Deep Learning Report

Programming Assignment - 3

Ayush Abrol
B20AI052

Question 01

Aim:

- Perform Attribute Prediction Task on CelebA dataset for the attributes available in the dataset. For this task, use a VGG16/ResNet18 backbone and train your model in a Multi-Task Learning fashion. Report the following:
    - Train the model for the prediction of 8 attributes out of 40 attributes and report the following:
        - Mention your choice of attributes, backbone, and other parameters and the reason behind your choice.
        - Report task-wise accuracy.
        - Report the overall accuracy.

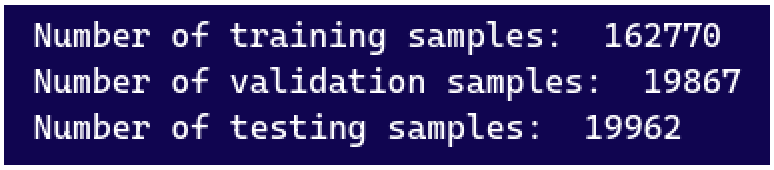
Dataset used: CelebA

Procedure:

- All the necessary imports such as numpy, pandas, torch, torchvision, tqdm, warnings, etc. were done.
- Set up the device as GPU (cuda).
- Added train transformations such as Random Horizontal Flip, RandomCrop, Resize converted the images to Tensor and normalized the images with mean = [0.5, 0.5, 0.5] and std=[0.5, 0.5, 0.5]
- Added validation and test transformations as CenterCrop, Resize, converted the images to Tensor and normalized the images with mean = [0.5, 0.5, 0.5] and std=[0.5, 0.5, 0.5]
- Downloaded the dataset from the link provided in the question and kept it inside the data directory inside the current working directory.
- Then, I split the dataset into train, validation and test sets.
- Set up the train indices, val indices and test indices.
- Selected the following 8 attributes out of all the 40 attributes:

  labels = ['Bald', 'Bangs', 'Black_Hair', 'Blond_Hair', 'Brown_Hair', 'Male', 'Smiling','Young']

- I have selected these attributes as they are diverse and cover a range of attributes such as hair color, gender, and age and can contribute the most for the facial features.
- Then, I defined a custom PyTorch dataset class called SubsetCustom. This class takes in three parameters:
    - **dataset:** This is the original dataset from which we want to create a subset.
    - **indices:** This is a list of indices that represents the subset of the original dataset that we want to create.
    - **selected_indices**: This is a list of indices that represents the attributes we want to select for each image in the subset.
    - **The __getitem__ method** of this class takes an index idx and returns the image and attribute values for that index from the original dataset. However, instead of returning all the attributes for that index, it only returns the selected attributes specified by selected_indices. This allows us to create a subset of the original dataset with only the selected attributes.
    - **The __len__ method** returns the length of the subset, which is the length of the indices list.
- Then I created three new datasets: train_dataset_new, val_dataset_new, and test_dataset_new by calling the SubsetCustom class that we defined earlier.
- Each of these new datasets is created from the original train_dataset, val_dataset, and test_dataset respectively, but with only the selected attributes for each image.
- The selected_indices list represents the indices of the attributes that we want to select for each image. This list was chosen earlier based on the 8 attributes that we want to predict for this task.
- Then, I finally defined the Dataloaders with batch_size = 128 for the selected classes.

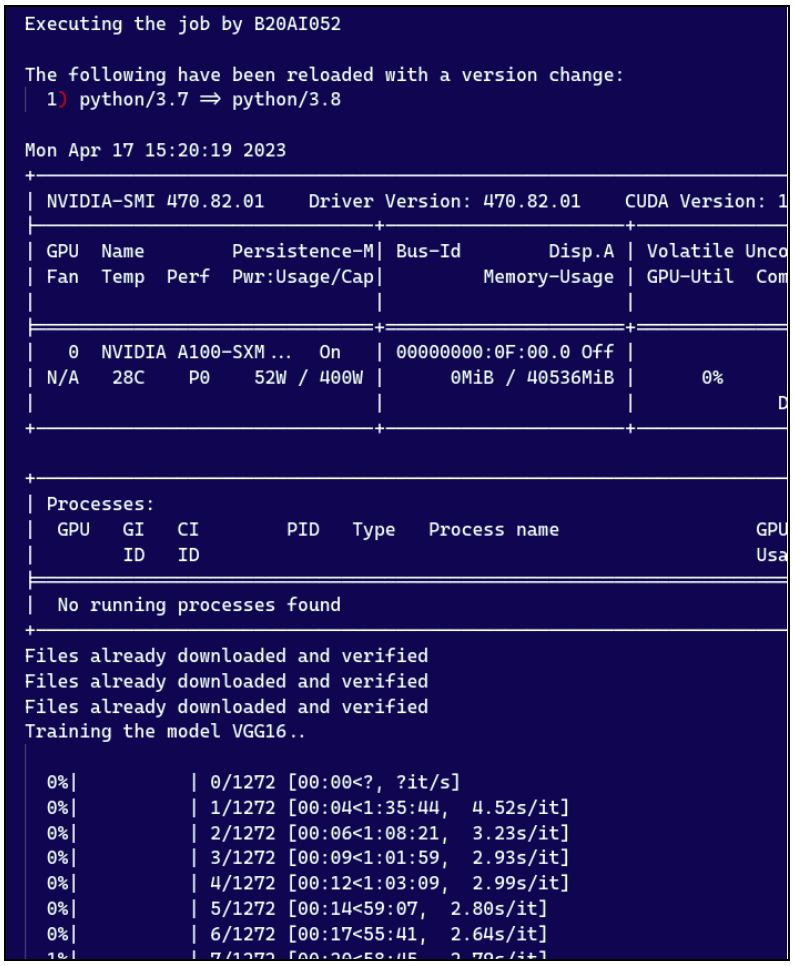


```
Number of training samples:  162770
Number of validation samples:  19867
Number of testing samples:  19962
```

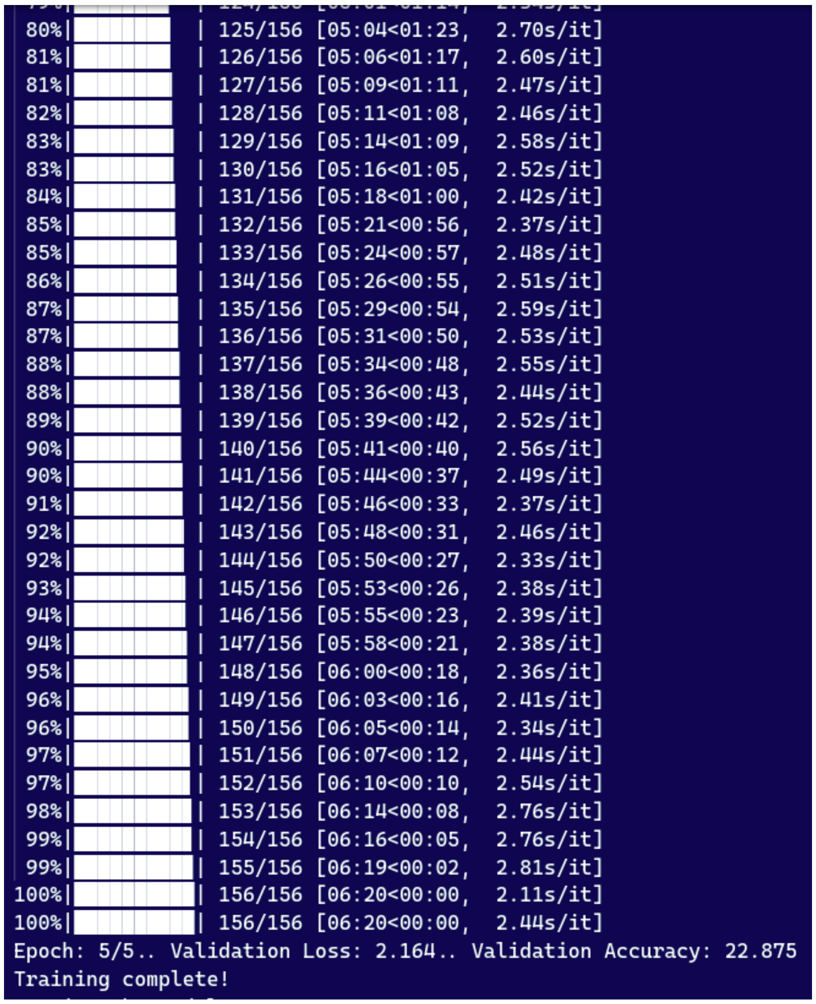
- Visualization of some sample images from the training dataset

- Now, In the case of the CelebA dataset, which consists of 202,599 RGB images with a resolution of 178x218, both VGG16 and ResNet18 can be used as the backbone model. However, VGG16 may be a better choice than ResNet18 for the following reasons:
    - VGG16 has a simpler architecture with fewer layers than ResNet18, making it easier to train and computationally efficient. This can be especially important when working with large datasets like CelebA.
    - VGG16 has shown better performance than ResNet18 on certain image classification tasks, especially when the number of parameters is not a constraint.
- Used torchvision.model pretrained VGG16 model and fine-tuned it for the final fc-layer to predict the 8 selected attributes for our task. This is achieved by modifying the last fully connected layer of the VGG16 model (classifier[6]) and replacing it with a new fully connected layer that has 8 output neurons.
- The nn.Linear(4096, 8) creates a new fully connected layer with 4096 input features (which is the output size of the previous layer) and 8 output features (one for each of the selected attributes).
- Then, I defined my optimizer as Adam optimizer with learnin_rate = 3e-4 and loss function as Cross Entropy loss.

- Then I defined a training function that takes as input a model, data loaders, optimizer, and criterion, and trains the model for a specified number of epochs (5 in my case). In each epoch, the function loops through the training data, computes the loss and updates the model's parameters using backpropagation. It also evaluates the model's performance on the validation data and stores the losses and accuracies for plotting later.
- Trained the model on hpc server using my login credentials for the same and got slurm-42536.out file as the output. (Link attached herewith)



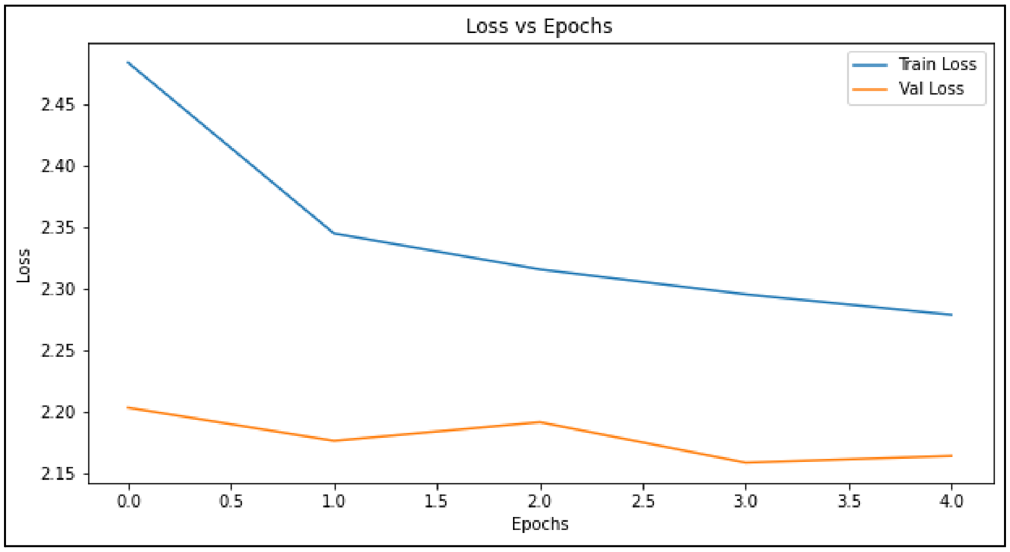
```
Executing the job by B20AI052

The following have been reloaded with a version change:
  1) python/3.7 => python/3.8

Mon Apr 17 15:20:19 2023
+-----------------------------------------------------------------------------
| NVIDIA-SMI 470.82.01    Driver Version: 470.82.01    CUDA Version: 1
|-------------------------------+----------------------+----------------------
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Unco
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Com
|                               |                      |
|===============================+======================+======================
|   0  NVIDIA A100-SXM...  On   | 00000000:0F:00.0 Off |
| N/A   28C    P0    52W / 400W |      0MiB / 40536MiB |      0%
|                               |                      |                   D
+-------------------------------+----------------------+----------------------

+-----------------------------------------------------------------------------
| Processes:
|  GPU   GI   CI        PID   Type   Process name                          GPU
|        ID   ID                                                          Usa
|=============================================================================
|  No running processes found
+-----------------------------------------------------------------------------
Files already downloaded and verified
Files already downloaded and verified
Files already downloaded and verified
Training the model VGG16..

  0%|          | 0/1272 [00:00<?, ?it/s]
  0%|          | 1/1272 [00:04<1:35:44,  4.52s/it]
  0%|          | 2/1272 [00:06<1:08:21,  3.23s/it]
  0%|          | 3/1272 [00:09<1:01:59,  2.93s/it]
  0%|          | 4/1272 [00:12<1:03:09,  2.99s/it]
  0%|          | 5/1272 [00:14<59:07,  2.80s/it]
  0%|          | 6/1272 [00:17<55:41,  2.64s/it]
  1%|          | 7/1272 [00:20<58:45,  2.79s/it]
```



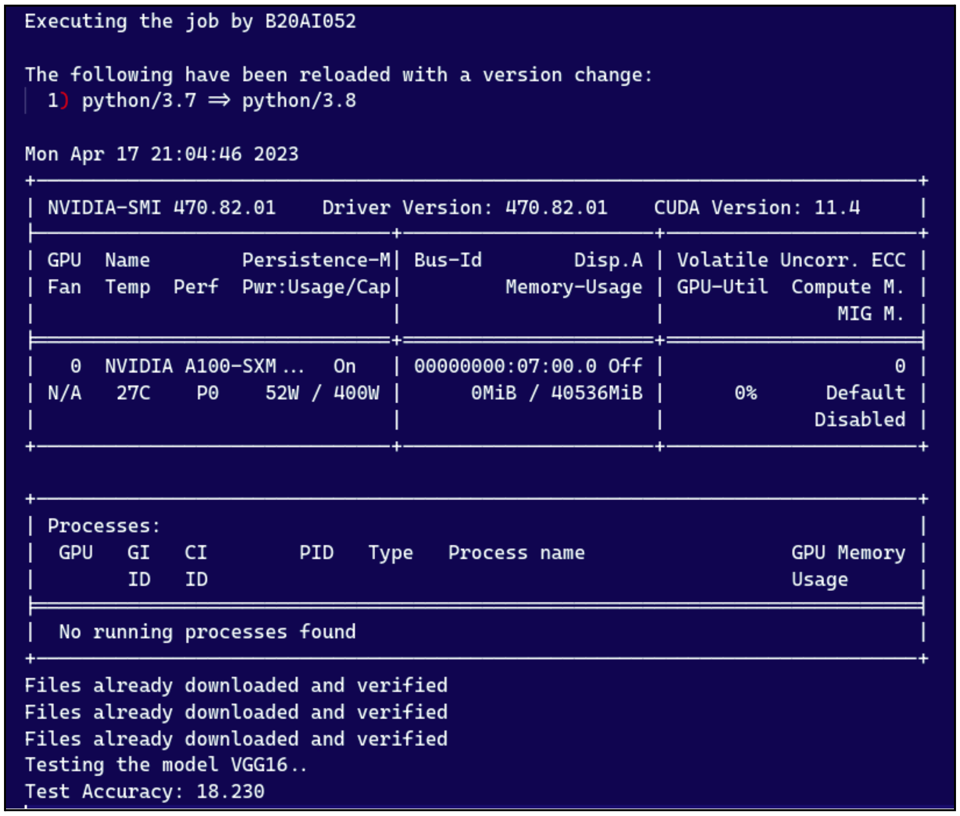
```
 80%|████████  | 125/156 [05:04<01:23,  2.70s/it]
 81%|████████  | 126/156 [05:06<01:17,  2.60s/it]
 81%|████████▏ | 127/156 [05:09<01:11,  2.47s/it]
 82%|████████▏ | 128/156 [05:11<01:08,  2.46s/it]
 83%|████████▎ | 129/156 [05:14<01:09,  2.58s/it]
 83%|████████▎ | 130/156 [05:16<01:05,  2.52s/it]
 84%|████████▍ | 131/156 [05:18<01:00,  2.42s/it]
 85%|████████▍ | 132/156 [05:21<00:56,  2.37s/it]
 85%|████████▌ | 133/156 [05:24<00:57,  2.48s/it]
 86%|████████▌ | 134/156 [05:26<00:55,  2.51s/it]
 87%|████████▋ | 135/156 [05:29<00:54,  2.59s/it]
 87%|████████▋ | 136/156 [05:31<00:50,  2.53s/it]
 88%|████████▊ | 137/156 [05:34<00:48,  2.55s/it]
 88%|████████▊ | 138/156 [05:36<00:43,  2.44s/it]
 89%|████████▉ | 139/156 [05:39<00:42,  2.52s/it]
 90%|████████▉ | 140/156 [05:41<00:40,  2.56s/it]
 90%|█████████ | 141/156 [05:44<00:37,  2.49s/it]
 91%|█████████ | 142/156 [05:46<00:33,  2.37s/it]
 92%|█████████▏| 143/156 [05:48<00:31,  2.46s/it]
 92%|█████████▏| 144/156 [05:50<00:27,  2.33s/it]
 93%|█████████▎| 145/156 [05:53<00:26,  2.38s/it]
 94%|█████████▎| 146/156 [05:55<00:23,  2.39s/it]
 94%|█████████▍| 147/156 [05:58<00:21,  2.38s/it]
 95%|█████████▍| 148/156 [06:00<00:18,  2.36s/it]
 96%|█████████▌| 149/156 [06:03<00:16,  2.41s/it]
 96%|█████████▌| 150/156 [06:05<00:14,  2.34s/it]
 97%|█████████▋| 151/156 [06:07<00:12,  2.44s/it]
 97%|█████████▋| 152/156 [06:10<00:10,  2.54s/it]
 98%|█████████▊| 153/156 [06:14<00:08,  2.76s/it]
 99%|█████████▊| 154/156 [06:16<00:05,  2.76s/it]
 99%|█████████▉| 155/156 [06:19<00:02,  2.81s/it]
100%|██████████| 156/156 [06:20<00:00,  2.11s/it]
100%|██████████| 156/156 [06:20<00:00,  2.44s/it]
Epoch: 5/5.. Validation Loss: 2.164.. Validation Accuracy: 22.875
Training complete!
```

- Obtained the final results after 5 epochs as:
    - Training loss = 2.279
    - Training accuracy = 23.500%
    - Validation loss = 2.164
    - Validation accuracy = 22.875%
- Then, I saved the models in the models directory inside the same working directory. Also save the train and val loss and accuracy arrays created.

- Plots obtained after training:

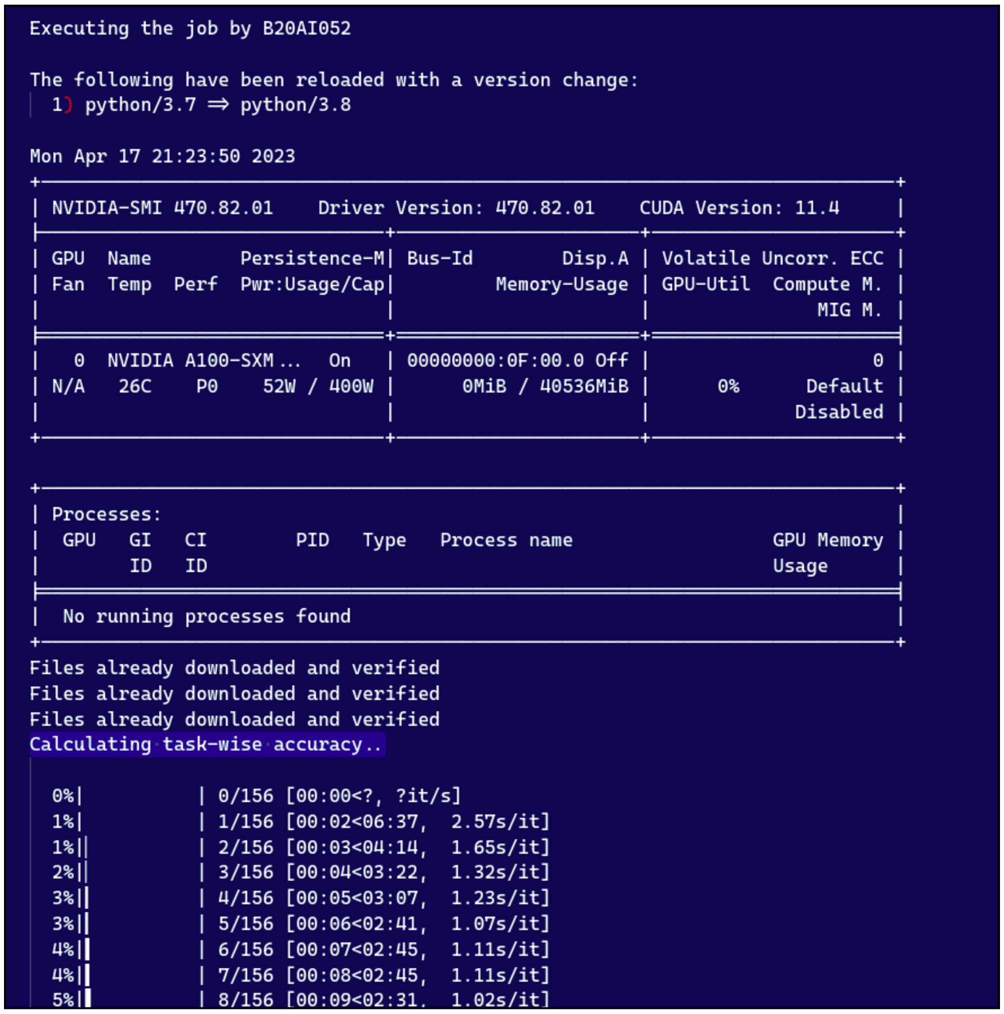


- Then, tested the model on test set and this also I did on the hpc server due to GPU constraints and obtained a slurm-46202.out file ([Link attached herewith]())

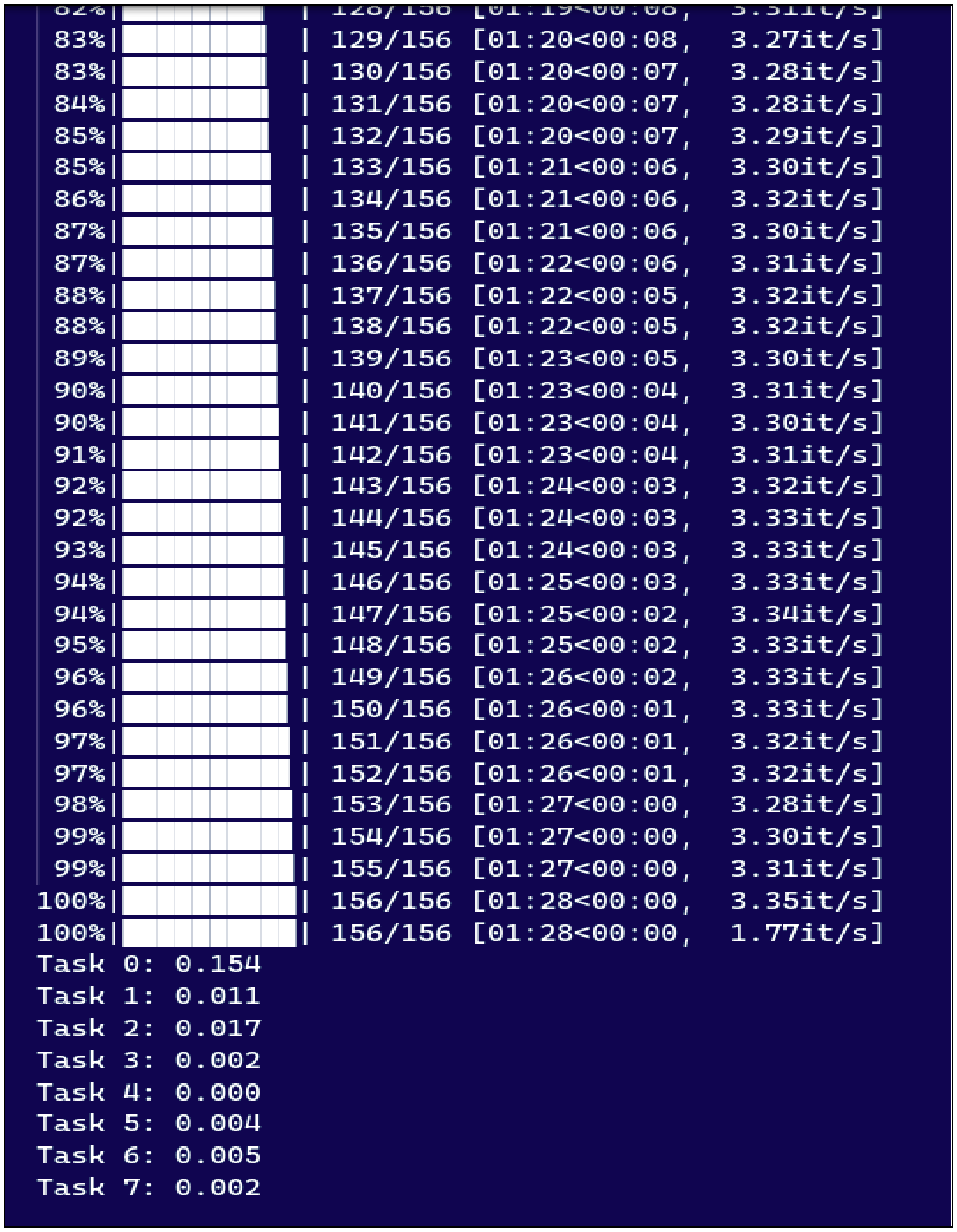


```
Executing the job by B20AI052

The following have been reloaded with a version change:
  1) python/3.7 => python/3.8

Mon Apr 17 21:04:46 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.82.01    Driver Version: 470.82.01    CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A100-SXM...  On   | 00000000:07:00.0 Off |                    0 |
| N/A   27C    P0    52W / 400W |      0MiB / 40536MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
Files already downloaded and verified
Files already downloaded and verified
Files already downloaded and verified
Testing the model VGG16..
Test Accuracy: 18.230
```

- Then, I tested again for the task-wise accuracy and obtained a slurm-42604.out file (Link attached herewith)



```
Executing the job by B20AI052

The following have been reloaded with a version change:
  1) python/3.7 => python/3.8

Mon Apr 17 21:23:50 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.82.01    Driver Version: 470.82.01    CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A100-SXM...  On   | 00000000:0F:00.0 Off |                    0 |
| N/A   26C    P0    52W / 400W |      0MiB / 40536MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+


+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
Files already downloaded and verified
Files already downloaded and verified
Files already downloaded and verified
Calculating task-wise accuracy..

  0%|          | 0/156 [00:00<?, ?it/s]
  1%|          | 1/156 [00:02<06:37,  2.57s/it]
  1%|▏         | 2/156 [00:03<04:14,  1.65s/it]
  2%|▏         | 3/156 [00:04<03:22,  1.32s/it]
  3%|▎         | 4/156 [00:05<03:07,  1.23s/it]
  3%|▎         | 5/156 [00:06<02:41,  1.07s/it]
  4%|▍         | 6/156 [00:07<02:45,  1.11s/it]
  4%|▍         | 7/156 [00:08<02:45,  1.11s/it]
  5%|▌         | 8/156 [00:09<02:31,  1.02s/it]
```



```
 82%|████████▏ | 128/156 [01:19<00:08,  3.31it/s]
 83%|████████▎ | 129/156 [01:20<00:08,  3.27it/s]
 83%|████████▎ | 130/156 [01:20<00:07,  3.28it/s]
 84%|████████▍ | 131/156 [01:20<00:07,  3.28it/s]
 85%|████████▍ | 132/156 [01:20<00:07,  3.29it/s]
 85%|████████▌ | 133/156 [01:21<00:06,  3.30it/s]
 86%|████████▌ | 134/156 [01:21<00:06,  3.32it/s]
 87%|████████▋ | 135/156 [01:21<00:06,  3.30it/s]
 87%|████████▋ | 136/156 [01:22<00:06,  3.31it/s]
 88%|████████▊ | 137/156 [01:22<00:05,  3.32it/s]
 88%|████████▊ | 138/156 [01:22<00:05,  3.32it/s]
 89%|████████▉ | 139/156 [01:23<00:05,  3.30it/s]
 90%|████████▉ | 140/156 [01:23<00:04,  3.31it/s]
 90%|█████████ | 141/156 [01:23<00:04,  3.30it/s]
 91%|█████████ | 142/156 [01:23<00:04,  3.31it/s]
 92%|█████████▏| 143/156 [01:24<00:03,  3.32it/s]
 92%|█████████▏| 144/156 [01:24<00:03,  3.33it/s]
 93%|█████████▎| 145/156 [01:24<00:03,  3.33it/s]
 94%|█████████▎| 146/156 [01:25<00:03,  3.33it/s]
 94%|█████████▍| 147/156 [01:25<00:02,  3.34it/s]
 95%|█████████▍| 148/156 [01:25<00:02,  3.33it/s]
 96%|█████████▌| 149/156 [01:26<00:02,  3.33it/s]
 96%|█████████▌| 150/156 [01:26<00:01,  3.33it/s]
 97%|█████████▋| 151/156 [01:26<00:01,  3.32it/s]
 97%|█████████▋| 152/156 [01:26<00:01,  3.32it/s]
 98%|█████████▊| 153/156 [01:27<00:00,  3.28it/s]
 99%|█████████▊| 154/156 [01:27<00:00,  3.30it/s]
 99%|█████████▉| 155/156 [01:27<00:00,  3.31it/s]
100%|██████████| 156/156 [01:28<00:00,  3.35it/s]
100%|██████████| 156/156 [01:28<00:00,  1.77it/s]
Task 0: 0.154
Task 1: 0.011
Task 2: 0.017
Task 3: 0.002
Task 4: 0.000
Task 5: 0.004
Task 6: 0.005
Task 7: 0.002
```

END OF ASSIGNMENT (QUESTION 01)