Proposal for COMP6721-Applied Artificial Intelligence Project - Group Q

Rohan Chopra 40233019 | Harman Singh Jolly 40204947 | Harman Preet Kaur 40198317 | Abhishek Handa 40231719

1. Problem Statement and Application

The destruction caused by COVID-19 has created a need for diagnosis that is reliable, effective, and fast. Chest X-ray acquisition is easy but needs to be evaluated by an expert radiologist. We propose that a deep CNN model can help diagnose new diseases much faster and accurately, so we plan to create models that will classify multiple lung diseases like COVID-19 and Atelectasis. This will allow us to reduce the turn-around-times for diagnosis of new diseases considerably. There are a number of challenges that we foresee, the datasets are highly imbalanced so we will have to take corrective measures. Images from different X-Ray scanners have different radiographic contrast [10] this could have a negative impact on our model's results. In some cases, multi-level classification might be required as the patient might be affected by multiple diseases.

We will try different models to solve this problem and present a detailed comparison of the results. After evaluating the results, we will choose the best model in terms of both efficiency and accuracy. We will try to make use of data augmentation and transfer learning to tackle the small dataset problem and design our custom CNN which could get us better results than the existing architectures.

2. Image Dataset Selection

We will use three chest X-Ray datasets with different diseases as listed in Tab. 1 to train and evaluate our models.

Dataset	No. of Images	Classes	Image Size
Pneumonia & COVID-19 [11]	1,583 (Normal) + 576 (COVID-19) + 4,273 (Pneumonia) = **6,432**	3	Variable
Pneumonia [8] [15]	1,583 (Normal) + 1,493 (Viral Pneumonia) + 2,780 (Baterial Pneumonia) = **5,856**	3	224 x 224
Chest X-Ray8 [16] [17]	60,190 (Normal) + 16,610 (Infiltration) + 8,284 (Atelectasis) = **85,084**	3	1024 x 1024

Table 1. Shortlisted Datasets.

3. Possible Methodology

As our datasets are from different sources, we will explore different pre-processing techniques like histogram equalization, Gaussian blur [3], updating contrast and image flipping in PyTorch using functions like normalize, gaussian_blur and equalize [12] to make training easier for our CNN. We will explore several neural network architectures like VGG 16 [13], Inception V3 [14], Resnet [6], and produce our own Custom CNN model as well with different depth, size of kernel, strides and types of layers. To train our models, we will use cross-entropy loss and experiment with optimizers [5] like Stochastic Gradient Descent, Adam, AdaDelta [7] etc. to select the optimizer that gives us a lower loss with less epochs. To ensure that our model does not get stuck at a local minima, we will try different learning rate decay methods. While training, we will tweak different hyperparameters like epoch, activation functions, and batch size to ensure that we get a well performing model. As the size of our dataset is small, we plan on experimenting with techniques like data augmentation and transfer learning to improve our results. To find the best hyperparameters, we will perform ablation studies and try to make use of Bayesian hyperparameter optimization [1].

Given that our datasets are highly imbalanced, we will not rely on accuracy and will explore different metrics like confusion matrix, ROC Curve and F-Measure [2]. We will also consider the FLOPs of our models as one of the key metrics. To explain the results of our models, we plan on using SHAP [9] and GradCAM [4] which will help us diagnose our models and also help end users gain more confidence in our model's decisions.

We plan to make use of the Google Colab platform to train our classifiers. In order to seamlessly collaborate and transfer changes from our local system to the training system on Colab we will make use of git and Google Drive.

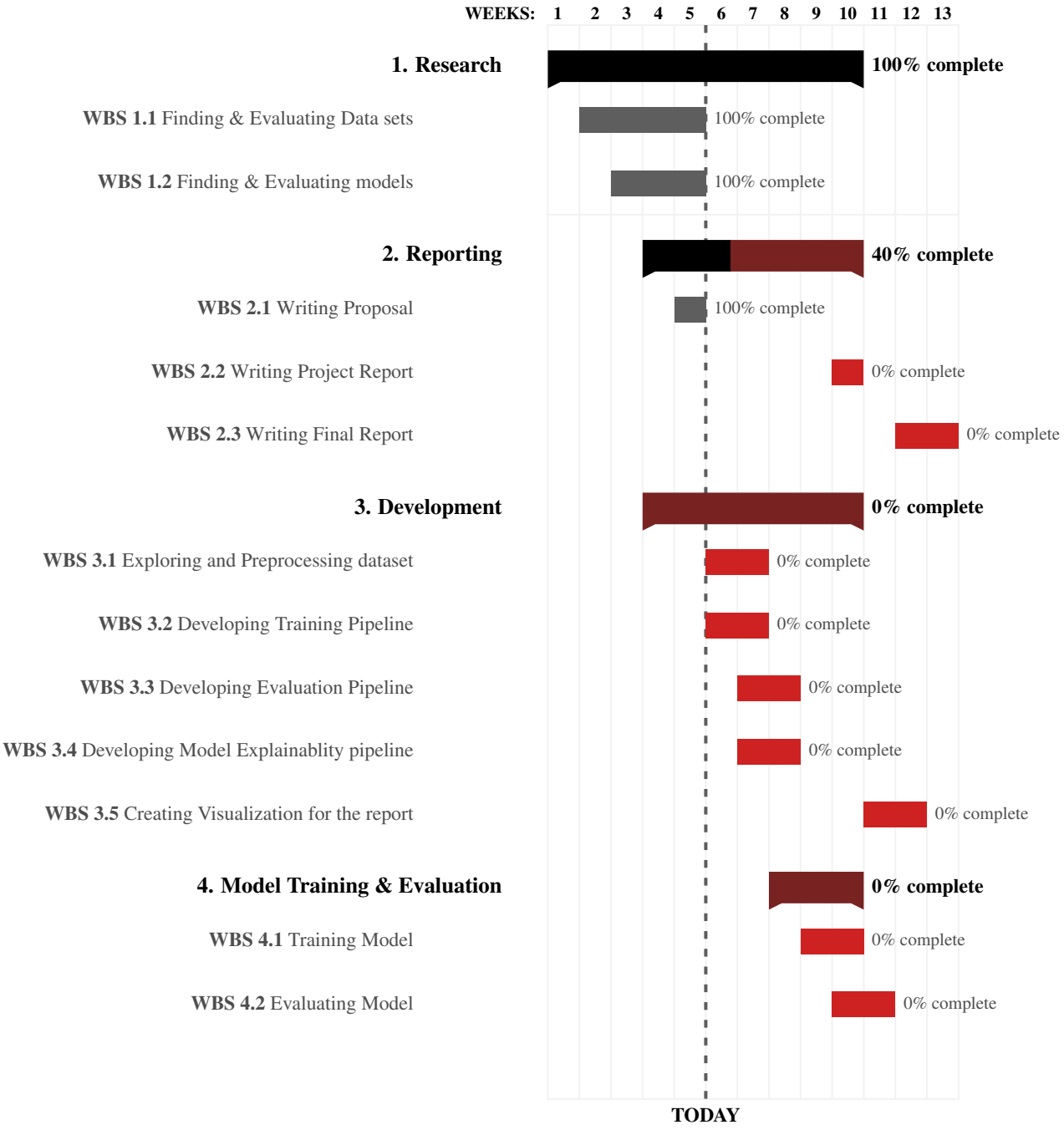

Figure 1. Gantt Chart.

References

[1] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020. 1

[2] Jason Brownlee. A gentle introduction to the fbeta-measure for machine learning. https://machinelearningmastery.com/fbeta-measure-for-machine-learning/, 2020. 1

[3] Agata Giełczyk, Anna Marciniak, Martyna Tarczewska, and Zbigniew Lutowski. Pre-processing methods in chest x-ray image classification. *Plos one*, 17(4):e0265949, 2022. 1

[4] Jacob Gildenblat and contributors. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam, 2021. 1

[5] Ayush Gupta. A comprehensive guide on deep learning optimizers. https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/, 2021. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[7] Ibrahem Kandel, Mauro Castelli, and Aleš Popovič. Comparative study of first order optimizers for image classification using convolutional neural networks on histopathology images. *Journal of imaging*, 6(9):92, 2020. 1

[8] Daniel Kermany, Kang Zhang, Michael Goldbaum, et al. Labeled optical coherence tomography (oct) and chest x-ray images for classification. *Mendeley data*, 2(2), 2018. 1

[9] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1

[10] Andrew Murphy. Radiographic contrast. https://radiopaedia.org/articles/radiographic-contrast, 2022. 1

[11] Prashant Patel. Chest x-ray (covid-19 & pneumonia). https://www.kaggle.com/datasets/prashant268/chest-xray-covid19-pneumonia, 2020. 1

[12] PyTorch. Transforming and augmenting images. https://pytorch.org/vision/stable/transforms.html. 1

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 1

[15] Tolga. Chest x-ray images. https://www.kaggle.com/datasets/tolgadincer/labeled-chest-xray-images, 2020. 1

[16] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017. 1

[17] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Nih chest x-rays. https://www.kaggle.com/datasets/nih-chest-xrays/data, 2017. 1