**Lung Virus Classifier: Classification of Lung X-Rays as COVID-19 or**

**Pneumonia via CNN and KNN Algorithms**

CMPT 340 Biomedical Computing - Ghassan Hamarneh

**Team** Sabrina Dalen <[sldalen@sfu.ca](mailto:sldalen@sfu.ca)>, Jin Lu <[jla388@sfu.ca](mailto:jla388@sfu.ca)>,

Kate Hui Wu <[wuhuiw@sfu.ca](mailto:wuhuiw@sfu.ca)>, Carina Caijie Zhao <[caijie\_zhao@sfu.ca](mailto:caijie_zhao@sfu.ca)>, Natalie Huiyi Zou <[huiyi\_zou@sfu.ca](mailto:huiyi_zou@sfu.ca)>

**Motivation** As COVID-19 spreads worldwide being declared a pandemic, virus identification from X-Rays may assist medical research and diagnosis.

**Goals** This project aims to classify lung X-rays as healthy or diseased with COVID-19 or pneumonia using algorithms for deep-learning and non-deep learning. The result will be a MATLAB program comparing the results of each method and returning a probable diagnosis.

**Steps** Our team will follow the project timeline to deepen our knowledge of the algorithms to analyze the selected datasets of lung X-rays. With our development environment setup, we will collaborate to extract useful data from the images, applying CNN for deep-learning and Random Forest and KNN for non-deep learning. We will use Python and other libraries to create a MATLAB program [3], [4], [5]. We will document our method and findings in a report and create a video demo of our results.

**Timeline Week 1 (Mar 29 - Apr 4)**

* Project Analysis and Requirements – Confirm project scope and desired results; determine software tools and tasks required
* Gather Research – Explore algorithms and existing techniques

**Week 2 (Apr 5 - Apr 11)**

* Design Specs – Detail code requirements to implement chosen methods. Non-deep learning: build decision tree using RF. Deep learning: reshape images (extract data, define sample, build model)
* Architecture Overview – Create high-level outline of code structure
* Report – Setup draft template outlining key points for each section

**Week 3 (Apr 12 - Apr 18)**

* Software Development – Main code production
* Code Review and Refactoring (optimize performance of classifier)
* Component Testing – Perform initial unit testing on helper functions
* Video Requirements & Design – Choose software, outline format
* Video Development – Create sample, identify production issues
* Report – Draft majority of sections and update results to date

**Week 4 (Apr 18 - Apr 21 due)**

* Production, Quality Assurance, and Usability Testing
* Video Development – Create demo
* Report – Complete abstract summary, final edits, and proofread

**Team Duties** Our team will share the roles of Researcher, Software Developer, and Quality Assurance Tester. Key roles are as follows:

* **Sabrina Dalen** – Documentation Lead, Code Reviewer, Copy Editor
* **Jin Liu** – Software Architect, Technical Lead
* **Kate Hui Wu** – Meeting Organizer, User Acceptance Lead
* **Carina Caijie Zhao** – Functional Manager
* **Natalie Huiyi Zou** – Project Manager, Team Lead

**Anticipated Techniques** – Implementing these algorithms is challenging. Solution:

**Problems** Split topics for individual research and share findings as a group.

**Accuracy** – Image classification accuracy may be low: 1) Imbalanced datasets and lack of images may not provide enough information for model learning. Solution: Use more samples to build a better model.

2) The choice of k value for KNN may greatly affect the algorithm. Solution: Use different k values to discover which value works best.

**Time Limits** – Classifying images may be time-consuming due to non-uniform images with large file sizes. Solution: Reduce and normalize the images during pre-processing.

**Material** **Datasets** – Chest X-ray images of COVID-19 from a GitHub repository and similarly of pneumonia from a kaggle dataset [1], [2].

**Software** – Collabedit (online group code editor), MATLAB, Python

**Algorithms** – Convolution Neural Network (CNN), Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM) [3], [4].

**Libraries** – OpenCV (to pre-process images), Tensorflow or Keras (CNN), Sklearn for Python (KNN)

**Anticipated** A MATLAB program that accepts one or more X-ray images of lungs as

**Results** input and returns the following output:

* A table displaying the percent similarity of the input to 1) a healthy lung, 2) a lung with COVID-19, and 3) a lung with pneumonia, with a final column suggesting the probable diagnosis. The results shown will be an average of the analysis from the two different algorithms using deep-learning and non-deep learning.
* A graph comparing the average difference in results from the two algorithms and their range of error.
* A figure grouping the input images into the three categories above to visually represent their similarity to each other and to selected sample models.

**References**

[1] ieee8023, “ieee8023/covid-chestxray-dataset,” GitHub, 30-Mar-2020. [Online]. Available:   https://github.com/ieee8023/covid-chestxray-dataset/tree/master/images. [Accessed: 02-Apr-2020].

[2] P. Mooney, “Chest X-Ray Images (Pneumonia),” Kaggle, 24-Mar-2018. [Online]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia. [Accessed: 02-Apr-2020].

[3] E. Allibhai, “Building a Convolutional Neural Network (CNN) in Keras,” Medium, 15-Nov-2018. [Online]. Available: https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5. [Accessed: 02-Apr-2020].

[4] W. Koehrsen, “An Implementation and Explanation of the Random Forest in Python,” Medium, 31-Aug-2018. [Online]. Available: https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76. [Accessed: 02-Apr-2020].

[5] O. Harrison, “Machine Learning Basics with the K-Nearest Neighbors Algorithm,” Medium, 14-Jul-2019. [Online]. Available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761. [Accessed: 02-Apr-2020].