Capstone Project Concept Note and Implementation Plan

Project Title: CervixCancerSegmentation

Automatic Segmentation of Cervical Lesions Using Deep Learning.

Team Members

1. Juma Rubea
2. Meman Salam
3. Soumana Dama
4. Plensia Lukosi

Concept Note

1. Project Overview

This project aims to develop a deep learning model for the segmentation of cervical lesions in medical images, with a particular focus on cervical cancer. Aligned with **Sustainable Development Goal (SDG) 3: Good health and well-being**, this project contributes to improved screening and early diagnosis of this disease, particularly in resource-limited regions.

The problem identified is the lack of affordable and automated solutions to detect cervical lesions accurately. By using an advanced segmentation model, this project has the potential to reduce mortality rates by enabling faster and more accurate diagnosis, thus promoting early and effective treatment.

2. Objectives

- Develop a deep learning model capable of accurately segmenting cervical lesions in colposcopic images.
- Reduce the impact of artifacts (blood, mucus, lighting) on model accuracy.
- Provide a robust solution suitable for low-resource clinical environments.
- Contribute to the early detection and reduction of cervical cancer mortality rates.

3. Background

Cervical cancer remains a major cause of female mortality, particularly in developing countries. Traditional screening approaches, such as cervical smear or manual colposcopy, require specialized expertise that is often unavailable in these areas.

Previous initiatives have incorporated deep learning models such as U-Net for medical image segmentation. However, they face limitations related to noise management and the availability of annotated data. By adopting advanced models such as **Mask R-CNN** or **U-Net++**, this project aims to overcome these challenges while offering a more accessible solution.

4. Methodology

- **Techniques used** :
    - Image segmentation model based on **U-Net++** or **DeepLabv3+** for its accuracy in noisy environments.
    - Data pre-processing including resizing, normalizing, and augmenting (rotation, brightness).
- **Tools and frameworks** :
    - **Frameworks** : TensorFlow, PyTorch.
    - **Environment** : Google Colab or GPU-equipped servers for training.
- **Process** :

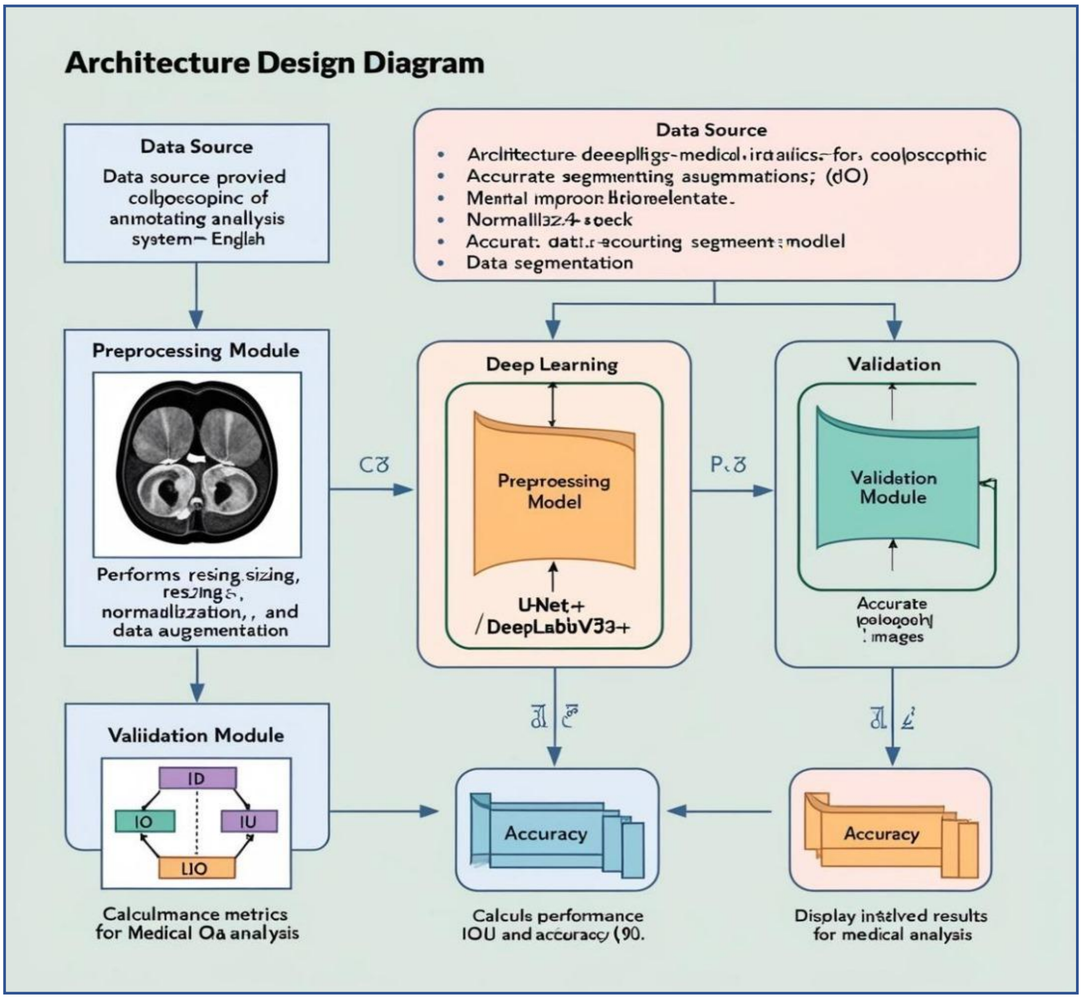
1. Data pre-processing to standardize sizes and reduce noise.
2. Training on an annotated dataset.
3. Validation using metrics such as IoU (Intersection over Union) and **Dice Coefficient**.

5. Architecture Design Diagram

Description of the components :

1. **Data source** : Provides annotated colposcopic images.
2. **Preprocessing module** : Applies data resizing, normalizing, and augmenting.
3. **Deep Learning Model** : Performs precise segmentation using U-Net++ or DeepLabv3+.
4. **Validation module** : Calculates performance via standard metrics (IoU, accuracy).
5. **User Interface** : Displays results for medical analysis.

Architecture Design Diagram illustration:



6. Data Sources

The data come from a private set of colposcopic images, annotated in four classes: lesions, blood, mucus, and light. These annotations are in **JSON** format and are used to generate segmentation masks for model training. Careful pre-processing is required to ensure that critical details are not lost during resizing or normalization.

7. Literature Review

Existing research shows that models like **U-Net** and **Mask R-CNN** offer high accuracy for medical image segmentation. Mehboob et al. (2022) demonstrated the effectiveness of CNNs with 85% accuracy, while Saini et al. (2020) validated the robustness of **ColpoNet** in noisy environments. This project builds on this work while proposing solutions tailored to the specific challenges of cervical images.

Implementation Plan

1. Technology Stack

- **Programming language:**
  - Python
  - Java-script
- **Libraries**
  - Panda
  - NumPy
  - Matplotlib/Seaborn
  - scikit learn
  - Pycocotools
  - cv2
- **Frameworks**
  - TensorFlow
  - detectron2
  - Django
- **Hardware components.**
  - GPU
- **Other tools:**
  - Google Colab
  - Docker/hugging face

2. Timeline

Data collected from Private Source with private contract due to sensitive nature of datasets

A. Data Annotation

Timeframe**:** 4-5 Days

Tasks:

Task 1: **Prepare Annotation Guidelines**

Define clear annotation guidelines for lesion, mucus, blood, and light.

Create training material for annotators to ensure consistency.

Task 2: **Annotate Images**

Annotators mark regions with lesions, mucus, blood, and light.

Use tools Coco Annotator to annotate images with polygons or masks.

Team Member Distribution:

Juma Rubea Task 1 (Annotation Guidelines)

Meman Awad: Task 2 (Annotation)

Dama Soumana: Task 2 (Annotation)

Sia: Task 2 (Annotation)

B. Data Preprocessing and Augmentation

Timeframe: Day 6

Tasks:

Task 1: **Preprocess Images (Day 6)**

Normalize images, resize, and perform any necessary transformations.

Task 2: **Data Augmentation (Day 6)**

Apply augmentation techniques (rotation, flipping, zooming, etc.) to improve model robustness.

Ensure that augmentation does not distort annotated regions.

Team Member Distribution:

Member A: Task 1 (Preprocessing)

Member B: Task 2 (Augmentation)

Member C: Task 2 (Augmentation)

Member D: Task 2 (Augmentation)

C. Model Development

Timeframe: Day 7-8

Tasks:

Task 1: **Set Up Mask R-CNN Framework (Day 7)**

Set up Mask R-CNN model using a framework in TensorFlow.

Modify the architecture for multi-class segmentation (lesion, mucus, blood, light).

Task 2: **Model Training (Day 7)**

Start training the Mask R-CNN model using annotated data.

Monitor loss and performance during training.

Task 3: **Model Optimization (Day 9)**

Tune hyperparameters, use learning rate scheduling, and experiment with different backbones (e.g., ResNet, VGG).

Team Member Distribution:

Member A: Task 1 (Model Setup)

Member B: Task 2 (Training)

Member C: Task 3 (Optimization)

Member D: Task 2 (Training)

D. Model Evaluation

Timeframe: Day 10

Tasks:

Task 1: **Evaluate Model Performance (Day 10)**

Evaluate the model on test data using metrics like Intersection over Union (IoU), Dice score, and pixel-wise accuracy.

Compare performance for each class (lesion, mucus, blood, light).

Task 2: **Fine-Tuning (Day 10)**

Fine-tune the model based on evaluation results to improve class-specific segmentation.

Team Member Distribution:

Member A: Task 1 (Evaluation)

Member B: Task 1 (Evaluation)

Member C: Task 2 (Fine-tuning)

Member D: Task 2 (Fine-tuning)

E. Web Application Development

Timeframe: Day 11 -13

Tasks:

Task 1: **Design Web App Interface (Day 11)**

Create UI mockups for the web application.

Define functionality for uploading images, displaying segmentation results, and visualizing segmentation maps.

Task 2: **Web App Development (Day 12)**

Develop the front-end using frameworks like React or Vue.js.

Set up the back-end (Flask or Django) to handle image processing and visualization.

Task 3: **Deploy Web Application (Day 13)**

Host the web app on a cloud service (e.g., AWS, Heroku).

Ensure that the web app can accept user input and return segmentation results.

Team Member Distribution:

Member A: Task 1 (UI/UX Design)

Member B: Task 2 (Web App Development - Frontend)

Member C: Task 2 (Web App Development - Backend)

Member D: Task 3 (Deployment)

F. Model Integration and Final Testing

Timeframe: Day 14 -15

Tasks:

Task 1: **Integrate Model into Web App (Day 14)**

Integrate the trained Mask R-CNN model with the web app for real-time segmentation.

Ensure smooth communication between the model and the web interface.

Task 2: **Final Testing & Debugging (Day 15)**

Perform end-to-end testing of the application (model predictions, image upload, display results).

Fix any bugs or performance issues.

Team Member Distribution:

Member A: Task 1 (Integration)

Member B: Task 1 (Integration)

Member C: Task 2 (Testing)

Member D: Task 2 (Testing)

2. Milestones

Table below visualize key milestones for our projects

No:	Key Milestones	Expected time
1	Image Annotation Completed	Day 5
2	Data Preprocessing and Augmentation Completed	Day 6
3	Mask R-CNN Model Development and Training Completed	Day 7
4	Model Evaluation and Fine-tuning Completed	Day 9
5	Web Application Developed and Deployed	Day 13
6	Final Integration and Testing Completed	Day 15

Gantt Chart

Day/Task	Data Annotation	Preprocessing & Augmentation	Model Development	Model Evaluation & Fine-tuning	Web Application Development	Integration & Final Testing
Day 1	X					
Day 2	X					
Day 3	X					
Day 4	X	X				
Day 5	X	X				
Day 6		X	X			X
Day 7			X			
Day 8			X	X		
Day 9				X		
Day 10				X	X	X
Day 11					X	X
Day 12					X	X
Day 13					X	X
Day 14						X
Day 15						X

Table task Distribution

Task	**Member A**	**Member B**	**Member C**	**Member D**
Data Annotation	X	X	X	X
Data Preprocessing and Augmentation	X	X	X	X
Mask R-CNN Model Setup (Architecture)	X			
Model Training and optimization	X	X		
Model Evaluation and Fine-tuning			X	X
	X	X	X	X
Web App UI/UX Design	X	X	X	X
Web App Frontend Development	X			
Web App Backend Development		X		X
Web App Deployment	X	X		
Model Integration with Web App	X	X		
Final Testing & Debugging			X	X

4. Challenges and Mitigations

- Data-set availability

  Challenge: Limited access to diverse, high-quality datasets specific to cervical cancer segmentation.

  Mitigation: Augment the dataset using synthetic data generation techniques to enhance diversity.

- Noises in data-set

  Challenge: Presence of artifacts like blood, mucus, and lighting variations in colposcopic images can impact model performance.

  Mitigation: Implement preprocessing steps, such as normalization and filtering, to reduce noise. Enhance the model's robustness through augmentation techniques like brightness adjustment and noise injection during training.

- Access to GPUs

  Challenge: Limited availability of high-performance hardware, such as GPUs, required for training deep learning models efficiently.

  Mitigation: Utilize cloud-based platforms like Google Colab, which offer on-demand GPU access.

- Labeling the data.

  Challenge: Labeling medical images with pixel-level precision is time-consuming and requires knowledge.

  Mitigation: Use semi-automated labeling tools or pre-trained models like SAM (Segment Anything Model) to assist in annotation. Reading thoroughly and explore diverse images to gain the knowledge needed to ensure accuracy and speed up the labeling process.

5. Ethical Considerations

Data Privacy: Ensure patient data is securely stored to protect privacy.

Bias:

Avoid bias that could impact diagnosis accuracy across populations.

Clinical Impact:

The model supports but does not replace expert medical judgment. Doctors should verify all outputs.

6. References

- https://thesai.org/Downloads/Volume13No9/Paper_104-Deep_Learning_based_Cervical_Cancer_Classification.pdf

- https://github.com/jsbroks/coco-annotator

- https://ieeexplore.ieee.org/document/9751891

Tools references:

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. Retrieved from https://arxiv.org/abs/1703.06870 SAM Official Documentation. Retrieved from https://segment-anything.com COCO Dataset and API Documentation. Retrieved from https://cocodataset.org Albumentations Documentation. Retrieved from https://albumentations.ai i want refrences for similar projects