Respiratory Disease Detection using Meta Learning

Akshaya Brian Tauro - A20502097
CS 584 - Machine Learning

ABSTRACT

Labelled images for detection of novel diseases are in small amount due to less affected patient population and clinical expertise are limited to label images. Deep networks implementation for image based diagnosis need to be intensive enough to predict novel diseases with fewer labelled images. When covid was initially discovered dataset was limited and was hard to differentiate with other respiratory disease dataset, where standard training approaches lead to poor categorizing on classes in such scenarios, which will lead to long tailed class distribution. This project addresses the issue of disease detection and quick model adaptation problems in the case of small dataset and long tailed distribution using meta learning. This involves training a neural network on few shot image classification tasks using the initial dataset and later testing on unseen data. Proposed method is named as 'Respiratory Disease Detection using Meta Learning' which uses gradient based Reptile for detection of few common Respiratory diseases, for this project considered two common respiratory diseases like Tuberculosis, Pneumonia and Covid which is a novel disease. To evaluate the effectiveness of this approach on publicly available Tuberculosis, Pneumonia and Covid dataset from kaggle, and obtain significant performance improvement using trained models with less labelled dataset.

1 RELATED WORK

[4] The Meta-Derm Diagnosis: Few shot Skin Disease Identification using Meta-Learning research project gives a 5-shot AUC of 86.8 percentage, compared to the DAML(difficulty-aware meta-learning(DAML) which give AUC of 83.3 percentage. The reason for their increase in performance was due to expressive 6-layer CNN, the use of Reptile algorithm and the integration of G-convolutions which allowed the network to achieve in-variance to the different transformations present in skin lesion images with very fewer images.

The figure 1 from [4] depicts the gradient-based meta learning pipeline used for disease classification from skin lesion images, and they used Reptile as the meta-learning algorithm for their analysis where theta is the operator corresponding to Adam optimizer or SGD, and that updates theta using k mini-batches in data sampled from each task.

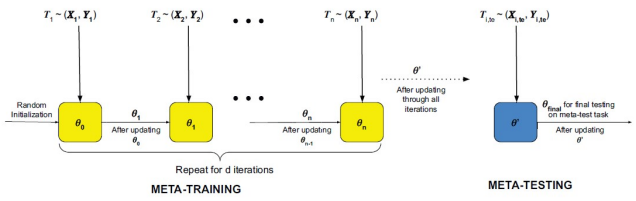

Figure 1: [4] Pipeline for gradient-based meta-learning

[4] Prototypical Networks, which is a distance metric based meta learning technique which computes a prototype vector as representation of each class, this is a mean vector which has embedded support instances that belongs to its class. Here, subset of N classes are randomly selected for one training task. For each training task, support set are created by sample examples from selected classes.the distribution over predicted labels for a query sample is computed using softmax over negative distances to the prototypes in the embedding space using a distance function, then parameters theta are updated to improve the likelihood computed on query set, which is computed using estimated prototypes.

[4]By exploiting symmetries, Group Equivariant CNNs have achieved state-of-the-art results on a variety of datasets like rotated MNIST and CIFAR10. Due to the rotations and reflections present in the skin lesion data, G-convolutions enhance the performance significantly.

[4] Meta-DermDiagnosis gives a 5-shot AUC of 86.8% The reason for this increase in performance is the more expressive 6-layer CNN, the use of Reptile algorithm and the deployment of G-convolutions which allow the network to achieve invariance to the different transformations present in skin lesion images with much fewer images.

2 PROBLEM STATEMENT

Annotations of novel diseases is very costly in-terms of labor intensive, time, prone to error even when labelled by experienced doctors. For automated diagnosis when deep models are applied, networks tend to fail due to the drawback of limited availability of annotated data since ability to generalize well is not noticed due to over-fitting issue, and also training of deep networks repeatedly is time-consuming in healthcare domain. However, to address this issue, transfer learning were proposed but here the drawback is sufficient labeled data should be available in the target domain.

3 PROPOSED SOLUTION

To address the issue with less annotated data, technique named [2]meta-learning have emerged, with less training examples this technique can adapt to the new environment and tasks.

In this project, i explored meta-learning based few-shot approach like [2]Reptile algorithm which is the gradient based method to detect novel disease like COVID19, and other common respiratory diseases like Pneumonia and Tuberculosis. I named the proposed network $Meta - RespiratoryDiseaseDetection$ which was motivated while reading [4] 'Meta-DermDiagnosis: Few-Shot Skin Disease Identification using Meta-Learning' paper. This consists of a meta-learner, with the initial set of class labels the neural network is trained to solve a number of few-shot image classification tasks.

An overview of the $Meta - RespiratoryDiseaseDetection$ pipeline that includes meta-training and testing stage is shown in the [4] Figure 1 which depicts the gradient-based meta learning pipeline used for classification respiratory disease images, and Reptile as

the meta-learning algorithm for the analysis where theta is the operator corresponding to SGD, and that updates theta using k mini-batches in data sampled from each task. In the Following subsection, described reptile algorithm in detail.

3.1 Reptile Algorithm: Gradient-based meta-learning

[4]In supervised learning, the common practice is to learn from a set of labeled examples, while meta-learning learns from a set of labeled tasks, each represented as a labeled training set and a labeled testing set. The learning algorithm trains for a representation that can be quickly adapted to a new task, via a few gradient steps. The meta-learner seeks to find an initialization that is not only useful for adapting to various problems, but also can be adapted quickly in a small number of steps and efficiently using only a few annotated examples. [2]Reptile learns an initialization for the parameters of a neural network model, such that when we optimize these parameters at test time, learning is fast—i.e., the model generalizes from a small number of examples from the test task. In the last step, instead of simply updating θ in the direction $\hat{\theta} - \theta$, we can treat $(\theta - \hat{\theta})$ as a gradient and plug it into an adaptive algorithm. Reptile[2][4] as the meta-learning algorithm for analysis in this project. It is a first-order gradient-based meta-learning algorithm which learns an initialization for the parameters of a neural network model, such that when we optimize these parameters at test time, learning is fast i.e., the model generalizes using a small number of examples from the test task. The Reptile algorithm is as follows:

1: Initialize $\boldsymbol{\theta}$, the vector of initial parameters
2: **for** $iteration = 1, 2, \ldots$ **do**
3: Sample task T, corresponding to loss L_T on weight vectors $\tilde{\boldsymbol{\theta}}$
4: Compute $\tilde{\boldsymbol{\theta}} = U_T^k(\boldsymbol{\theta})$, denoting k SGD or Adam steps
5: Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \epsilon(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta})$, where ϵ is the stepsize parameter
6: **end for**

Figure 2: [2] [4] Reptile Algorithm

[2][4]In Reptile algorithm, $U_k^T(\theta)$ is the operator (e.g. corresponding to SGD optimizer) that updates θ using k mini-batches on data sampled from T.

To evaluate the performance of the implemented method using Reptile algorithm on the publicly available dataset from [3]Kaggle which consists of Covid Tuberculosis, Pneumonia and Normal chest X-rays, and compared their performance against the pre-trained transfer learning baseline.

4 IMPLEMENTATION DETAILS

Chest X-Ray (Pneumonia,Covid-19,Tuberculosis) dataset in [3]kaggle had vast difference between the number of images that were split in to train, test and validation. So, i did split the dataset as described in the Figure 4.

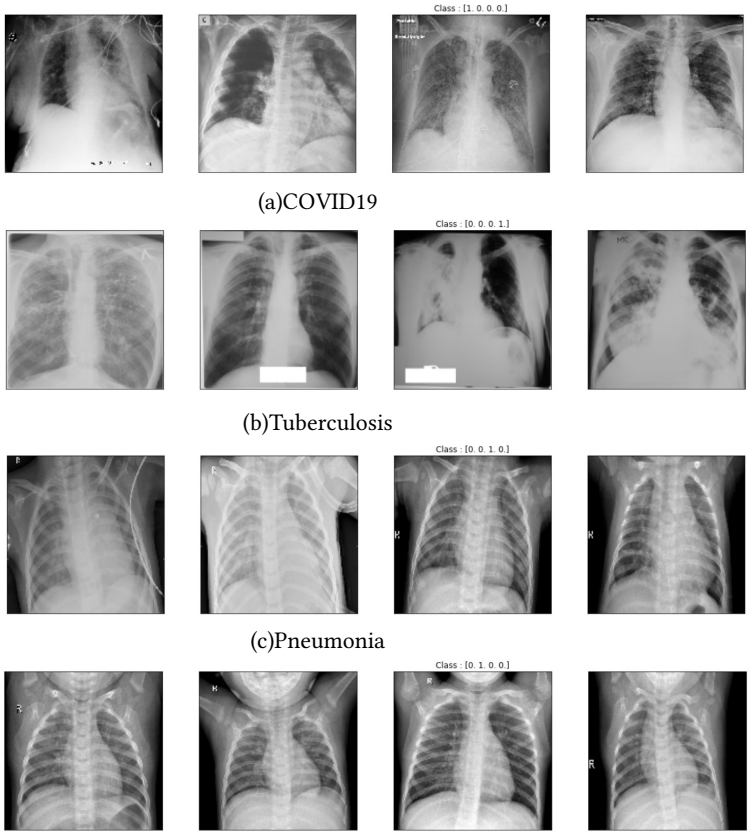

Figure 3: [3] Sample images of (a)COVID19, (b)Tuberculosis, (c)Pneumonia and (d)Normal from the respiratory diseases dataset used in this project

Complete data set(4670 Images) was used to verify the results on the implemented CNN(Convolutional neural network) and Transfer Learning.

CHEST X-RAY DATASET (4675 IMAGES)		
TRAIN (3053)	VALIDATION (787)	TEST (830)
COVID19 (368)	COVID19 (100)	COVID19 (106)
NORMAL (1080)	NORMAL (268)	NORMAL (234)
PNEUMONIA (1135)	PNEUMONIA (288)	PNEUMONIA(390)
TUBERCULOSIS (470)	TUBERCULOSIS (131)	TUBERCULOSIS (100)

Figure 4: Original/Complete Dataset Distribution

Using meta dataset(20 Images/class) which consists of only fewer images was also used to verify the results on CNN(Convolutional neural network) and Transfer Learning. Refer Figure 5 for the data split visualization.

META DATASET (80 IMAGES)		
TRAIN (40)	VALIDATION (20)	TEST (20)
COVID19 (10)	COVID19 (5)	COVID19 (5)
NORMAL (10)	NORMAL (5)	NORMAL (5)
PNEUMONIA (10)	PNEUMONIA (5)	PNEUMONIA(5)
TUBERCULOSIS (10)	TUBERCULOSIS (5)	TUBERCULOSIS (5)

Figure 5: Fewer/Meta Dataset Distribution

	META DATASET (20)	COMPLETE DATASET (4670)
META LEARNING	YES	NO
CLASSICAL CNN	YES	YES
TRANSFER LEARNING	YES	YES

Figure 6: Total 5 experiments

Added 4-layer CNN for all of my experiments refer Figure 7,
1) first CNN layer consists of 32 filters of size 3 x 3, activation function as ReLU, followed by max pooling layer of size 2 x 2.
2) second CNN layer consists of 64 filters of size 3 x 3, activation function as ReLU, followed by max pooling layer of size 2 x 2.
3) third CNN layer consists of 128 filters of size 3 x 3, activation function as ReLU, followed by max pooling layer of size 2 x 2.
4) fourth CNN layer consists of 128 filters of size 3 x 3, activation function as ReLU, followed by max pooling layer of size 2 x 2.
Followed by flatten layer, dropout layer of size 0.5, dense layer with 512 units and activation function as ReLU, and for the last dense layer 4 units and activation function as 'softmax' since there are four classes.
The input to the network is an image of size 150 x 150 pixels, and transformed the image to gray-scale as color mode, as chest x-rays can we learnt well with gray scale images.

For implementation overview refer Figure 8.

The specific implementation details for the Reptile and Transfer Learning approaches are as follows:-

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 32)      320

 max_pooling2d (MaxPooling2D  (None, 74, 74, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 72, 72, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 128)      0
 2D)

 conv2d_3 (Conv2D)           (None, 15, 15, 128)       147584

 max_pooling2d_3 (MaxPooling  (None, 7, 7, 128)        0
 2D)

 flatten (Flatten)           (None, 6272)              0

 dropout (Dropout)           (None, 6272)              0

 dense (Dense)               (None, 512)               3211776

 dense_1 (Dense)             (None, 4)                 2052

=================================================================
Total params: 3,454,084
Trainable params: 3,454,084
Non-trainable params: 0
_________________________________________________________________
```

Figure 7: Model Summary

- IMAGE DIMENSION FOE ALL THE EXPERIMENTS : 150 X 150 X 1
- NEURAL NETWORK:
  - INPUT LAYER – CONV 2D (32 UNITS, 3 X 3) , ACTIVATION - RELU
  - HIDDEN LAYERS
    - CONV 2D (32 UNITS, 3 X 3), MAXPOOLING2D(2X2) , ACTIVATION - RELU
    - CONV 2D (64 UNITS, 3 X 3), MAXPOOLING2D(2X2) , ACTIVATION - RELU
    - CONV 2D (128 UNITS, 3 X 3), MAXPOOLING2D(2X2) , ACTIVATION - RELU
    - CONV 2D (128 UNITS, 3 X 3), MAXPOOLING2D(2X2), , ACTIVATION - RELU
    - FLATTEN
    - DROPOUT(0.5)
    - DENSE (512)
  - OUTPUT LAYER – DENSE(4), ACTIVATION – SOFTMAX
- METRIC - ACCURACY
- META LEARNING: OPTIMIZER – SGD
- CNN & TRANSFER LEARNING : OPTIMIZER - RMSPROP

Figure 8: Implementation Overview

Transfer Learning: Implemented transfer learning on meta dataset and also on complete dataset: In complete dataset : Considered train, validation and test split belongs to four classes. train - 3053 images, validation - 787 images and test - 830 images.
In Meta dataset : 20 images/class, train - 40, validation 5 and test - 5.
For compiling, metric used is accuracy, loss is equal to categorical crossentropy, optimizer as RMSprop and learningrate is equal to 1e-4

initial epoch as 30, but implemented early stopping with patience 2, with reference to 'val_acc' for monitoring
For the loaded conv-base, added VGG-16 and performed data augmentation.

Reptile: Considered train and test split only where 3053 images belongs to 4 Classes for training and 830 images belongs to four classes for testing. For meta-train considered 20. The value of k in the experiment is 4 indicating 4-shot. k here is the shot which is the number of examples. For accuracy computation, evaluation is done on all the data in test split. Note: Referred [1] while implementing, which is in keras where they have experimented with omniglot dataset.

To evaluate the predictions made by all the models implemented a *models − test* code where are 4 known images belongs to four different classes (covid19, normal, pneumonia, tuberculosis) are placed in a folder, where all the models(meta learning, transfer learning and CNN) are loaded and evaluated the predictions made by the model.

5 RESULTS

Find the overall results of all the five experiments in Figure 9, and with respect to the same the graphical visualization is in Figure 10, 11, 12 and 13.

	ITERATIONS	EPOCHS	TRAIN ACCURACY	TEST ACCURACY
META LEARNING - FEW SHOT	1000	-	25	50
CLASSICAL CNN - META DATA	-	1/15	20	25
TRANSFER LEARNING - META DATA	-	2/15	35	30
CLASSICAL CNN - COMPLETE DATASET	-	9/15	88	75
TRANSFER LEARNING -COMPLETE DATASET	-	15/15	88	86

Figure 9: Results

5.1 CNN Transfer Learning:

Looking at Figures 9-13, might be wondering why it is just 1/15 epochs for classical cnn - metadata, and 2/15 epochs for transfer learning metadata. The reason is because of the input data which is very less(meta dataset), due to which tensorflow warned out of data. Firstly, i did use the same model and hyperparameters for fewer dataset and complete dataset, where while training the model early stopping with patience 2 is passed to monitor on validation accuracy. Since, these experiments are to check against meta learning few shot performance, didn't work on hypertuning much considering the time limit.

5.2 Meta Learning - few shot approach :

In this approach noticed fair results even for 1000 iterations that is 50% accuracy on test data. If 2000 iterations were run, expected to get good results, since i use google colab non-pro account to run

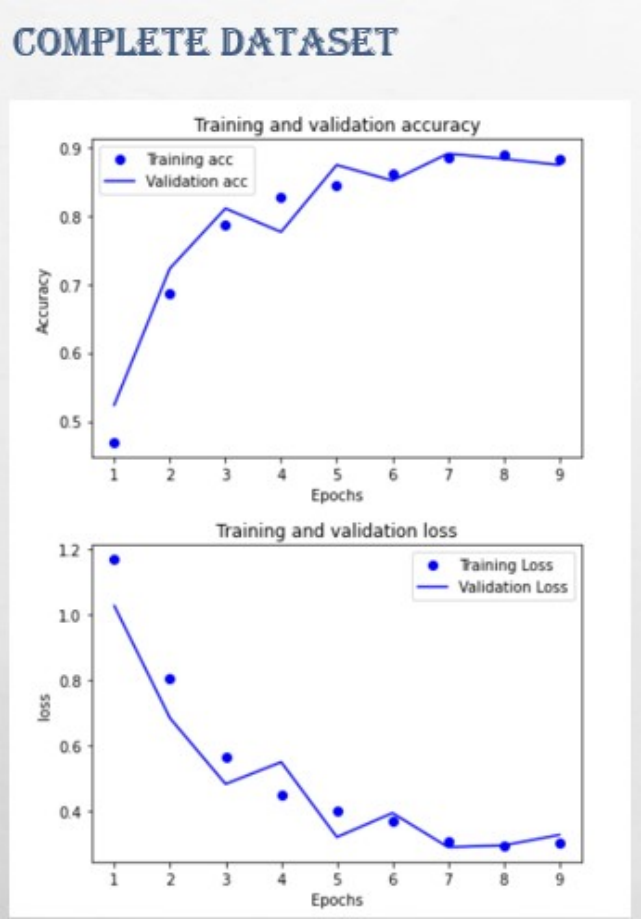

Figure 10: Results with Classic CNN on complete dataset

the experiments the time and space limitations was an hurdle. The graphical representation for the same is provided in Figure 14.

5.3 Prediction results - all the five models:

Overall results via the 'all_models-test.ipynb' code is in Figure 15, which evaluates the predictions made by all the five models on 4 images of chest x-ray images that belongs to each different class figures visualization for the same is in Figures 17-21.

6 CONCLUSIONS

With less annotated data META LEARNING performs better than transfer learning and classic cnn.
For low data scenarios, Meta learning approach shows good results.

7 FUTURE WORK

To get better results, would like to run more experiments.

I'm interested to explore state-of-the-art approaches in meta-learning techniques, and adoption of these in health care domain to detect the novel diseases.

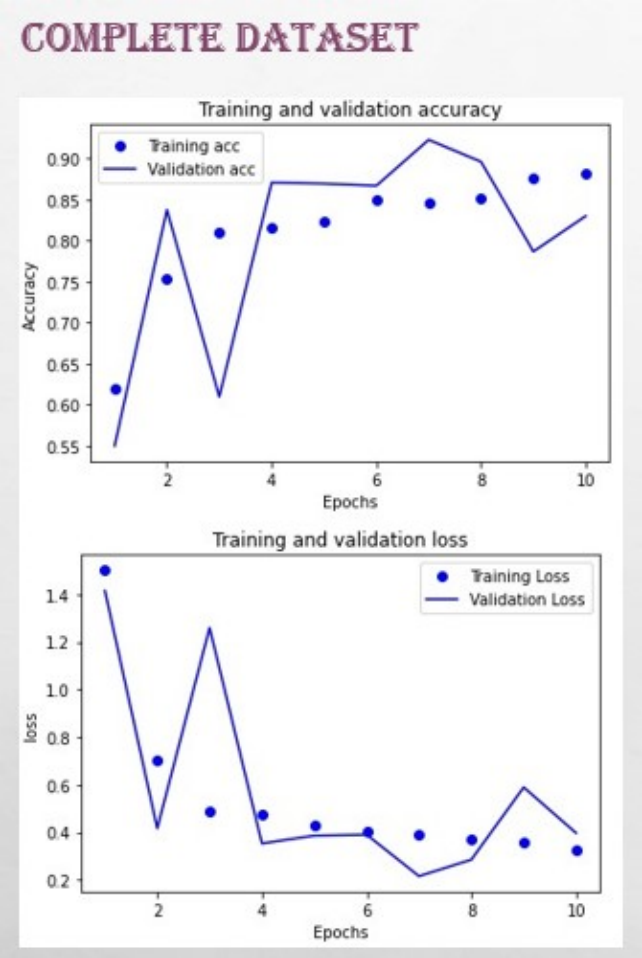

Figure 12: Results with Transfer Learning on complete dataset

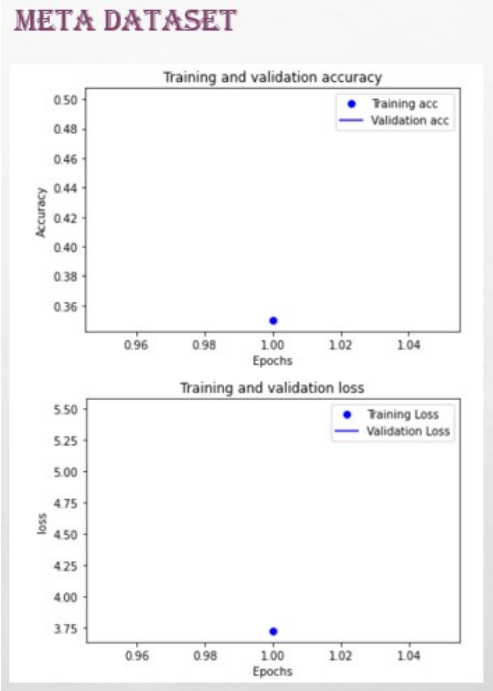

Figure 13: Results with Transfer Learning on fewer/meta dataset

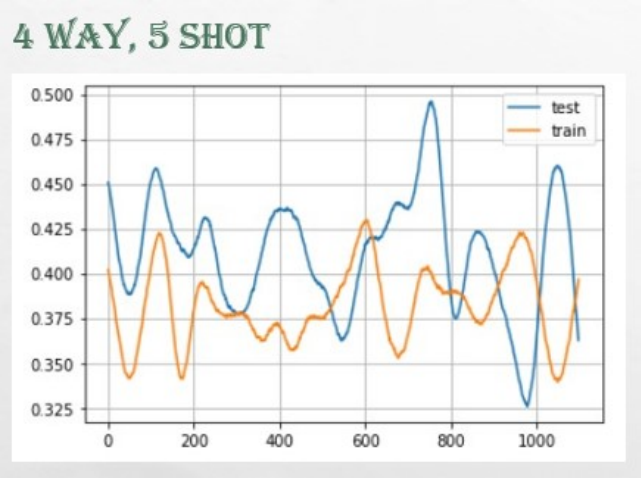

Figure 14: Meta Learning, 5 Shot - Reptile Algorithm

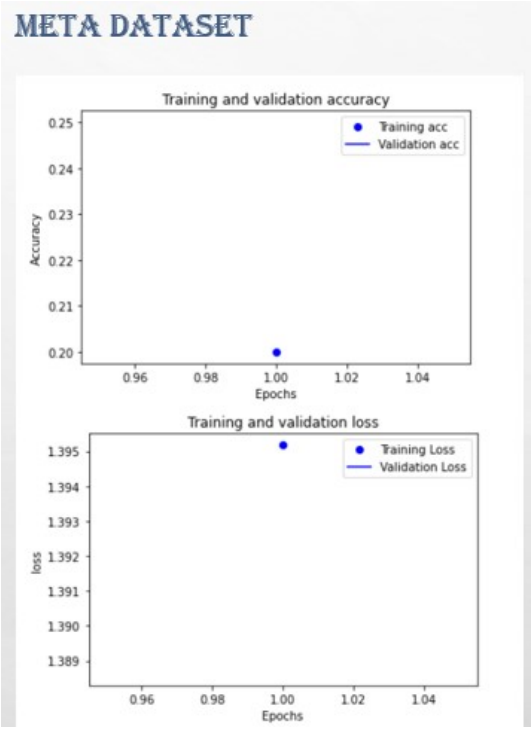

Figure 11: Results with Classic CNN on fewer/meta dataset

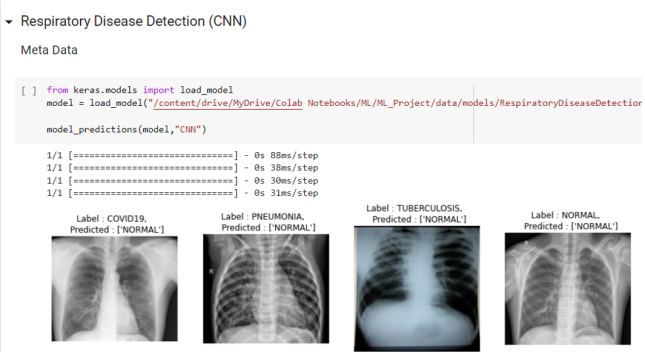

Figure 17: Visualization of CNN predictions with fewer/meta dataset

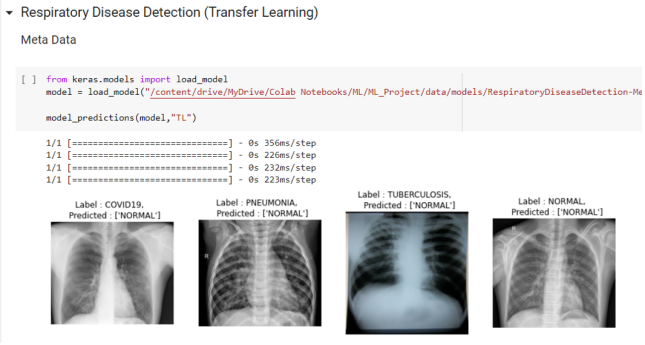

Figure 18: Visualization of Transfer Learning predictions with fewer/meta dataset

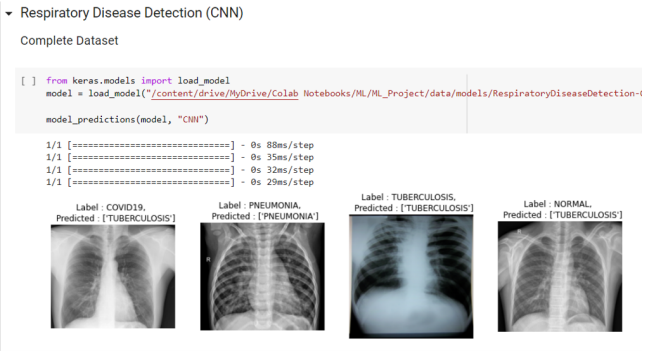

Figure 19: Visualization of CNN predictions with complete dataset

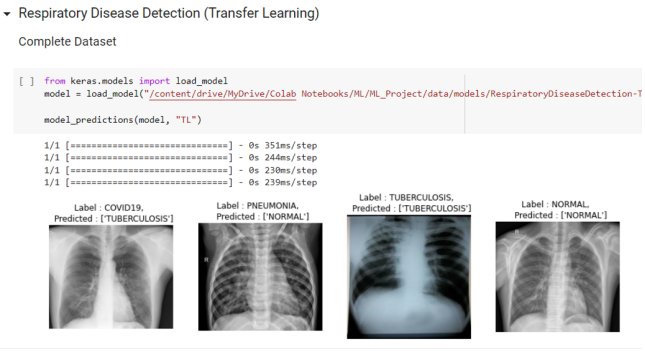

Figure 20: Visualization of Transfer Learning predictions with complete dataset

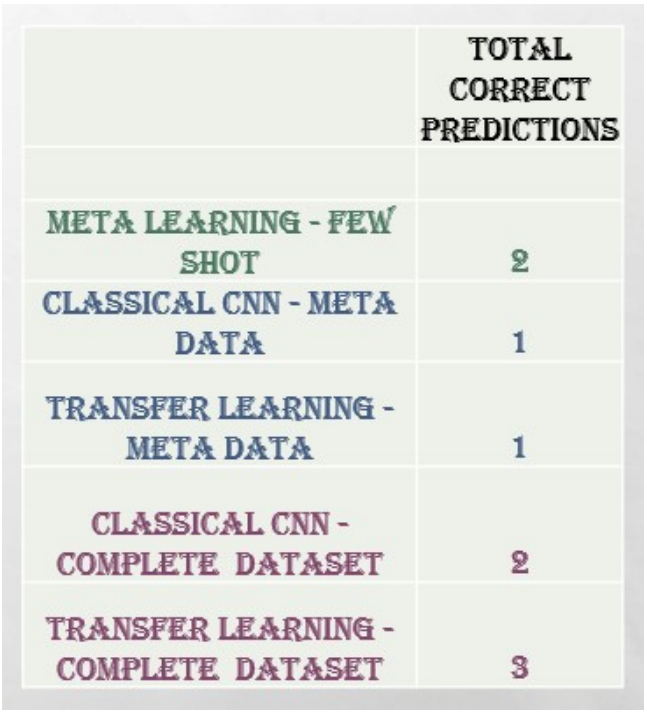

Figure 15: Models predictions evaluation results

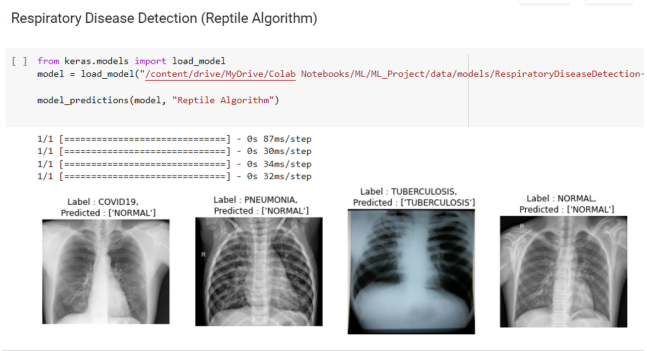

Figure 16: Visualization of Meta Learning predictions

8 GIT REPOSITORY

https://github.com/AkshayaBTauro/Fall2022_MetaLearning_FewShot_project

REFERENCES

[1] ADMoreau. [n.d.]. Few-Shot learning with Reptile. ([n. d.]).

[2] John Schulman Alex Nichol, Joshua Achiam. [n.d.]. On First-Order Meta-Learning Algorithms. ([n. d.]). https://arxiv.org/abs/1803.02999
[3] JtiptJ. [n.d.]. Chest X-Ray (Pneumonia,Covid-19,Tuberculosis). ([n. d.]). https://www.kaggle.com/datasets/jtiptj/chest-xray-pneumoniacovid19tuberculosis
[4] Kushagra Mahajan, Monika Sharma, and Lovekesh Vig. 2020. Meta-DermDiagnosis: Few-Shot Skin Disease Identification using Meta-Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 3142–3151. https://doi.org/10.1109/CVPRW50498.2020.00373