Introduction to Computer Vision (ECSE 415)

Assignment 4: Image Segmentation

Due Date: 11:59 PM, November 22, 2024

- Please submit your assignment solutions electronically via the myCourses assignment dropbox.
- Use Python code to implement all operations. You may utilize PyTorch, TensorFlow, OpenCV, Numpy, and Scikit-learn library functions for all parts of the assignment, or alternatively, you can write your own implementations.
- Students are expected to write their own code and complete the assignments independently. (Refer to the academic integrity guidelines at McGill Academic Integrity).
- Late submissions will incur a penalty of 10% per day.

Note: Members within each group will receive the same score for the assignment.

Submission Instructions

On myCourses:

- Submit a single Jupyter notebook with all code, visualizations, and explanations.
- Ensure your code is well-commented for clarity.
- Remember to run all Markdown cells so that the output images appear correctly.
- Input/output images must be displayed within the Jupyter notebook itself; do not submit them separately. Assume input images are located in the same directory as your code.
- Ensure the submitted code runs without errors. Clearly describe any specific requirements for executing the code in your notebook.
- If you use external libraries in your code, please specify their names and versions within the notebook.
- **Include your report as a separate PDF.**

On Kaggle:

- **Submit the predicted segmentation CSV file to the competition.**

Preliminaries

1. **Group Enrollment**: Enroll in a group of up to 2 people on myCourses to access the assignment files.

2. **Join the Kaggle Competition**: Join the competition through this link. (If you don't have a Kaggle account, you'll need to create one first.)
3. **Team Naming**: Name your Kaggle team to match your group number, e.g., "Group 3".

Objective

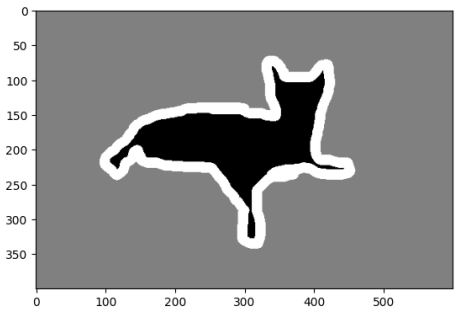
The goal of this assignment is to implement a semantic segmentation model using the Oxford-IIIT Pets Dataset, evaluate its performance, and gain hands-on experience with deep learning techniques for image segmentation.

Dataset Description

Image Files: The Oxford-IIIT Pet Dataset is a 37-category pet dataset with roughly 200 images for each class. Images are in JPEG format with varied resolutions and contain a single pet per image.

Segmentation Masks: Each image has a pixel-wise mask in PNG format with three classes (trimap format):

- **1 (Foreground)**: Pet region (black).
- **2 (Background)**: Area outside the pet (gray).
- **3 (Boundary)**: Small boundary around the pet (white).



Dataset Download: **Download the dataset from the Kaggle competition** and unzip it into your working directory. The structure should be as follows:

- Oxford
  - train
    - images: 5,390 training images
    - masks: 5,390 segmentation masks (trimap format)
  - test
    - images: 2,000 test images

---

1 Data Preparation (10 points)

1. **Download and Extract**:
   - Download the dataset from Kaggle and unzip it.
2. **Explore the Dataset**:
   - Display a few example images along with their masks.
3. **Preprocess the Data**:
   - Resize images (e.g., 128x128 or 256x256).
   - Normalize pixel values (e.g., to the [0, 1] range).
   - Split the training set into training and validation subsets (e.g., 70% training, 30% validation).

2 Baseline Methods (10 points)

1. **Implement Baseline Segmentation**:
   - Use at least one traditional segmentation method, such as graph-cut segmentation, k-means, or Gaussian mixture models with k=3, as a baseline comparison.
   - Evaluate the baseline method's performance on the validation set (using the metric in Section 4).
2. **Analyze Results**:
   - Briefly describe and visualize the segmentation output from the baseline method in your report, and include observations on their performance compared to your model (see Section 6).

3 Model Implementation (30 points)

1. **Choose a Model**:
   - Implement a semantic segmentation model like U-Net, DeepLabV3, or an alternative (**with references**).
2. **Optional Data Augmentation**:
   - Apply data augmentations (e.g., rotations, flips, zoom) to improve generalization.
3. **Train the Model**:
   - Train the model on the training set using appropriate loss functions and optimizers.
   - Track training and validation loss over epochs and monitor model accuracy.

4 Model Evaluation (10 points)

1. **Evaluate Performance**:

- Use the Dice Score (DSC) to evaluate performance on the validation set. The Dice Score formula is:

$$DSC(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{1}$$

2. **Visualize Results:**
   - Display some images from the validation set, showing the original image, predicted mask, and ground truth mask for visual assessment.
3. **Generalization Test:**
   - To assess how well the model generalizes beyond the dataset, test your trained model on an external pet image, either a picture of your own pet or one found online.
   - Include the external image, along with its predicted segmentation, in the report as a qualitative example (note that this does not affect the Kaggle competition score).

5 Prediction & Kaggle Competition (10 points)

1. **Generate Test Predictions:**
   - Use your model to predict the 2,000 test images' trimap masks and save them in `test/pred_masks/` with filenames matching the original style (e.g., `1.png`, `2.png`).
2. **Prepare Submission File:**
   - Convert your predictions to a single `CSV` file using `to_csv.ipynb` (provided). The file should contain:
     - **6,000 rows** with columns "id" and "mask".
     - Each row's "id" should follow the format `x_y`, where `x` is the image ID (1-2000) and `y` is the label ID (1-3).
     - Each row's "mask" is in the format of run-length encoding (RLE), which is automatically derived through `to_csv.ipynb`.
3. **Kaggle Submission:**
   - Submit your `CSV` file to Kaggle to see your public leaderboard score, which is calculated on 30% of the test images. Final rankings will be based on the remaining 70%, revealed after the competition closes.
   - **Submission Limit:** Each team can submit up to 5 times per day, so plan accordingly.

Grading: Your grade for this section will be determined by your ranking, with the top 3 teams receiving 10 points, and the lowest rank receiving 0.

6 Report (30 points)

In a separate PDF report, summarize your findings, including:

- **Introduction:** Describe the dataset and the segmentation task.

- **Model Architecture**: Outline the model used, including references.
- **Results**:

  (1) Include metrics and example images from your experiments.

  (2) Include the performance comparison with baseline methods.

  (3) Include qualitative results from the out-of-dataset image test.
- **Challenges and Solutions**: Describe any difficulties you encountered and how you resolved them.