Computer Vision 2025
(CSE344/ CSE544/ ECE344/ ECE544)
Assignment-3

Max Marks: 40 **Due Date**: 20-Apr-2025, 11:59 PM

Instructions

- Keep collaborations at high-level discussions. Copying/plagiarism will be dealt with strictly.
- Your submission should be a single zip file **HW[n]_Roll_Number.zip**. Include only the **relevant files** arranged with proper names. A single **.pdf report** explaining your codes with relevant graphs, visualization and solution to theory questions.
- Remember to **turn in** after uploading on Google Classroom. No justifications would be taken regarding this after the deadline.
- Start the assignment early. Resolve all your doubts from TAs during their office hours **two days before the deadline.**
- Kindly **document** your code. Don't forget to include all the necessary plots and images in your report.
- All [**BONUS**] questions, if any, are optional for all the students. As the name suggests, BONUS marks will be awarded to all the students who solve these questions.
- Please ensure that your submission includes all the code used to solve the questions. You must submit a separate Jupyter Notebook (.ipynb) for each question. For example, name the file for **Question 2 as Q2.ipynb** and for **Question 3 as Q3.ipynb**. *Bonus questions can be included in the Q1 notebook file.* There are five questions so five separate notebooks are expected.
- **NOTE:** This assignment will be evaluated based on your submitted code and report. In the submitted report, include your figures, plots, or comments. For questions such as Q1.1, where you need to install libraries or you need to load weights, kindly include a screenshot of the code. It should be the same as the one given in your code.

---

1. (12 points) **CLIP** (Contrastive Language-Image Pretraining) is a multi-modal deep learning model developed by OpenAI that enables zero-shot learning for vision tasks. It learns to associate images and text by training on a vast dataset of image-text pairs collected from the internet. Following CLIP, a recent work CLIPS: An Enhanced CLIP Framework for Learning with Synthetic Captions improves the zero-shot vision tasks. For the following questions, you need to perform a detailed study and a comparative analysis.

1. (2 points) Refer to github repository and follow the readme to install required dependencies for CLIP. Alternatively, you can use hugging face's transformers library for the same.
2. (1 points) Download CLIP pretrained weights (**clip-vit-base-patch32**) and load the CLIPModal with pretrained weights.
3. (2 points) For the given sample image of human and dog, choose any 10 random textual description and generate their similarity scores.
4. (2 points) Refer to CLIPS github repo and follow README.md to install required dependencies.
5. (1 point) Load the pretrained weights for the *CLIPS-Large-14-224* model.
6. (2 points) For the previous image of human and dog, calculate the similarity scores for previous captions using CLIPS.
7. (2 point) Comment on the results obtained by both CLIP and CLIPS.

2. (5 points) **Visual question answering:** Visual Question Answering is the task of answering open-ended questions based on an image. They output natural language responses to natural language questions.
    1. (2 points) Refer to the paper BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation, and its github repository. Follow README.md to install all the dependencies and download pre-trained weights for answering visual questions.
    2. (1 point) For the previous sample image of human and dog, generate an answer to the question **"Where is the dog present in the image?"**.
    3. (1 point) For the same image, generate an answer to the question **"Where is the man present in the image?"**.
    4. (1 point) Comment on the output and accuracy of the answer for the previous two questions.

3. (10 points) **BLIP vs CLIP** BLIP can also be used for generating image captions, for the following questions, you will need to do a comparative analysis between BLIP and CLIP.
    1. (2 points) For BLIP, load the pretrained weights of image captioning.
    2. (2 points) For given sample of images, generate a caption for each image using pretrained BLIP model.
    3. (2 point) Use CLIP to evaluate the semantic accuracy of the BLIP-generated captions. Compute and interpret the similarity score between the image and the generated caption.
    4. (2 point) Use CLIPS to evaluate the scores as asked in the question above.
    5. (2 point) Discuss different metrics that can be used to quantify alignment between CLIP and BLIP outputs. Provide examples of when each metric would be most useful.

4. (8 points) **Referring Image Segmentation (RIS)** is a vision-language task where a model segments an object in an image based on a natural language description. Unlike traditional segmentation methods that rely on predefined categories, RIS understands contextual and relational descriptions (e.g., "the cat sitting on the sofa" instead of just "cat").

    1. (2 points) Refer to the LAVT paper and its github repository. Follow the README to install the required libraries and download pre-trained weights.
    2. (2 points) For each of the images in the sample folder, we also provide a reference text in this file. Show the segmented image using the given references.
    3. (2 points) Also plot the Y1 feature map obtained for each model in the given **Figure 2** of the paper.
    4. (2 points) For each of the images, provide your own reference texts and show the failure segmentation results of the given images. Show both reference text and segmentation results.

5. (5 points) **Image as reference** One shot segmentation using image reference. Unlike large language models that excel at directly tackling various language tasks, vision foundation models require a task-specific model structure followed by fine-tuning on specific tasks. For this question, we will use Matcher, a novel perception paradigm that utilizes off-the-shelf vision foundation models to address various perception tasks. Matcher can segment anything by using an in-context example without training.

    1. (2 points) Refer to the github repo for matcher. Follow the README to install the required libraries and download the necessary pre-trained weights.
    2. (3 points) We will evaluate the matcher on simple images. Using the reference images show the segmentation results for each of the images provided in this folder. Each subfolder contains two images. Take one image as a reference, one at a time, and the other as the input image. So, each subfolder will give 2 segmentation results. Show the segmentation results obtained.