

**Summary:** In this workshop you will gain practical experience with Visual Question Answering – the topic of the assignment of assessment item 1 of this module. You are invited to work in pairs with one of you doing task 2 and the other doing task 3 for example, and both of you doing task 1.

### Task 1: Download the assignment baseline and dataset

Go to Blackboard, under CMP9137 module, and download file `visual7w.dataset-cmp9137-item1.zip` from the assessment documents (item 1), which contains the following:

- Folder `visual7w-images` contains all the images selected for the assignment, which have been resized to resolution 224x224 from the original sizes.
- Folder `visual7w-text` contains the data splits for training, development and test. Two formats are provided, one including tuples of question-answers (with four answers per question) and the other containing question-answer pairs (one answer per question).
- File `ITM_Classifier-baselines.py` is the implementation of a baseline that you can use to get you started with task 1 of the assignment.
- File `v7w.sentence_embeddings-gtr-t5-large.pkl` contains sentence embeddings (dense vectors) for all possible sentences in all the files of folder `visual7w-text`, which aims to speed up the process of generating text-based features. You can browse the content of this file by running the following from the command line: 

```
python -c "import pickle; print(pickle.load(open('v7w.sentence_embeddings-gtr-t5-large.pkl', 'rb')))"
```

Have a look at the materials above and discuss them with your peers or lecturer/demonstrator.

### Task 2: CNN/ViT models for multi-choice visual question answering

Open the folder downloaded in task 1 with Visual Code and select file `ITM_Classifier-baselines.py`. Edit the program to set the following flags with the values of your choice:

- **train\_ratio**=0.2 => where 0.2=20% of training data and 1.0=100% of training data
- **MODEL\_ARCHITECTURE** = "CNN" or "ViT" => only one of them
- **USE\_PRETRAINED\_MODEL** = True => or False, True trains faster due to less weights

Then run the program from a terminal as follows:

```
Python ITM_Classifier-baselines.py
```

The debugging information prints the model architecture, trainable parameters, loss values per epoch/batch, and classification accuracy on test data.

Record your results and discuss with your peers and lecturer/demonstrator. While the models train, you are invited to inspect the code starting with the top part (comments) and bottom part (main execution). The other code can be inspected once you have got the basic understanding.