Machine Learning - Final Project

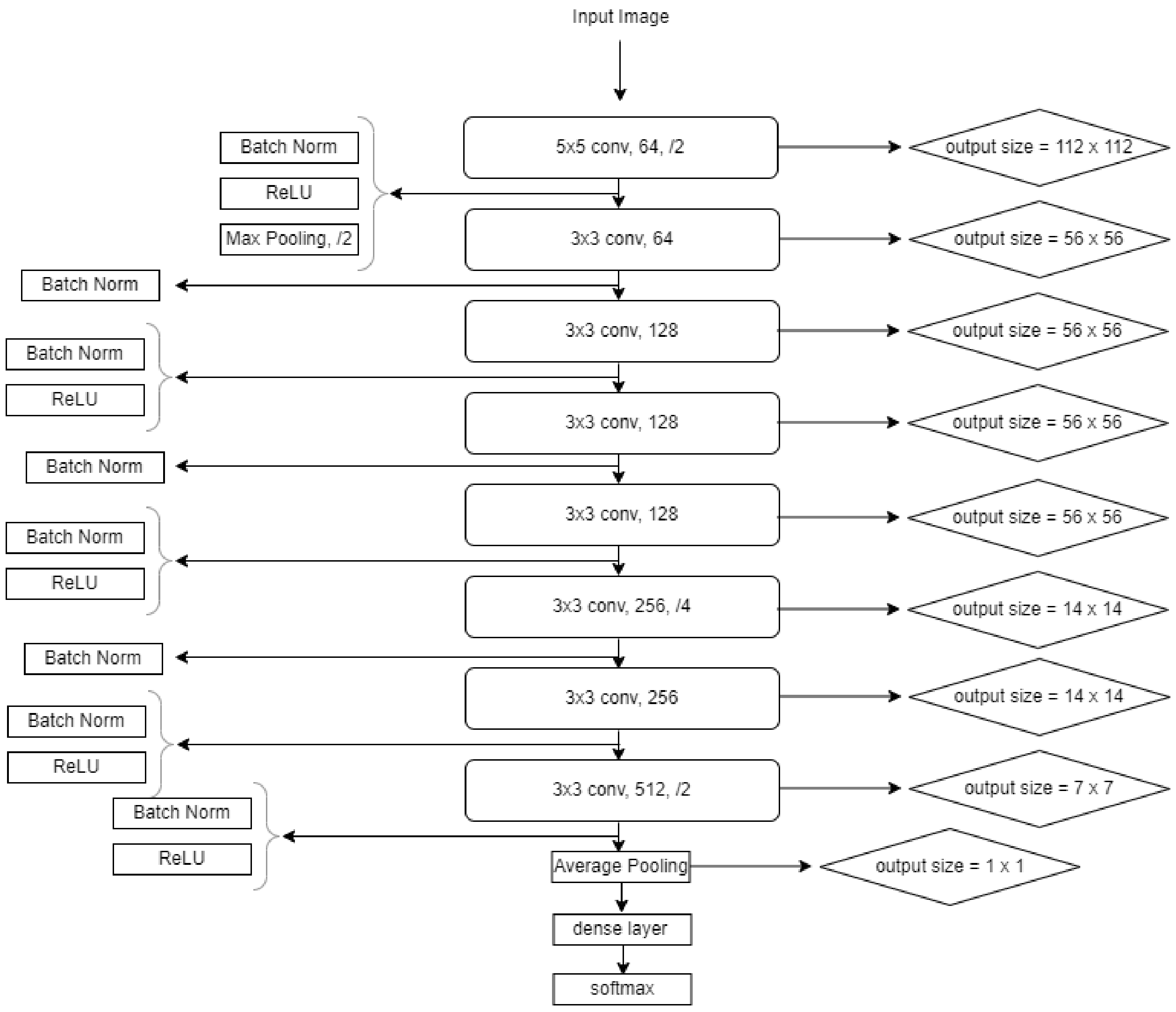
This project is all about images and how good we can classify them. Here we are trying to work with a dataset containing images of seas, mountains, and jungles. Half of these images are real images gathered form different sources and the other half generated using deep learning, text-to-image models such as Stable Diffusion and DALL-E.

We would like you to perform classification for two specific cases:

1. Determine whether the image depicts a sea, mountain, or jungle.
2. Identify whether the image is real or fake.

To accomplish the aforementioned classifications, you will need to consider two procedures:

1. **Training from Scratch:** Implement the CNN architecture provided below and train it using the provided dataset. As you may have guessed, you will need to train two separate models, each with a distinct objective (one model for each classification). This approach will help you successfully complete this part of the project.



2. **Transfer Learning:** Now, it's time to use a pretrained model. As ChatGPT says:

   > "Pretrained models are models that have been trained on large-scale datasets and are made available by researchers or organizations. They can save significant time and computational resources since they are already trained on large datasets. By leveraging these pretrained models, researchers and developers can build upon existing knowledge and fine-tune the models for specific tasks with smaller, task-specific datasets. This transfer learning approach enables faster development and often leads to better performance compared to training models from scratch."

   For our task the selected pretrained models are:

   - ResNet-152
   - MobileNetV3

   You have to fine-tune these models on the provided dataset.

To optimize the classification procedures mentioned above, it is crucial to consider the following techniques:

1. **Data Augmentation:** According to ChatGPT:

   > "Data augmentation allows us to artificially expand the size of our training dataset by generating new, modified versions of the existing data. This is particularly beneficial when the original dataset is limited in size, as it helps prevent overfitting and improves generalization by providing additional diverse samples for the model to learn from."

   Since our dataset has a limited number of instances, employing data augmentation techniques is highly recommended to augment the available data and enhance the model's performance. You can use as many techniques as you want in order to augment the data to produce better results. You can find numerous examples here. However, there are specific data augmentation methods that are essential to use and analyze the results:

   - ColorJitter
   - RandomRotation
   - RandomCrop
   - GaussianBlur
   - Grayscale
   - RandomHorizontalFlip and RandomVerticalFlip

   You don't need to implement these techniques from scratch; most deep learning frameworks provide predefined transformations that you can utilize. Make sure to include in your report an evaluation of the impact of these transformations and an explanation of what each transformation accomplishes.

   Another data augmentation technique is *mixup* which **you need to implement by your own**. You can find about it in the provided paper. Don't forget to explain how it works in your report and analyze your results after applying it.

2. **Optimization Algorithm:** The following optimizers are those you must consider in your work:

- Stochastic Gradient Descent (SGD)
- Adam
- AdamW
- RMSprop

You can take advantage of the implementations of these optimizers in your selected framework. You should provide details about each one of them and how they work in your report and again don't forget to analyze the impact of each one of them in your classification task.

In order to increase the generalization of deep learning models, Sharpness-Aware Minimization (SAM) has been introduced. According to the paper:

> "For modern overparameterized models such as deep neural networks, typical optimization approaches can easily result in suboptimal performance at test time. In particular, for modern models, the training set loss is typically non-convex, with multiple local and even global minima that may yield similar values of the loss function while having significantly different generalization performance. Motivated by the connection between sharpness of the loss landscape and generalization, SAM functions by seeking parameters that lie in neighborhoods having uniformly low loss value (rather than parameters that only themselves have low loss value)."

SAM method utilizes a base optimizer, such as SGD, to execute its technique. More information about SAM can be found in the referenced paper.

It is not necessary for you to implement SAM by yourself. Just use it in your model and provide the results. You can use the implementation provided in this link. You also need to provide an intuitive explanation for the SAM method in your report.

3. **Learning Rate Schedular:** One straightforward option for setting the learning rate in optimization algorithms like SGD is to use a constant value. But there is another option to use. A learning rate schedule is a technique used to adapt the learning rate during the gradient descent optimization process. By adjusting the learning rate over time, it aims to improve performance and reduce training time in deep learning models. During the training of your model, there are several options that you should consider:
    - LinearLR
    - ExponentialLR
    - CyclicLR
    - StepLR

   In your report, it is essential to explain each learning rate schedule option and thoroughly analyze the impact of applying them to the classification task on the provided dataset.

Important notes:

1. You will have an **in-person meeting session** to present your work (both your code and the results). It is recommended to prepare some slides to present. You also need to be able to explain the details you provided in your report.

2. In addition, you need to provide a **comprehensive report** for your project. It is crucial to provide detailed analysis of your results. This analysis should explain the significance of the findings, highlight any patterns or trends observed, and offer interpretations that contribute to a deeper understanding of the data or problem at hand. One of the things you need to include in your report is the **train and validation learning curves** for all of your training processes. Using these learning curves, you can assess whether your model is suffering from overfitting or underfitting.
3. The provided dataset consists of 3000 instances. A portion of this data has been reserved for evaluating the performance of your model. During the presentation day, this reserved portion will be provided to you for testing your model's performance. So, **don’t forget** to save the parameters of your models.
4. To enhance the performance of the model, an option is to augment the dataset by collecting and generating new images from various sources. However, it is important to note that this step is **optional**. If additional data is incorporated, **you must** save the parameters of the trained model **separately**. This allows for testing the model without the added data, ensuring a fair evaluation and comparison of the models' performance.
5. The choice of framework is up to you and you can use PyTorch, TensorFlow, or Keras.
6. The honor code, as mentioned in the first homework of the course, remains valid, and any form of cheating or academic dishonesty **will not be tolerated**.

Good luck,

Sina Abbasi, Armin Tourajmehr