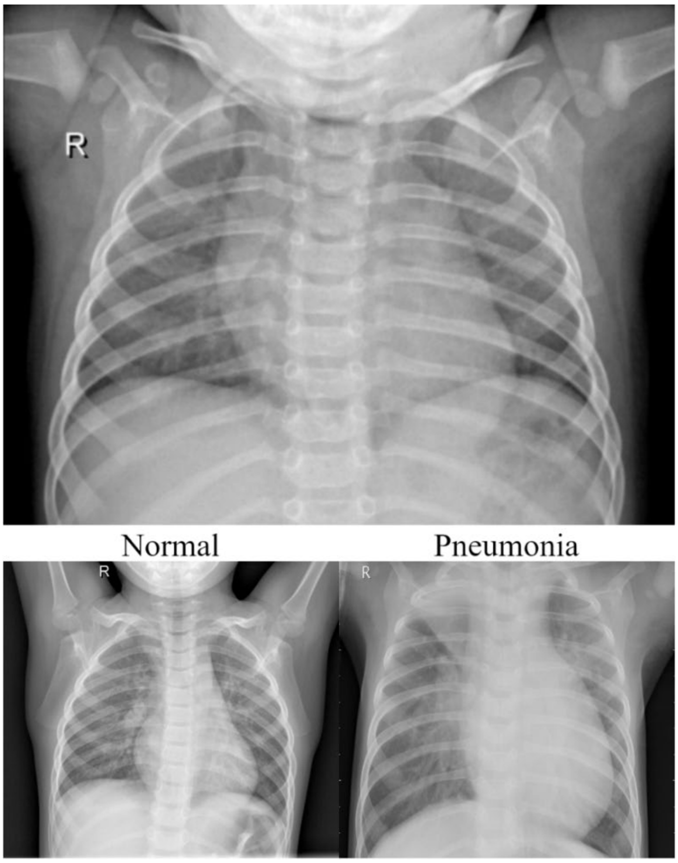
Deep Learning Summer Course - Project

Summer 2024
Wednesday 14-8-2024



In light of the recent COVID-19 pandemic, we will take a look at one of the severe symptoms of the disease, pneumonia. The dataset we have consists of x-ray images of patients who either has pneumonia or are healthy, and you must develop a model to distinguish them.

Question or comments: Lucas Alexander Damberg Torp Dyssel

When to submit

You need to submit before Friday 16-08-2024 at 9:00.

What to submit

You must hand-in your code, your model, you architecture, all visualizations saved as images. Instead of creating a report, we expect your code to be heavily commented, as well as specifications as to who worked on each section. Clearly specify at the beginning of your code all group members who participated in the project. Please save model files in .pth format, and only submit your best model.

- The name of your model should be **group_x.pth** where x is your group number.
- The name of your architecture should be **group_x.py**, and should ONLY contain your architecture class which should also be named **class group_x(nn.Module)**, including relevant imports.
    - It should also include a global variable with the dimension of your images called **input_dim = (h, w)**, which should be a tuple where h is the height and w is the width of the images.
    - It should also include a global variable called **channel_dim = c** with the number of channels your model trained on. For grayscale this would be 1 and for RGB this would be 3.

Below you can see an example of how your group_x.py file should look.

```
# Your imports
input_dim=(h,w)
channel_dim=c

class group_x(nn.Module):
    def __init__(self):
        super(group_x, self).__init__()
        # Your layers
    def forward(self, x):
        # Your layers
        return x
```

The test ranking script will use the following transformations. Ensure you apply these transformations to your test and validation datasets to maintain consistency. This will help avoid discrepancies between training and test images, ensuring your model's performance remains optimal.

```
from torchvision import transforms
transform = transforms.Compose([transforms.ToTensor(),
transforms.Resize(input_dim)])
```

If you fail to follow the submission guidelines you will be told to redo your submission if time permits. If not, your model will NOT be part of the competition.

How to submit

You need to submit everything in the following Google Drive in the group folder corresponding to the group you were assigned on the first day of the course.

Notes

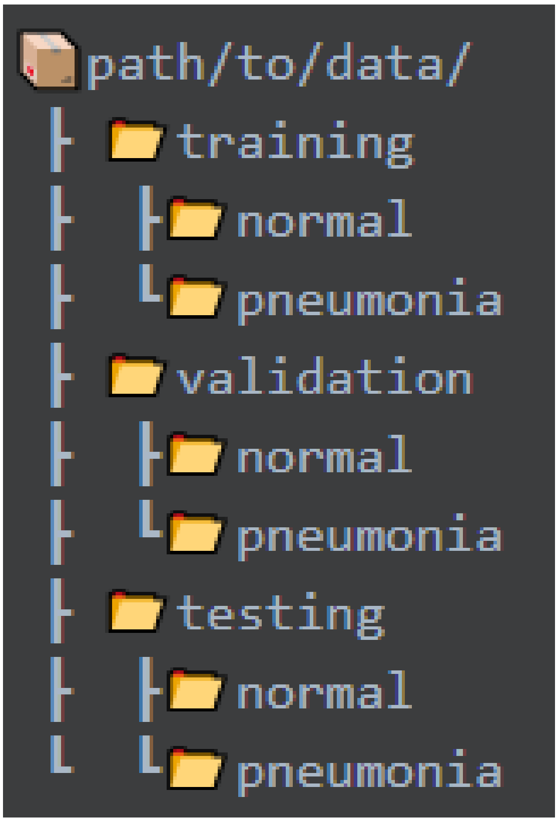
You must do this assignment in your pre-assigned groups, and only one submission is needed per group. The data is available on ItsLearning under Plans/Project/data.zip.

Training the networks can be a slow process, keep this in mind. You are NOT allowed to use pretrained networks of any kind, but you are allowed to use any networks discussed during the exercises.

Please read through the entire project description, as there will be a Q&A session on Thursday 15-08-2024 at 10:00. For questions or comments, send me an email. Bear in mind, this should not be used for assistance in the exercises, as you are expected to complete this project on your own.

Task 1 – The Dataset

All images are x-ray images taken in the chest region of children aged 1-5 from the Guangzhou Women and Children's Medical Center. There are 1100 images of healthy humans, as well as 1100 images of people with pneumonia.
Keep in mind, that the image format is jpeg, and there are 3 color channels.
You need to organize the data into directories as shown on Figure 1. You need to determine the training/validation/testing split yourselves but need to justify your split choice.



```
path/to/data/
 ├ training
 ├   ├ normal
 ├   └ pneumonia
 ├ validation
 ├   ├ normal
 ├   └ pneumonia
 ├ testing
 ├   ├ normal
 └   └ pneumonia
```

Task 2 – Create a custom dataset class and a dataloader

In this assignment, you will be working with a substantial dataset of images. You will need to implement a data loader using PyTorch's data loaders and create a custom dataset class to efficiently manage and preprocess the data.

Furthermore, you should enhance the model's performance by applying data augmentation. Remember that all images should be of same size during training, validation and testing. Which data augmentation is up to you, but you need to justify your choice in a comment.

Task 3 – Construction the network

After successfully creating both the custom dataset and the dataloader, you need to create a neural network, and use the data loader to feed the network. The architecture, complexity and regularization are all up to you, but you need to justify your choices in comments. You are more than welcome to replicate already known architectures or architectures we made during the course, but you are NOT allowed to use any pretrained networks. You are also not allowed to use any training data that is not included on ItsLearning.

Carefully consider which hyperparameters to test and strategically try to find the optimal architecture for the task. In the comments, please describe your method for the optimization and your choice of hyperparameters. Remember that there is an underlying competition, and the highest accuracy wins. The competition will be measured based on the saved model, so make sure to submit only the best one!

Task 4 - Visualizing your results

Finally, you must visualize some aspects of your model. It can be a graph of the training/validation performance, visualization of the filters or feature maps, or anything you can think of. This must be saved as an image file and uploaded along with your model and code.