Hackathon FAQs

This document contains all the general FAQ. There are VERY CRITICAL POINTS (and frequently encountered errors) related to other aspects (such as saving/loading data; dumping the model to CPU etc) as well here.

NOTE 1: ## VERY IMP NOTE when accessing the aiml-sandbox.talentsprint.com during model deployment

- Do ensure you have run the following two commands before you start any work *"source venv/bin/activate " and " ./runserver.sh"* (you might need to give execute permissions on the file using 'chmod +x runserver.sh'; Also note that it's important to run runserver.sh from the folder where it resides).
- The *'runserver.sh'* script will "NOT" run if *'source venv/bin/activate'* is not initiated

NOTE 2: All the datasets you want to use for the hacakthon are provided in the starter code of Hackathon. However pasting the wget commands below:

1. The main datasets for the hackathon:
   a. https://cdn.talentsprint.com/aiml/Experiment_related_data/Expression_data.zip
   b. https://cdn.talentsprint.com/aiml/FaceRecogHackathon/Datasets/ATandT/data-20190607T005435Z-001.zip
2. Additional datasets for your use:
   a. https://cdn.talentsprint.com/aiml/FaceRecogHackathon/Datasets/IMFDB_face_recog/IMFDB_final-20190607T024441Z-001.zip

NOTE 3: Ensure you're in the right folder structure when running your Colab cells. Several parts of the code sometimes depend on the right base directory.

NOTE 4: Add %%capture as a top line on the cell to suppress log messages on Colab output. This would be necessary sometimes because log messages might hang the Colab session. And this could act as a convenient switch.

NOTE 5: One of the keys of cross deployment is that you can train your models on GPU (i.e. in Colab) but deploy on a CPU device (such as the server aiml-sandbox.talentsprint.com which is listening to the mobile app). Therefore when you load your *state_dictionary* into the model using *torch.load*, you have to ensure that you map the model to the CPU device.

Here is the link to understand this issue: https://discuss.pytorch.org/t/on-a-cpu-device-how-to-load-checkpoint-saved-on-gpu-device/349

NOTE 6: When you pass a single input image to a model to get the feature representation, it is important to ensure that your input image dimension matches the dimension of the first layer of the model. Say face1 is your input image tensor, you can consider using *torch.unsqueeze(face1,0)* or *face1.unsqueeze_(0)* i.e. adding an empty dimension.