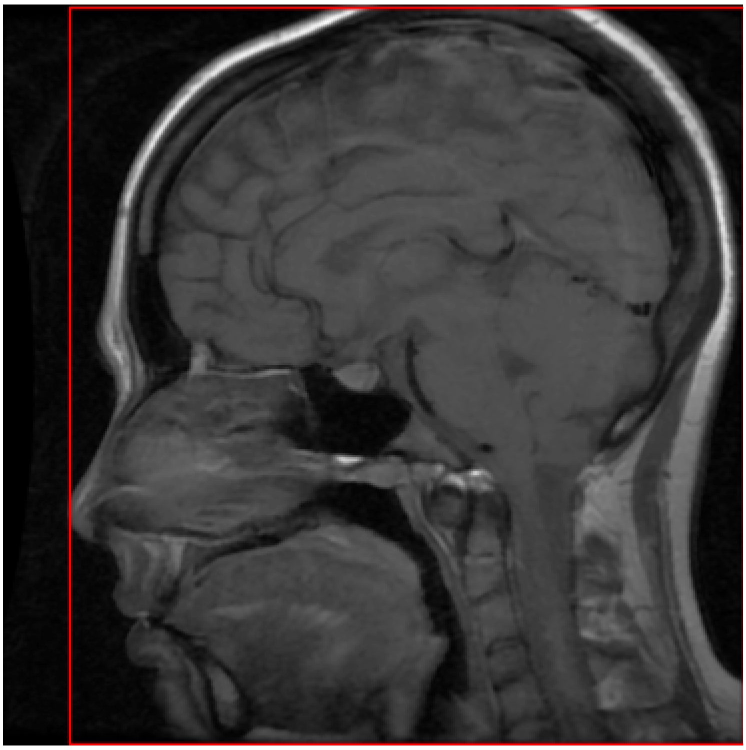
Here's an example of what the output looks like:



- As part of my project, I wanted to apply an example of the protocol we're interested in, and I found online images with the coronal and sagittal planes and applied two different protocols to them.
- For the first image, I applied the "Left ear to Right ear extent" protocol. Here's an example of what the output looks like:



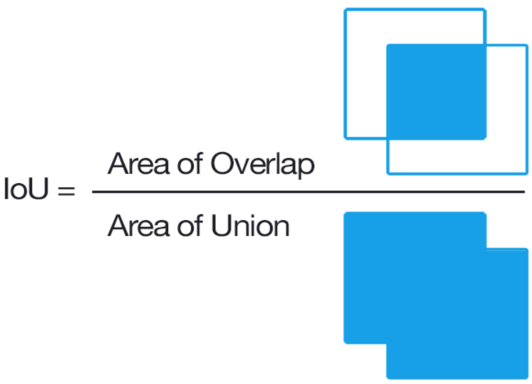
- For the second image, I applied the "Top of the head to ~C2/3 vertebral bodies" protocol. Here's an example of what the output looks like:



- Moving forward, my plan for future implementation is to detect landmarks on both the coronal and sagittal planes. This will help me to fully automate the process of applying our protocol to images with different orientations.
- By detecting landmarks in both planes, I will be able to accurately determine the location and orientation of the image and apply the appropriate protocol automatically. This will save time and ensure consistency in our analysis of different images.

- **Binary Compliance Score:**

  To assess the overall quality and compliance of our MRI images, we can use the IoU (Intersection over Union) function to compare the actual boundary box with the one generated by our algorithm. The IoU measures the overlap between the two bounding boxes as a ratio of their areas, providing a metric for evaluating the accuracy of our algorithm.



```python
def get_iou(bb1, bb2):
    """
    Calculate the Intersection over Union (IoU) of two bounding boxes.
    Parameters
    ----------
    bb1 : dict
        Keys: {'x1', 'x2', 'y1', 'y2'}
        The (x1, y1) position is at the top left corner,
        the (x2, y2) position is at the bottom right corner
    bb2 : dict
        Keys: {'x1', 'x2', 'y1', 'y2'}
        The (x, y) position is at the top left corner,
        the (x2, y2) position is at the bottom right corner
    Returns
    -------
    float
        in [0, 1]
    """
    # determine the coordinates of the intersection rectangle
    x_left = max(bb1['x1'], bb2['x1'])
    y_top = max(bb1['y1'], bb2['y1'])
    x_right = min(bb1['x2'], bb2['x2'])
    y_bottom = min(bb1['y2'], bb2['y2'])

    if x_right < x_left or y_bottom < y_top:
        return 0.0
    # The intersection of two axis-aligned bounding boxes is always an
    # axis-aligned bounding box
    intersection_area = (x_right - x_left) * (y_bottom - y_top)

    # compute the area of both AABBs
    bb1_area = (bb1['x2'] - bb1['x1']) * (bb1['y2'] - bb1['y1'])
    bb2_area = (bb2['x2'] - bb2['x1']) * (bb2['y2'] - bb2['y1'])

    # compute the intersection over the union
    iou = intersection_area / float(bb1_area + bb2_area - intersection_area)
    return iou
```

- I Also found a very interesting paper that proposes a method for identifying anatomical point landmarks in brain image scans that can be used to initialize brain image processing algorithms.
- The proposed method uses an empirical training procedure that can locate these landmarks with high precision, regardless of image resolution or MRI weighting.
- The authors demonstrate the effectiveness of their approach using a Java GUI application called LONI ICE, which can determine the MRI weighting of brain scans and locate features in T1-weighted and T2-weighted scans
- And here is the link to the paper, Automatic Localization of Anatomical Point Landmarks for Brain Image Processing Algorithms.

11. Why me

As a biomedical engineering student with a background in medical imaging and machine learning, I believe I would be a strong candidate for the Google Summer of Code project that you described. I have experience working on projects involving medical imaging, such as my graduation project, which was a web-based DICOM viewer for automatic breast cancer segmentation in ultrasound images.

During my graduation project, I worked closely with a team of medical professionals and computer scientists to develop and test our algorithm for automatic breast cancer segmentation. We used a combination of image processing techniques and machine learning algorithms to identify and segment cancerous tissue in ultrasound images accurately and here is a link to our project paper. Final-Report.pdf

In addition to my academic background, I have also worked on various personal projects related to medical imaging and machine learning, such as developing an image classification algorithm for identifying skin lesions in dermatology images during my internship at Omdena. I am confident that my skills and experience would enable me to contribute meaningfully to the Google Summer of Code project you have proposed.

My motivation for pursuing this project is to contribute to the development of innovative solutions that can improve medical imaging analysis. As someone with a strong interest in data science and the medical field, I believe this project is an excellent opportunity to apply my skills and gain valuable experience. Participating in this program will not only help me develop new skills and expand my knowledge but also make a significant contribution to the medical community. It will also be a strong addition to my resume and help me become a competitive candidate for future academic and professional opportunities as I want to apply to master’s programs in data science after my graduation.