I want to do the following over a vision dataset-

1. I have a data loader python file. I want to update it to load the image files' train, test, and validation data. It should also select 6 random images from the train, test, and validation sets and arrange them on a two-row, three-column layout with the tight box and in 16in x 10 in 300 dpi format. These images should be saved in “output/image\_output” with the date and time of the run. Each image should have its name as a title in bold, Times New Roman font.
2. I have a base trainer Python file. I want to update it to process the given dataset loaded through the data loader file and process it over the Cuda device. The processed data is then loaded for training for default 10 epochs with 2 warmup epochs. The model is then tested and validated over the test and validation dataset. All of these (train, test, validation) will have TQDM. Once completed, the training, testing, and validation results are saved in a CSV file with the time required for each epoch. This result is also plotted and saved in a pdf file of size 16\*10 in 300 dpi format. The base trainer class should also have training testing and validation confusion metrics calculated. I also wanted to save them in pdf format with 32in\*32 in pdf bbbox tight for each algorithm. The title needs to be the name of the algorithm (Resnet34, for example), the type of the analysis (train, test, or validation), and the number of epochs. With only X-axis labels at 45 degrees and the color map “Blues.”
3. I have resnet\_trainer and vgg\_trainer separate Python files; these take independent models saved in separate files. Both use the base trainer file and add a specific layer list where 20 images are randomly selected for feature extraction and saved in JPG files based on the listed layers.
4. Is there a way we can modify the base trainer class to have training testing and validation confusion metrics calculated? I also wanted to save them in pdf format with 32in\*32 in pdf bbbox tight for each algorithm. The title needs to be the name of the algorithm (Resnet34, for example), the type of the analysis (train, test, or validation), and the number of epochs. With X-axis labels at 45 degrees and color map “Blues”.
5. For each ResNet and VGG Model, I would like to create a model file that will take the ResnetTrainer and VGGTrainer Class (respectively) and perform the analysis. When running, all the models should train, test, and validate the model over the set data and save the analysis result in addition to the epoch time, save the feature maps of the random 20 images based on the freeze layers described, and save the confusion matrix for train, test, and validation for all the models in a prescribed folder. I will give you the code for each model one prompt at a time, please update it accordingly and return the complete code.

main.py calls resnettrainer and vgg trainers, which call for base trainer and the sub models per group correct. as the model is being trained we are also freezing layers to extract features from random 20 images for each sub model. we then test and validate the model. save the results of the training testing validation in csv file and plot the graphs. we also save the extracted features in jpg file per image

I now want to create files for vgg13bn, 16bn, and 19bn models with the layer list specific to each model I will use this base trainer class and train, test, validate, extract features, save features, and create a confusion matrix as well.

Dataset Information:

Can you share details about your dataset? What type of data are you working with, and what are the classes you want to classify?

How many data samples do you have, and is your dataset labeled?

You mentioned that your dataset contains images captured from different camera models. Could you provide more details about the dataset? Specifically:

How many classes (camera models) are there in your dataset?

How many images do you have for each camera model class?

Do you have a rough idea of the distribution of images among the classes?

Understanding the class distribution and the number of images per class is important for model training and evaluation.

Existing Code:

Please share your current code or script for building the CNN model. This will help me understand your current approach.

Research Objectives:

What are the main research objectives or questions you aim to address with this project?

Data Preprocessing:

Have you performed any data preprocessing steps, such as data augmentation, normalization, or resizing?

Model Architecture:

What is the current architecture of your CNN model? If you haven't started, do you have any specific architecture in mind?

1. VGG (Visual Geometry Group) Models:
   1. VGG16
   2. VGG19
2. ResNet Models (Residual Networks):
   1. ResNet18
   2. ResNet34
   3. ResNet50
   4. ResNet101
   5. ResNet152
3. Inception Models (GoogleNet):
   1. InceptionV1
   2. InceptionV2
   3. InceptionV3
   4. InceptionResNetV2
4. ResNeXt Models:
   1. ResNeXt50
   2. ResNeXt101
5. MnasNet (Mobile NASNet)
   1. MnasNet A1
6. MobileNet Models:
   1. MobileNetV1
   2. MobileNetV2
   3. MobileNetV3

How do you split your dataset into training, validation, and test sets?

What optimization and loss functions are you currently using?

Do you have a training plan, or are you looking for guidance on setting up a training pipeline?

Evaluation Metrics:

What evaluation metrics are you using to measure the performance of your model? (e.g., accuracy, F1 score, etc.)

Challenges:

Are there any specific challenges or issues you've encountered during your work so far?

Desired Outcomes:

What are your expectations for the research project? What results or insights are you hoping to obtain?