Machine Learning Project Documentation

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**

Model Refinement

1. Overview

The model refinement phase focused on enhancing the performance of the Mask R-CNN model used for cervical lesion segmentation. This involved iterative improvements based on evaluation metrics, hyperparameter tuning to address challenges like noise in images and class-specific segmentation accuracy, particularly for mucus and light artifacts.

2. Model Evaluation

Initial evaluations showed strong performance in lesion segmentation (IoU: 0.85) but moderate results for light (IoU: 0.76) and weaker results for mucus (IoU: 0.69). Visual inspections revealed alignment issues, especially in differentiating mucus from lesions, highlighting areas for improvement.

3. Refinement Techniques

Key techniques applied include:

- **Data Augmentation**: Enhanced image robustness with rotations, flips, brightness adjustments, and noise injection.
- **Feature Optimization**: Improved annotations to better differentiate lesion boundaries from artifacts.

4. Hyperparameter Tuning

Hyperparameters such as learning rate, batch size, and max iterations were adjusted:

- Learning rate reduced to **0.0002** for better convergence.
- Batch size increased to **4** to stabilize training.
- Max iterations extended to **5000**, allowing the model to generalize further.

5. Feature Selection

Feature importance was revisited to emphasize high-value regions of interest (lesions). Artifact removal techniques during preprocessing minimized noise impact on training data, leading to more precise segmentation.

Test Submission

1. Overview

The test submission phase involves deploying and evaluating a Mask R-CNN model for segmenting lesions in colposcopic images, part of a cervical cancer segmentation project. The process includes preparing a robust test dataset, applying the trained model, evaluating performance metrics, and deploying the model in a production-like environment.

2. Data Preparation for Testing

Key steps in preparing the test dataset included:

- **Image Quality Filtering**: Removed blurry and corrupted images to enhance annotation accuracy.
- **Noise Management**: Addressed noise issues, including light reflections and mucus, using preprocessing techniques.
- **Annotation Validation**: Used COCO Annotator for marking lesions as primary regions of interest (ROI) while labeling light and mucus as distinct classes.
- **Standardization**: Resized images to a uniform dimension and ensured annotation paths aligned with file locations in the Google Drive environment.

3. Model Application

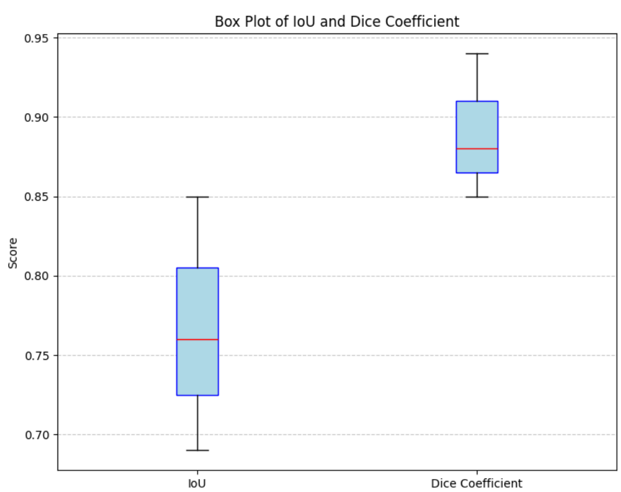
The trained Mask R-CNN model was applied to the test dataset using a workflow that involved loading the model weights and configuration, running predictions, and visualizing results:

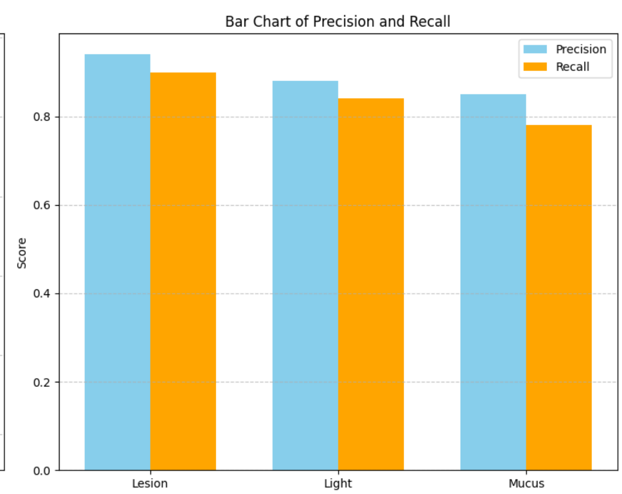
```python
import cv2
from model import Predictor

# Initialize the model and predictor
model_path = "final_model.pth"
config_path = "config.yaml"

predictor = Predictor(model_path, config_path)

# Load and predict on a sample image
image = cv2.imread("test_image.jpg")
output = predictor.predict(image)

# Visualization
for bbox, mask, label, score in zip(output['bboxes'], output['masks'], output['labels'], output['scores']):
    cv2.rectangle(image, (bbox[0], bbox[1]), (bbox[2], bbox[3]), (255, 0, 0), 2)
    cv2.putText(image, f"{label}: {score:.2f}", (bbox[0], bbox[1] - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2)

cv2.imshow("Output Image", image)
cv2.waitKey(0)
```

4. Test Metrics

The model's performance on the test dataset was evaluated using the following metrics:

- **IoU (Intersection over Union):** Assessed the overlap between predicted and ground truth masks.
- **Dice Score:** Measured segmentation accuracy, especially useful for medical image analysis.
- **Per-Class Metrics:** Evaluated precision, recall, and scores for each lesion class in the test images





The model demonstrated consistent performance across training and test datasets, balancing its predictions effectively.

5. Model Deployment

The trained model was deployed to a Hugging Face Space, enabling public access for testing and inference. Key deployment steps included:

- **Containerization:** Used Docker to package the model and dependencies into a lightweight, deployable container.
- **Backend Development:** Implemented a FastAPI backend to handle model inference requests and provide API endpoints for real-time predictions.
- **Integration with Hugging Face:** Leveraged Hugging Face Space for hosting the deployed application, providing an interactive user interface for testing. This is url to the deployed running container (https://huggingface.co/spaces/JumaRubea/model_visualization)

6. Code Implementation

Relevant snippets for prediction and deployment:

```
from fastapi import FastAPI, File, UploadFile
import cv2
from model import Predictor

app = FastAPI()
predictor = Predictor(model_path="model.pth", config_path="config.yaml")

@app.post("/predict")
async def predict(file: UploadFile = File(...)):
    image = cv2.imread(file.file)
    output = predictor.predict(image)
    return {
        "scores": output['scores'],
        "labels": output['labels'],
        "bboxes": output['bboxes']
    }
```

Conclusion

The test submission phase validated the Mask R-CNN model's ability to segment lesions with high accuracy. Incorporating metrics like IoU, Dice Score, and class-specific performance confirmed the model's robustness. Deployment to Hugging Face Space, supported by Docker and FastAPI, provided a scalable and accessible platform for further testing and practical use. These efforts set the stage for integration into clinical workflows, enhancing diagnostic accuracy for cervical cancer detection.

References

Chena, J., & Massa, F. (2019). *facebookresearch*. Retrieved from GitHub: https://github.com/facebookresearch/maskrcnn-benchmark

He, K., Gkioxari, G., Dollár, P., & Girshick, R. B. (n.d.). *Computer Vision and Pattern Recognition: Mask R-CNN*. Retrieved from arxiv: https://arxiv.org/abs/1703.06870

Spaces. (n.d.). Retrieved from Hugging Face: https://huggingface.co/docs/hub/spaces

Tutorial-User Guide. (n.d.). Retrieved from FastAPI: https://fastapi.tiangolo.com/tutorial/