Mini-Project (20 Marks) :

Image Captioning:

1. Design a CNN-LSTM system (preferably in pytroch) that can perform image captioning based on the following details: You are free to decide the CNN and LSTM architecture that best suits your case.

- Use the Flickr8K data for training and testing the model. The data is available at

  https://drive.google.com/drive/folders/1RQ5qHm0aVFqWDG9VBiSnXlNPl5T15Wf_?usp=sharing

  (Check Readme.txt for details of txt files)

The LSTM in the CNN-LSTM system accepts two types of input modalities –

- Vision embedding , which is the averaged image feature vector from CNN that goes as the first input to LSTM
- Word embedding , which is a learnable word representation given as input in every time instant ti the LSTM.

Lets call the above system as the 'baseline'.

The project objective is to enhance the vision embedding and the language embedding with the various techniques you have learnt in this course. Remember that the LSTM architecture will remain as same as the baseline , but only the input embeddings are going to be changed. Lets call your new system - 'modified baseline'.

-Produce objective numbers on the Flickr8K test data using BLEU and METEOR scores and compare it with baseline and modified baseline. (BLEU and METEOR are objective metric used for automatic evaluation of machine translation. The same metric could be used for image captioning)

-Generate subjective comparison results using baseline method and modified baseline. Justify the differences you observe in the context of your applied technique.

[20 Marks]

Contents to be shared:

- Share the training code, testing code and a technical report as a single zip file *Your_Team_ID*.zip
- Technical Report should contain
  - Team Id and all team member Ids and Names.
  - Architecture details baseline and modified baseline
  -Analysis of the results.

Note These Points:

1. To save on training time, run all images in training data to obtain CNN image embeddings for once and save it to drive.
   Use file read to store the image embeddings to local variable before the start of training. Same applies for validation and test data.