**Capstone Project Concept Note and Implementation Plan**

**Project Title: Cervical Cancer Segmentation**

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 is dedicated to developing an advanced deep learning model for the **segmentation of cervical lesions** in medical images, with a particular emphasis on cervical cancer detection. By leveraging the power of **Mask R-CNN**, a cutting-edge model for image segmentation, this project aligns with **Sustainable Development Goal (SDG) 3: Good Health and Well-being**, contributing to better health outcomes through enhanced screening and early diagnosis.

One of the critical challenges in cervical cancer diagnosis is the lack of affordable, accurate, and automated solutions for detecting cervical lesions at an early stage. Our model, built on **Mask R-CNN**, has the potential to overcome this barrier by providing a more efficient, precise, and scalable method for identifying lesions. With faster and more reliable detection, we aim to reduce mortality rates and promote timely, life-saving treatment—ultimately improving the quality of healthcare and outcomes for those affected by cervical cancer.

1. **Objectives**

The main objectives of this project were to:

* 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.

1. **Background**

Cervical cancer remains one of the leading causes of cancer-related mortality among women worldwide, as highlighted by the World Health Organization (**WHO**). It is a significant public health challenge, especially in low- and middle-income countries, where access to early diagnostic tools and preventive measures is limited.

Despite advances in screening techniques such as **Pap smears and HPV testing**, a substantial number of cases are still detected at advanced stages, leading to a high mortality rate. Early and accurate detection of cervical lesions can greatly improve patient outcomes and reduce mortality.

The use of **colposcopic** imaging in cervical cancer screening has become increasingly widespread due to its ability to provide detailed visual assessments of the cervix. However, analyzing colposcopic images can be complex and subjective, often requiring significant expertise. Lesions are often obscured by the presence of mucus, blood, and lighting artifacts, making manual interpretation challenging.

To address these challenges, our project aims to develop an advanced image segmentation system using **Mask R-CNN**, a state-of-the-art deep learning model. This model is designed to identify and **segment cervical lesions** in colposcopic images with high precision. In addition to lesions, the model will also identify and segment mucus, blood, and light artifacts, enabling a more comprehensive and accurate analysis of the images.

By leveraging this technology, we hope to provide a reliable and automated tool for early detection of cervical cancer, ultimately contributing to a reduction in the global burden of this deadly disease.

**4. Methodology**

**Techniques Used**

* **Image Segmentation Model:**  
  Our project employs **U-Net++** or **DeepLabv3+** as the primary segmentation model due to their proven accuracy in handling noisy environments and complex boundaries. These models are well-suited for medical imaging tasks where precision is critical.
* **Data Pre-processing:**  
  Effective pre-processing techniques are applied to enhance the quality of input data:
  + Resizing all images to a standardized dimension for consistency during model training.
  + Normalizing pixel values to improve convergence during training.
  + Augmenting the dataset with operations like rotation, brightness adjustment, flipping, and scaling to increase model robustness and prevent overfitting.

**Tools and Frameworks**

* **Frameworks:**
  + TensorFlow serve as the core framework for model development and training. its flexibility and support for state-of-the-art neural network architectures make it ideal for this project.
* **Environment:**
  + Training and experimentation are conducted on platforms like **Google Colab**, which provides free access to **GPUs**, or dedicated **GPU-equipped servers** for faster computation and scalability.

**Process**

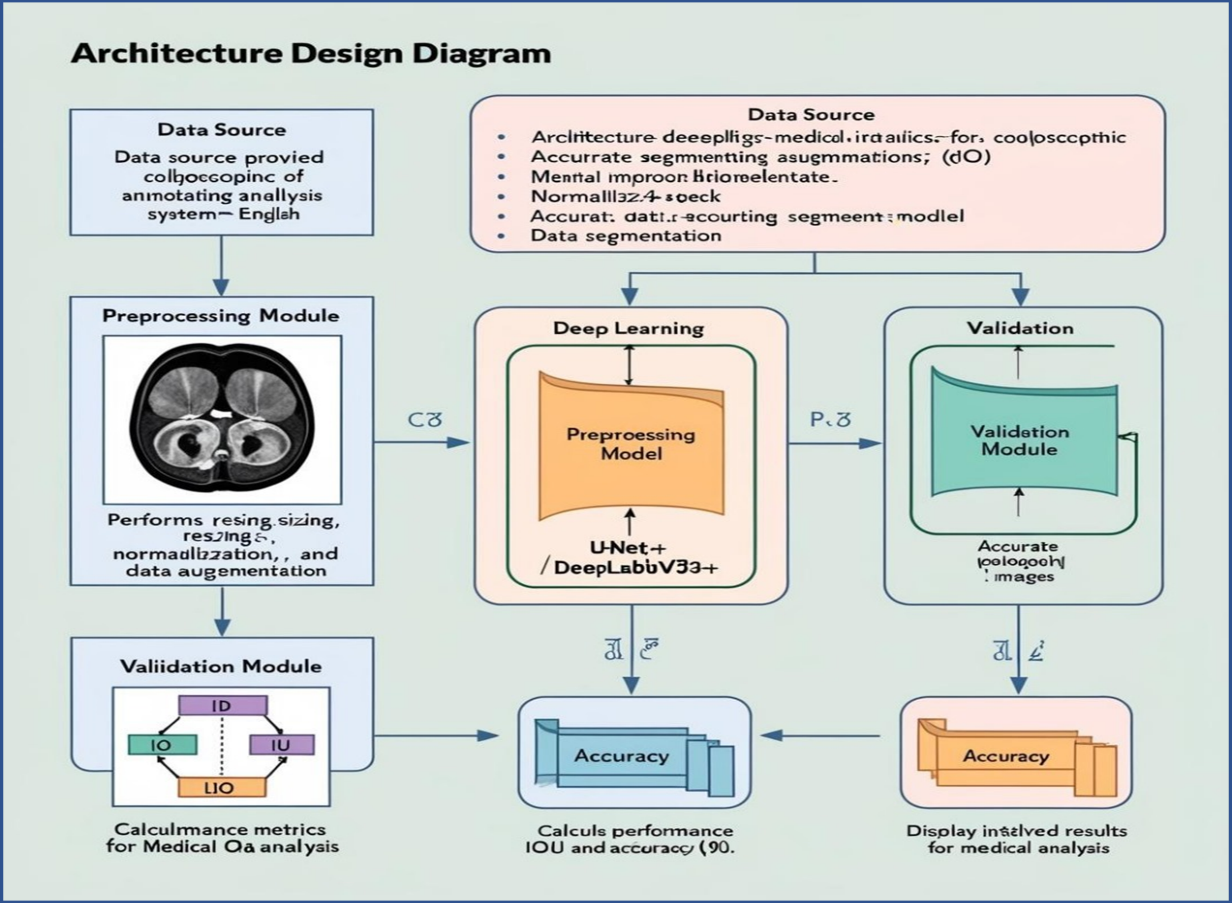
1. **Data Pre-processing:**
   * Images are pre-processed to address variations in size and to reduce noise caused by artifacts such as lighting, blood, and mucus. Augmented datasets improve the model's ability to generalize across unseen data.
2. **Model Training:**
   * The model is trained using a labeled dataset of colposcopic images where regions of interest (lesions, mucus, blood, and lighting artifacts) are annotated. This allows the model to learn pixel-level segmentation.
3. **Validation and Evaluation:**
   * The model's performance is validated using quantitative metrics such as:
     + **IoU (Intersection over Union):** Measures the overlap between predicted and ground truth segmentations.
     + **Dice Coefficient:** Evaluates the similarity of the predicted segmentation to the ground truth, especially in scenarios where the region of interest is small.
   * Iterative optimization is conducted to improve accuracy and reduce false positives or negatives.

**5.** **Architecture Design Diagram**

**Description of the Components**

1. **Data Source**  
   The data source consists of a repository of annotated colposcopic images that serve as the foundation for model training and evaluation. These images are pre-labeled with regions of interest, such as lesions, mucus, blood, and light artifacts, ensuring the model can learn from accurate and clinically relevant examples.
2. **Preprocessing Module**  
   This module is responsible for preparing the data before feeding it into the deep learning model. Key tasks include:
   * **Resizing:** Standardizing the dimensions of all images to ensure compatibility with the model.
   * **Normalizing:** Scaling pixel intensity values to a uniform range to improve model stability and performance.
   * **Augmenting:** Generating variations of the original dataset through transformations like rotation, brightness adjustment, flipping, and scaling to enhance model robustness and reduce overfitting.
3. **Deep Learning Model**  
   The core component of the system, leveraging advanced architectures **Mask RCNN** for precise image segmentation. This model was specifically chosen for its ability to handle noisy data and capture fine details, making them ideal for segmenting lesions and other features in colposcopic images.
4. **Validation Module**  
   This module evaluates the model's performance using standard metrics, ensuring reliable results:
   * **IoU (Intersection over Union):** Quantifies the overlap between the predicted segmentation and the ground truth.
   * **Dice Coefficient:** Measures the similarity between the predicted and actual segmentations, particularly effective for small or complex regions.
   * **Accuracy Metrics:** Evaluate the overall correctness of the segmentation model.
5. **User Interface (UI)**  
   The UI is designed for medical professionals to interact with the system seamlessly. It displays the segmentation results overlaid on the original colposcopic images, highlighting regions of interest for analysis. Additional features may include zoom functionality, lesion measurements, and options for exporting results, making it a practical tool for clinical applications.

**Architecture Design Diagram illustration:**

****

**6. Data Sources**

The data come from a private set of colposcopic images, then annotated in four classes: lesions, blood, mucus, and light. These annotations are in COCO **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**

Recent advancements in deep learning have shown significant potential in improving cervical cancer diagnosis through image segmentation. In their study, Wang et al. (2022) emphasized the critical role of segmentation in isolating cervical lesions and addressing challenges posed by artifacts such as mucus, blood, and lighting variations in colposcopic images. Similar studies by Zhou et al. (2018) on U-Net++ highlight its capability to capture intricate boundaries through nested architectures and dense skip connections, making it particularly effective for complex medical image segmentation tasks. Additionally, Mask R-CNN has been recognized for its robust instance segmentation capabilities, as demonstrated by He et al. (2017), enabling precise identification and localization of individual regions of interest within medical images.

**8. Implementation Plan**

### 1. Technology Stack

**Programming Languages**

* **Python**: The primary language used for model development, data preprocessing, and integration with AI frameworks.
* **JavaScript**: Utilized for developing the user interface and web-based visualization tools.

**Libraries**

* **Pandas**: For efficient data manipulation and handling tabular datasets such as image metadata and annotations.
* **NumPy**: Facilitates numerical operations and array computations required during model training and preprocessing.
* **Matplotlib/Seaborn**: Provides tools for visualizing training results, performance metrics, and data distributions.
* **Scikit-learn**: Used for dataset splitting, statistical analyses, and calculating evaluation metrics.
* **Pycocotools**: Aids in handling datasets formatted in the COCO standard, particularly useful for annotation and model evaluation.
* **OpenCV**: Used for image preprocessing tasks such as resizing, augmentation, and noise reduction.

**Frameworks**

* **TensorFlow**: Serves as the foundational deep learning framework for training segmentation models like U-Net++.
* **Detectron2**: A PyTorch-based framework specifically used for implementing Mask R-CNN for instance segmentation tasks.
* **Django**: A high-level Python web framework used for building the user interface and deploying the application.

**Hardware Components**

* **GPU**: Essential for accelerating deep learning model training and inference, significantly reducing computation time.

**Other Tools**

* **Google Colab**: Provides an accessible environment for model training with GPU support, facilitating experimentation and collaboration.
* **Docker**: Ensures reproducibility and seamless deployment by containerizing the application.
* **Hugging Face**: Utilized for leveraging pre-trained models and sharing fine-tuned models.

**2. Timeline**

**A. Data Annotation Process**

Timeframe:4-5 Days

**Tasks:**

1. **Prepare Annotation Guidelines**
   * Define comprehensive guidelines for annotating key regions, including lesions, mucus, blood, and lighting artifacts.
   * Develop training materials and provide instructions to annotators to ensure consistency and accuracy in the annotation process.
2. **Annotate Images**
   * Annotators mark regions of interest (lesions, mucus, blood, and light) on the images.
   * Utilize tools like **COCO Annotator** to create precise annotations in the form of polygons or masks.

**Team Member Distribution**

* **Juma Rubea**: Preparing annotation guidelines (Task 1).
* **Meman Awad**: Assigned to image annotation (Task 2).
* **Dama Soumana**: Assigned to image annotation (Task 2).
* **Plensia Lukosi**: Assigned to image annotation (Task 2).

**B. Data Preprocessing and Augmentation**

Timeframe: Day 6

**Tasks:**

1. Preprocess Images
   * Normalize image pixel values to a uniform scale for improved model convergence.
   * Resize images to a standard dimension to ensure compatibility with the model architecture.
   * Apply necessary transformations to prepare the images for training.
2. Data Augmentation
   * Enhance the dataset by applying augmentation techniques such as rotation, flipping, zooming, and brightness adjustment to improve model robustness and reduce overfitting.
   * Ensure that augmentation techniques do not distort the annotated regions of interest.

**Team Member Distribution**

* **Juma Rubea**: Task 1 (Preprocessing)
* **Meman Awad**: Task 2 (Augmentation)
* **Dama Soumana**: Task 2 (Augmentation)
* **Plensia Lukosi**: Task 2 (Augmentation)

**C. Model Development**

Timeframe: Day 7-9

**Tasks**:

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

Set up the Mask R-CNN model using 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:**

* **Juma Rubea:** Task 1 (Model Setup)
* **Meman Salam:** Task 2 (Training)
* **Soumana Dama:** Task 3 (Optimization)
* **Plensia Lukosi:** 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:**

* **Juma Rubea:** Task 1 (Evaluation)
* **Meman Salam:** Task 1 (Evaluation)
* **Soumana Dama:** Task 2 (Fine-Tuning)
* **Plensia Lukosi:** 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:**

* **Juma Rubea:** Task 1 (Web App Development - Frontend)
* **Meman Salam:** Task (Web App Development - Backend)
* **Soumana Dama:** Task 2 (Web App Development - Frontend)
* **Plensia Lukosi:** Task 3 (UI/UX Design)

**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:

* **Juma Rubea:** Task 1 (Integration)
* **Meman Salam:** Task 1 (Integration)
* **Soumana Dama:** Task 2 (Integration)
* **Plensia Lukosi:** 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** | **Juma** | **Meman** | **Soumana** | **Plensia** |
| 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 |
| Web App UI/UX Design | X | X | X | X |
| Web App Frontend Development | X |  |  |  |
| Web App Backend Development |  | X |  | X |
| Web App Deployment and Model Integration | X | X |  |  |
| Final Testing & Debugging |  |  | X | X |

1. **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.

1. **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.

1. **References**
2. Wang, L., Zhang, Y., Li, X., et al. (2022). AI-Assisted Cervical Cancer Screening via Colposcopic Image Analysis. *Frontiers in Oncology.* [Link to article](https://www.frontiersin.org/journals/oncology/articles/10.3389/fonc.2022.952847/full)
3. Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). UNet++: A Nested U-Net Architecture for Medical Image Segmentation. *arXiv preprint arXiv:1807.10165.*
4. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2961-2969
5. https://thesai.org/Downloads/Volume13No9/Paper\_104-Deep\_Learning\_based\_Cervical\_Cancer\_Classification.pdf
6. <https://github.com/jsbroks/coco-annotator>
7. <https://ieeexplore.ieee.org/document/9751891>
8. **SAM Official Documentation.** (n.d.). Retrieved from <https://segment-anything.com>
9. **COCO Dataset and API Documentation.** (n.d.). Retrieved from <https://cocodataset.org>