is to extract the lung contours from the CXR images prior to classification. The motivation behind this is to have the classifier focus on the specific lung regions versus the whole CXR image. In addition, another improvement we propose is to modify each base model in the ensemble to process the segmented CXR image in K patches as shown in Figure 1. To accomplish this, the steps we will follow are

1. Extract the lung contours from the CXR images using Segment Model.
2. Train k-patch classifiers by starting with pre-trained model from ImageNet. Divide extracted lung contours into k-patches. Run each patch through model to generate classification, prediction is calculated based on majority voting. Update shared weights for each of the k models.
3. Construct improved FLANNEL architecture by using extracted lung contours as input and k-patch classifiers used as the base models.
4. Train Improved FLANNEL architecture. From input CXR images, extract the lung contours using Segment Model. Create k-patches of segmented CXR. Each k-patch classifier processes k-patches and produces predictions. Calculate weighted ensemble through neural weighting module. Compute prediction based on k-patch model predictions and weights. Compute focal loss and update neural weighting module weights. Continuously calculate metrics to measure performance such as but not limited to: accuracy, recall, precision, F1 and ROC.
5. Perform ensemble (combining multiple K patch classifiers) to calculate the weighted ensemble.
6. Get the prediction and compare with the ground truth.
7. Apply Focal Loss to train the model (improve weights).
8. Test the model on the 'test' dataset to calculate accuracy, precision, F1 measure, ROC and other metrics.

3.3 Performance Analysis

We will record the classification accuracy for 4 classes using F1-score. We compare the F1-score accuracy for COVID-19 vs other classes for five base learners, FLANNEL with ensemble strategies voting and stacking, FLANNEL with cross entropy loss, FLANNEL with re-sampling and FLANNEL with k-patch improvement. We use receiving operating characteristic (ROC) curve and precision-recall (PR) curve to display classification performance against threshold. Finally we will provide visual description of FLANNEL and proposed improvement performance using confusion matrix.

4. EXPERIMENTAL SETUP

We are planning to use FLANNEL source code as our baseline and enhance on top of it. Our codebase would be using below software/python packages as shown in Table 3

As FLANNEL model requires significant compute and GPU resources (3 NVIDIA Tesla P100 GPUs). We would be utilizing AWS EC2 service with instance type p3.8xlarge which provides 4 NVIDIA Tesla V100 GPUs along with 32 core CPU and 64GB RAM.

For data analysis and exploration, we are going to use Google Colab.

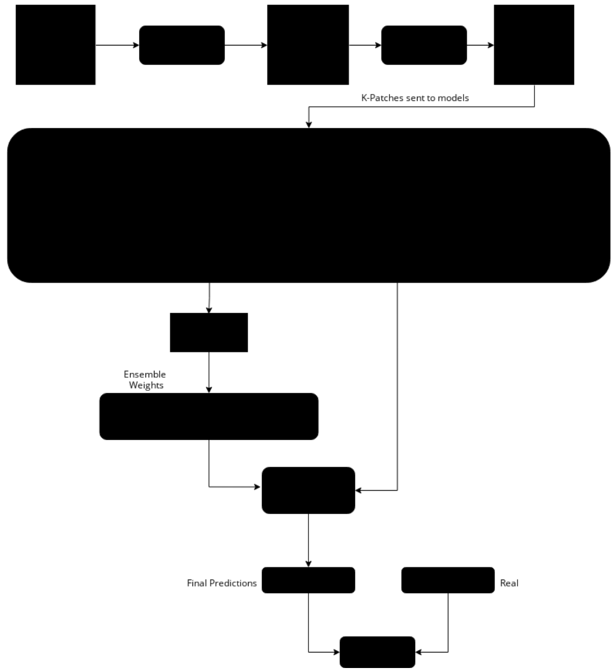

Figure 1: FLANNEL Improvement

Table 3: Software/Tools used

Software/Tool	Version
Python	3.8.5
numpy	1.20.2
torch	1.8.1
torchvision	0.9.1
matplotlib	3.4.0
scikit-learn	0.24.1
pandas	1.2.3

Figure 2: FLANNEL Improvement

5. REFERENCES

[1] Y. Oh, S. Park, and J. C. Ye. Deep Learning COVID-19 Features on CXR Using Limited Training Data Sets. *IEEE Trans Med Imaging*, 39(8):2688–2700, 08 2020.