# Pairwise Meta Learning Pipeline: Classifying COVID-19 abnormalities on chest radio-graphs

Sourajit Saha[a], Michael Morris[b], Babak Saboury[b], Phuong Nguyen[a], Yelena Yesha[a], Yaacov Yesha[a], Aryya Gangopadhyay[a], and David Chapman[a]

[a]University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD, 21250
[b]National Institutes of Health Clinical Center, 9000 Rockville Pike, MD 21201

## ABSTRACT

Viral and bacterial pneumonia exhibit a striking similarity with COVID-19 infections on chest radio-graphs. Therefore, initiating the right treatment before the virus causing significant health damage is contingent upon detecting and classifying the type of pneumonia from chest radio-graphs with a increased and faster efficacy. The objective of this study is to have a Computer Aided Diagnosis (CAD) system which is powered by deep learning and investigate methods that help us increase the classification accuracy for such a task. In this study, we use a labeled image dataset provided by Society for Imaging Informatics in Medicine for a kaggle competition that contains chest radio-graphs. And the task of our proposed CAD is to categorize between negative for pneumonia or typical, indeterminate, atypical for COVID-19. We propose a novel ensemble meta learning pipeline that juxtaposes multiple Convolutional Neural Network (CNN) architectures for the task of image classification on the aforementioned dataset. We devise a method that enables our proposed meta learning pipeline to ensemble multiple heavy weight CNN architectures and we further conduct a ablation study to report that our proposed meta learning pipeline outperforms all of the individual CNNs that we train on this dataset. We perform a 5-fold validation on the training set with provided labels and do a train-test split to run our experiments to confirm our hypothesis. Applying our proposed meta learner on top of 14 different trained CNN models we achieve a test accuracy of 99.23 % and our proposed CAD classifies chest radio-graph images in real time.

**Keywords:** COVID-19 Classification, Image Classification, Computer Aided Diagnosis (CAD), Deep Learning, Chest X-Ray Image Classification, Ensemble, Meta Learning, Computer Vision

## 1. DESCRIPTION OF PURPOSE

The purpose of this study is to devise a Computer Aided Diagnosis (CAD) system that is able to detect COVID-19 abnormalities from chest radio-graphs with increased efficiency and accuracy.

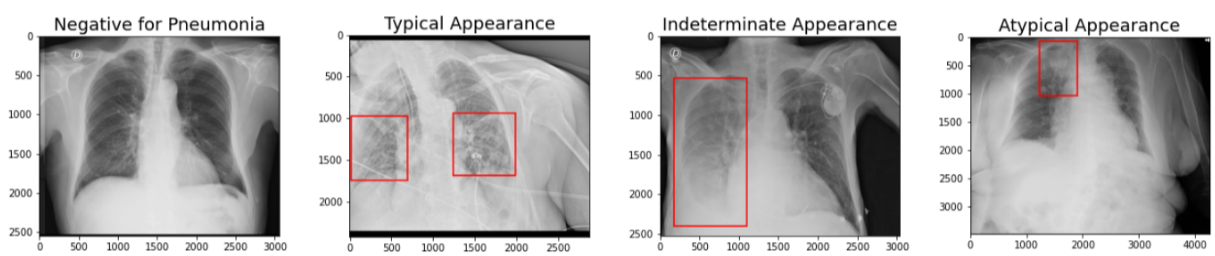


Figure 1. Sample of the Chest radio-graph dataset for detection of pneumonia.

We investigate a novel deep learning based ensemble model to classify the category of pneumonia from chest X-ray images. We use a labeled image dataset provided by Society for Imaging Informatics in Medicine for a kaggle competition that contains chest radio-graphs. And the task of our proposed CAD is to categorize between

Further author information: (Send correspondence to Sourajit Saha)
Sourajit Saha: E-mail: ssaha2@umbc.edu, Telephone: 1 443 889 4165

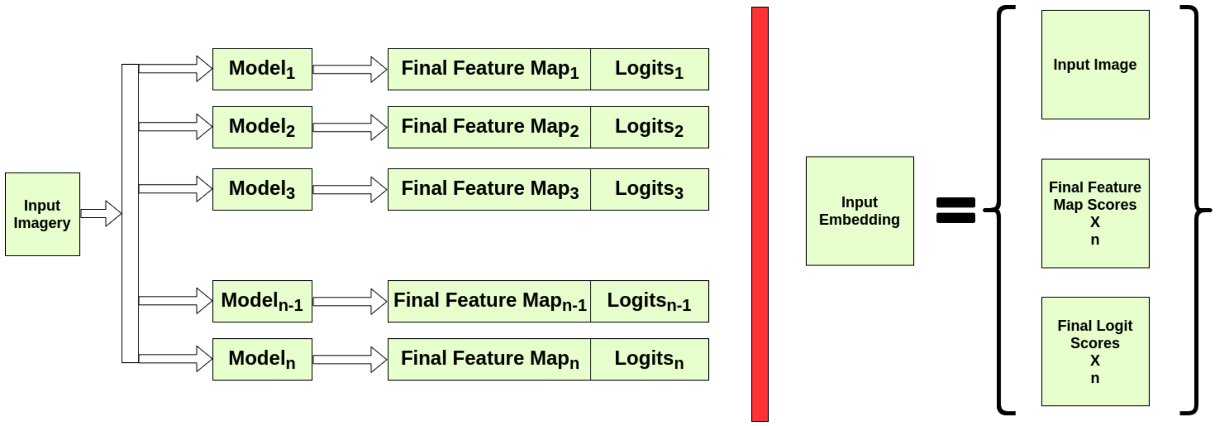


Figure 2. Extraction of final convolutional feature maps and logit scores from n models and embedding them in pairwise fashion along with the image.

negative for pneumonia or typical, indeterminate, atypical for COVID-19. The training set (with labels publicly available) of this dataset contains 6334 images belonging to 4 classes. Fig. 1 depicts a sample of the dataset. This dataset contains all the images in DICOM format which we convert into numpy format and resize all the images into $400 \times 400$. Furthermore, we experiment on the efficacy of our proposed ensemble method. Accordingly, we perform a ablation study to confirm that our proposed pipeline drives the classification accuracy higher and also compare our ensemble technique with the existing one both quantitatively and qualitatively.

## 2. METHOD

Firstly, we resize the 6334 images into $400 \times 400$. Then, we split them into 80:20 train-test ratio. Thereafter, we train 14 different CNN models on our train split (5067 images). We train each of the 14 CNN models with a 5-fold validation and report the average classification accuracy in Table 1. The training of all of these models are independent and after training all of these models for the aforementioned number of epochs in Table 1. Moreover, while training each of these models, five times for each fold of the train split, we initialize the weights every time with the image net pre-training weights which reduces the probability of gradient degradation while training. Sequentially, for our meta learner we propose a novel input embedding. After all the n=14 models are trained, we run each image through all n=14 models and embed all n=14 final convolutional feature maps and logit scores from the fully connected layer in a pairwise fashion with the corresponding image as depicted in Fig. 2.

After the completion of embedding, we propose a novel meta learning pipeline which leverages the pairwise

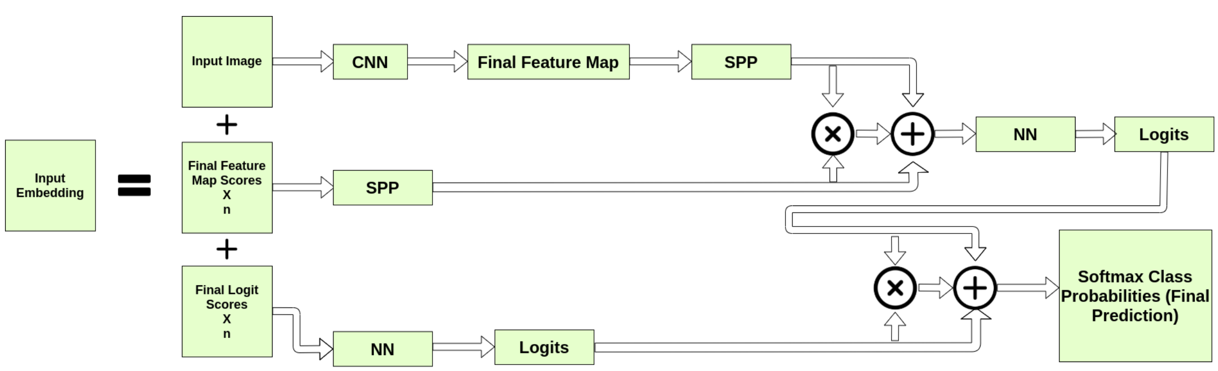


Figure 3. Proposed Meta Learning Pipeline: Training on proposed pairwise feature and image embedding to produce final prediction tensor.

embedding that we devised, hence the title of our proposed method. On Table 1 we observe that Efficient Net-b7 model yields the highest average 5-fold classification accuracy for our train-test split and therefore we chose this model as the backbone for our meta learner CNN. As detailed in Fig. 3 we run the images through Efficient Net-b7 to produce the final convolutional feature map which we then run through a Spatial Pyramid Pooling (SPP) layer. The second part of our embedding tensor which is the set of n final convolutional feature maps is then run through another SPP layer. This step is essential because different models yield final convolutional feature maps of different spatial and channel size and running through a SPP layer ensures that the final outputs are of consistent window size. Afterwards, we concatenate the two SPP signals using a series of operations. Let's call the SPP signal generated by the CNN $\alpha$ and the SPP signal generated from the embedding $\beta$. The concatanated signal, $\tau$ then follows Equation. 1. We concatenate in this manner because, if gradients for one branch decay the cascade of signal will act as a residual layer to prevent gradient degradation. For example, if $\alpha \approx 0$ then $\tau = 0 + \beta(1 + 0) = \beta$ and if $\beta \approx 0$ then $\tau = 0 + \alpha(1 + 0) = \alpha$.

$$\tau = \alpha + \beta + \alpha \times \beta = \alpha + \beta(1 + \alpha) = \beta + \alpha(1 + \beta) \tag{1}$$

We then run the concatenated signal through a Neural Network of two layers to produce the logit scores. And finally we run the embedded n-stack logit scores through a 3 layer neural network to produce another set of logits which we concatenate with our existing logits using the signal propagation consistent with Equation. 1. And finally, we run our concatenated logits through a softmax activation to produce class probability scores.

## 3. RESULTS

As shown in Table 1, the best performing model on our train-test split is the Efficient Net b7 model. To ensure that the learners do not overfit, we trained all 14 models on a 5-fold cross validation and the result reported in Table 1 is the average of all the folds. Now in Table 2 we report the performance of 5-fold training on the train-test split using our proposed joint pairwise meta learning pipeline. Comparing between Table 1 and Table 2 we observe that our proposed meta learning pipeline outperformed all the individual models that were trained. And our final Test accuracy is 99.23 % and our proposed CAD can diagnose a single image with a average of 0.12 s which is real time. Fig. 4 depicts the training loss on three variant of Efficient Net models.

Table 1. Training Performance on n=14 Models.

| Model | Number of parameters | Number of Epochs | Test Accuracy |
|---|---|---|---|
| wide resnet50 v2 | 68,887,244 | 30 | 95.778 % |
| googlenet | 6,628,908 | 60 | 96.368 % |
| resnext50 32X4d | 25,032,908 | 50 | 96.968 % |
| shufflenet v2X1 | 2,282,608 | 60 | 92.137 % |
| resnet50 | 25,561,036 | 50 | 96.984 % |
| resnet152 | 60,196,812 | 30 | 96.652 % |
| resnet18 | 11,693,516 | 60 | 96.921 % |
| squeezenet1X0 | 1,248,424 | 150 | 79.239 % |
| densenet161 | 28,681,000 | 50 | 94.936 % |
| mobilenet v3 large | 5,483,032 | 100 | 93.684 % |
| mnasnet1X0 | 4,383,312 | 90 | 83.407 % |
| Efficientnet-b0 | 5,292,552 | 100 | 94.542 % |
| Efficientnet-b4 | 19,345,620 | 80 | 97.812 % |
| Efficientnet-b7 | 66,351,964 | 50 | 98.336 % |

Table 2. 5-Fold training of our Proposed Pairwise Meta Learning Pipeline.

| Fold | Number of Epochs | Training Time (s) | Train Accuracy | Test Accuracy |
|---|---|---|---|---|
| Fold-1 | 40 | 608 | 97.561 | 98.715 |
| Fold-2 | 40 | 612 | 96.532 | 99.321 |
| Fold-3 | 40 | 597 | 98.817 | 99.453 |
| Fold-4 | 40 | 620 | 99.105 | 99.273 |
| Fold-5 | 40 | 611 | 99.458 | 99.392 |
| Average | 40 | 609 | 98.295 | 99.230 |

Furthermore, in our main manuscript we shall report mode details about our training pipeline, literature review on ensemble classifiers, comparative analysis of performance by our proposed ensemble meta learning training pipeline and existing meta learning methods. We shall also discuss in greater details how our proposed method is novel and more time and memory efficient. And to further strengthen our argument on the improved efficacy for the task of COVID-19 abnormality classification from chest radio-graph imagery we will report more standard metrics such as confusion matrix, sensitivity, specificity and AUC score.

Furthermore, the methods proposed in this study and the observations and results reported in this article have not been submitted for any other private or public scholarly or non-scholarly article.

## 4. ACKNOWLEDGMENTS

We would like to thank the Society for Imaging Informatics in Medicine (SIIM) for providing the public dataset on kaggle for the SIIM-FISABIO-RSNA COVID-19 Detection competition. We would also like to thank NFS as this study is funded by NFS RAPID Grant for University of Maryland Baltimore County.

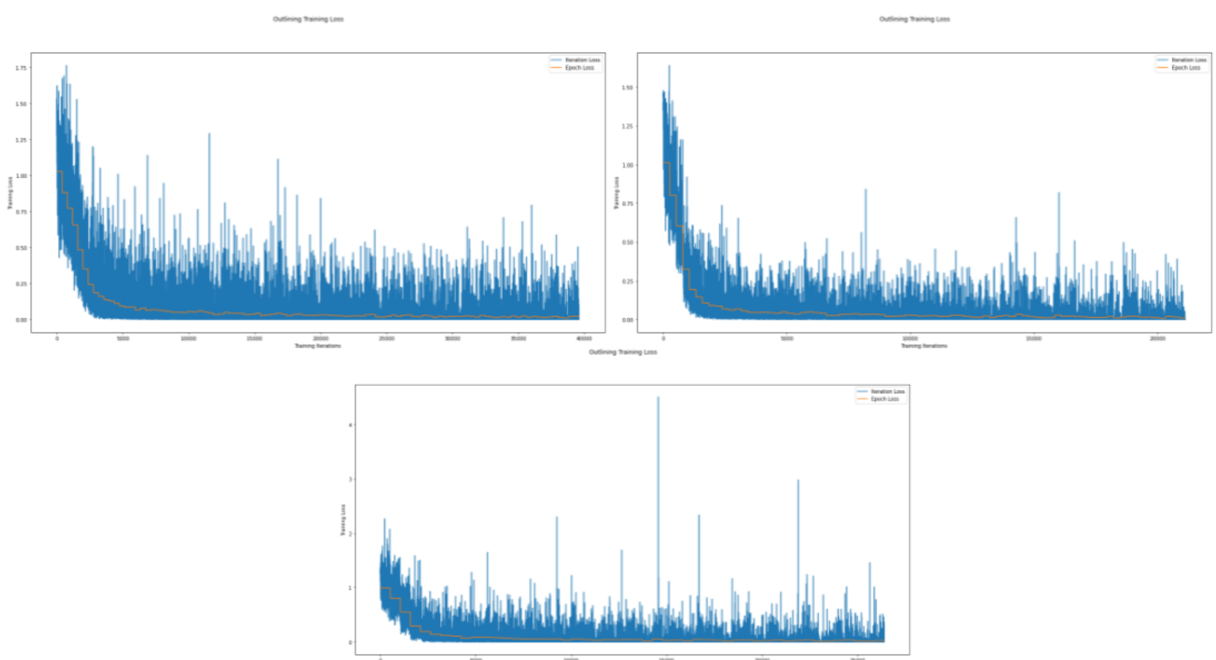

Figure 4. Training Loss: Efficientnet-b0 on Top Left, Efficientnet-b4 on Top Right and Efficientnet-b7 on bottom.