Deep Transfer Learning for Covid-19 Detection

Xin Zeng
Massachusetts Institute of Technology
Cambridge, MA, USA
xinzeng@mit.edu

Tale Lokvenec
Harvard University
Cambridge, MA, USA
talelokvenec@g.harvard.edu

Abstract

Recent studies have shown that comprehending and mitigating the low clinical sensitivity of the SARS-CoV-2 RT-PCR testing is pivotal to the containment of the global coronavirus pandemic. In this paper, we propose a complementary method for rapid coronavirus detection from CT scans, based on the deep transfer learning paradigm. The idea is to use fine-tuning methods as a baseline, and to apply mapping-based and generative methods to enhance the transferability of knowledge and to improve the model performance on the Covid-19 detection task. As for the fine-tuning method, the VGG architecture performs the best, achieving an accuracy ***75.83%****, an F-1 score of 76.68% and an AUC of 83.95%. The mapping-based method is used to predict correct labels while keeping the domain invariant. Applying our selected domain confusion metric on the ResNet50 model, the method achieves an accuracy of* ***80.00%****, an F1-score of 80.93% and an AUC of 86.29%, outperforming the baseline strongly. The generative methods are used to alleviate the training data scarcity issue, associated with the fine-tuning methods. The VGG architecture fine-tuned on the updated dataset with generative and affine augmentations had an accuracy of* ***78.31%****, an F-1 score of 80.45% and an AUC of 85.39%. The results indicate that both proposed methods outperform the baseline and thus bring closer the idea of using deep transfer learning on CT scans as a complementary method for rapid Covid-19 detection.*

1. Introduction

In 2019, the world witnessed the outbreak of the coronavirus disease (Covid-19), an infectious SARS-CoV-2 virus causing respiratory illness [13]. On March 11, 2020 the World Health Organization (WHO) declared the coronavirus outbreak a global pandemic. More than 509 million Covid-19 cases have been confirmed worldwide, to date.

Reverse-transcription polymerise chain reaction (RT-PCR), based on selective primers that amplify the SARS-CoV-2 segments of the DNA sequence, is the most commonly used laboratory technique to detect the presence of coronavirus. Despite the very high analytical specificity and sensitivity of the RT-PCR testing regimen, the clinical sensitivity of the test is as low as 61% - 71%, predominantly due to the pre-analytic phase, i.e. the sample collection, transportation and storage. Due to the sensitivity as high as 56% - 98%, computed tomography (CT) scans are frequently alluded to as a complementary testing procedure to RT-PCR. As an example, when patients who test negative using RT-PCR are suffering pneumonia or acute respiratory complications, the computed tomography scans show patterns indicating coronavirus infection [8]. Due to the low specificity of CT scans, as low as 25% [1], the medical researchers and practitioners have been reluctant towards the adoption of CT scans as a standalone technique for coronavirus detection. The scarcity of educated radiologists, along with the time-consuming nature of manual CT scan analysis, presents CT scans as an inadmissible complementary testing procedure to RT-PCR. Deep learning models have been proven to provide speedy analysis of CT scans, accurate to the level of radiologists, for numerous common diseases. However, deep learning models are data-hungry. The absence of open-source datasets limits the adequacy of deep learning models for rare disease analysis.

In this paper we attempt to circumvent the above problems by proposing methods for rapid analysis of CT-scans based on deep transfer learning, i.e. transfering the knowledge obtained by common image classification, to identify the presence of coronavirus from CT scans.

2. Literature Review

In general, there are two categories of deep transfer learning [4]: network-based approaches and adversarial-based approaches. Network-based approaches generally include combinations of various techniques like pre-training, freezing, fine-tuning, and adding fresh layers. Adversarial-based approaches aim at finding transferable representations that are applicable to both the source domain and the target domain through adversarial technology [15]. Both

techniques have been used extensively in the field of disease detection from medical imaging.

As for network-based approaches, the most common method is using a pre-trained model on a similar dataset to target data and fine-tuning it on target data. [5] fine-tuned DenseNet201 based architectures as an automated tool for detection and diagnosis of Covid-19 in chest CT. [12] conducted Covid-19 detection by applying an extreme version of inception (Xception) which can train the weights of networks on large datasets as well as fine-tune the weights of pre-trained networks on small datasets.

The second popular network-based approach is freezing CNN layers in a pre-trained model and fine-tuning only lateral fully connected layers. [2] used a pre-trained GoogLeNet to extract features from brain MRI images and conducted a 3-class classification to differentiate among glioma, meningioma and pituitary tumors.

Progress has also been made on successfully combining different network-based techniques. [3] explored a two-step progressive transfer learning. Progressive learning means some or all layers of a pre-trained model are selected and used frozen, and some fresh layers are added to the model to be trained on target data. [10] proposed an approach to jointly learn adaptive classifiers and transferable features from labeled data in the source domain and unlabeled data in the target domain. They enable classifier adaptation by plugging several layers into a deep network to learn the residuals with reference to the target classifier.

As for adversarial-based approaches, [9] used conditional generative adversarial networks (CGAN) to expand limited target data of chest X-Ray images for detecting Covid-19 using deep transfer learning models, obtaining a testing accuracy of 82.91% on the Covid-CT-Dataset. [7] applies domain adversarial training to obtain the shared features between multiple source datasets, where the diagnostic knowledge learned from sufficient supervised data of multiple rotating machines is transferred to the target equipment with domain adversarial training.

3. Proposed Methodology

First, we fine-tune well-known, pre-trained models. Considering the issues of data dissimilarity and data scarcity in the fine-tuning methods, we propose mapping-based methods and generative methods to enhance the transferability of knowledge and improve the model performance.

3.1. Fine-tuning Method

In this part, we propose to use pre-trained models and fine-tune them on the target data. We choose to use well-known models such as DenseNet, VGGNet, AlexNet, and ResNet, which have been pre-trained on the ImageNet dataset. We regard the performance of the fine-tuning methods as our baseline.

3.2. Mapping-based Method

Considering that we have another, larger data source from similar, but different domains or distributions, directly fine-tuning on this data would be problematic. To alleviate the issue of data dissimilarity between the source and target data in the fine-tuning method, we apply a mapping-based method, aimed at learning to predict the class labels while simultaneously finding a representation that makes the domains appear as similar as possible.

Our mapping-based method is motivated by the Deep Domain Confusion (DDC) model, which introduces an adaptation layer and an additional domain confusion loss, to learn a representation that is both semantically meaningful and domain invariant [16]. They propose to add an adaptation layer on a deep 7-layer CNN network, and minimize a joint loss $\mathcal{L}$ during optimization, which consists of two components, classification loss on the available labeled data $\mathcal{L}_C(X_L, y)$ and distance loss evaluated by Maximum Mean Discrepancy (MMD). The goal of the distance loss is to minimize the distance between source and target representations. And the classification loss term enables the model to discriminate between the classes in the dataset.

In equation 1, X_L denotes the labelled dataset, y denotes the ground truth labels, and $\text{MMD}(X_S, X_T)$ denotes the distance between the source data, X_S, and the target data, X_T. MMD is computed with respect to a particular representation, $\phi(\cdot)$, and an empirical approximation to this distance is computed through equation 2.

$$\mathcal{L} = \mathcal{L}_C(X_L, y) + \lambda \text{MMD}^2(X_S, X_T) \tag{1}$$

$$\text{MMD}(X_S, X_T) = ||\frac{1}{|X_S|}\sum_{x_s \in X_S}\phi(x_s) - \frac{1}{|X_T|}\sum_{x_t \in X_T}\phi(x_t)|| \tag{2}$$

Based on the characteristics of the dataset for Covid-19 detection and the insights from fine-tuning methods, our proposed novelties for DDC model include the following:

1. Considering the imbalance between the size of the source dataset and target dataset, we propose to use a joint loss $\mathcal{L}^*$, consisting of three components to balance the classification loss from both datasets. Equation 3 shows the computation process for $\mathcal{L}^*$, where $\mathcal{L}_C(X_S, y_S)$ is the classification loss on the labeled source data and $\mathcal{L}_C(X_T, y_T)$ is the classification loss on the labeled target data. This way, we treated the classification loss from target data and the loss from the labeled source data equally, instead of treating the classification loss from the target data as only a small component of the labelled data loss in DDC model.

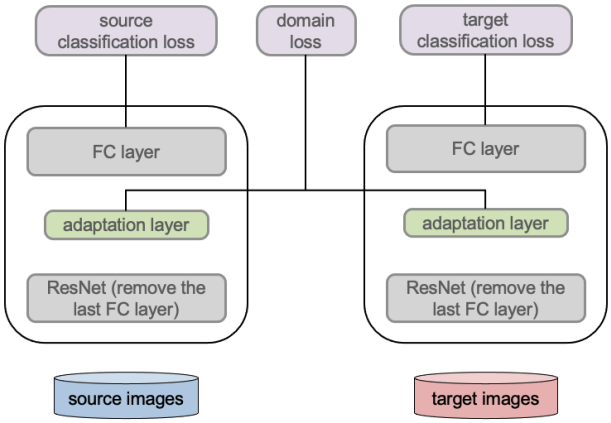

Figure 1. Our Architecture for Mapping-based Method

$$\mathcal{L}^* = \mathcal{L}_C(X_S, y_S) + \mathcal{L}_C(X_T, y_T) + \lambda \text{MMD}^2(X_S, X_T) \quad (3)$$

2. We propose to add an adaptation layer on the structure of ResNet instead of the 7-layer CNN model in DDC, which enables us to make good use of the well-developed ResNet structure and the performance gained from pretrained model. Figure 2 shows the architecture of our proposed mapping-based method. The adaptation layer is added before the last fully-connected layer of the ResNet model, and the domain loss is computed based on the output of the adaptation layer.

3. As for the representation $\phi(\cdot)$ to calculate MMD, we propose to use either radial basis function (RBF) kernel or linear kernel, which are popular kernel functions used in various kernelized learning algorithms.

4. Motivated by the CORrelation ALignment (CORAL) method [14], which minimizes domain shift by aligning the second-order statistics of source and target distributions. We further analyze the performance of substituting MMD with CORAL loss. Equation 4 shows the computation process for $\mathcal{L}^{**}$, where $\mathcal{L}_{\text{CORAL}}$ is the CORAL loss. Equation 5 shows the calculation process of $\mathcal{L}_{\text{CORAL}}$, where C_S and C_T are the covariance matrices of the source and target data respectively, and $||\cdot||_F^2$ denotes the squared matrix Frobenius norm, and d is the dimension of deep layer activations of input.

$$\mathcal{L}^{**} = \mathcal{L}_C(X_S, y_S) + \mathcal{L}_C(X_T, y_T) + \lambda \mathcal{L}_{\text{CORAL}}(X_S, X_T) \quad (4)$$

$$\mathcal{L}_{\text{CORAL}}(X_S, X_T) = \frac{1}{4d^2} ||C_S - C_T||_F^2 \quad (5)$$

5. We further conduct an ablation study to see how the deficiency of labels from target dataset influences the performance, which shows the generalizability of our model in target data without labels. In this case, we are not able to use $\mathcal{L}_C(X_T, y_T)$ in our loss function. $\mathcal{L}_-^*$ and $\mathcal{L}_-^{**}$ denote the loss function without $\mathcal{L}_C(X_T, y_T)$ for using MMD and CORAL respectively. And for the representation of MMD here, we use the RBF kernel. The computation for $\mathcal{L}_-^*$ and $\mathcal{L}_-^{**}$ are shown in equation 6 and 7 respectively.

$$\mathcal{L}_-^* = \mathcal{L}_C(X_S, y_S) + \lambda \text{MMD}^2(X_S, X_T) \quad (6)$$

$$\mathcal{L}_-^{**} = \mathcal{L}_C(X_S, y_S) + \lambda \mathcal{L}_{\text{CORAL}}(X_S, X_T) \quad (7)$$

3.3. Generative Method

To overcome the issue of data scarcity in the fine-tuning method, we apply a generative method to create synthetic Covid-19 images and increase the size of the original dataset. We use the generative method as an add-on to our classical data augmentation procedure which includes a combination of affine image transformations.

Motivated by [9], the first approach we used to generate additional Covid-19 images was a Conditional Generative Adversarial Network (CGAN). We re-implemented the CGAN architecture described in the paper above. The generator consisted of 3 hidden blocks, each containing a transposed convolutional layer, a ReLU activation layer and a batch normalization layer, and an output block consisting of a transposed convolutional layer and a Tanh activation. The discriminator consisted of 2 hidden blocks, each containing a convolutional layer, a leaky ReLU activation layer and a batch normalization layer, and an output block consisting of a convolutional layer. The optimization function minimized during training was the Binary Cross Entropy (BCE) with Logits, i.e. the more numerically stable version of stacking a Sigmoid output layer in the discriminator, and using the BCE loss.

$$\mathcal{L} = -\sum_{i=1}^{n} [y_i \cdot \log(\sigma(x_i)) + (1 - y_i) \cdot \log(1 - \sigma(x_i))] \quad (8)$$

The second generative method we implemented was a Variational Autoencoder (VAE). The VAE architecture consists of an encoder and a decoder. The encoder was comprised of 6 convolutional layers, followed by 2 linear layers, one computing the mean vector and one computing the log-variance vector of the latent space distribution. The decoder was comprised of 1 linear layer, followed by 6 transposed convolutional layers.

The loss function minimized during training consists of two components, a reconstruction term evaluated through

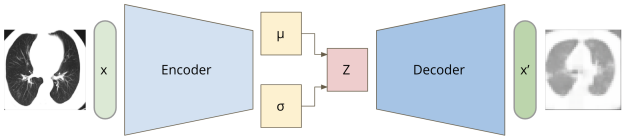

Figure 2. Our Architecture for VAE, Adversarial-based Method

the Binary Cross Entropy (BCE) loss and a regularization term aimed at minimizing the Kullback-Leibler (KL) divergence between the inferred latent distribution and the fixed prior distribution (fixed prior distribution $P \sim \mathcal{N}(0,1)$). The goal of the regularization term is two-fold: to ensure continuity - points that are close in the sampling space should contain similar content after decoding and to ensure completeness - samples from the latent space should contain meaningful content after decoding. Equation 9, shows the joint loss function, where β_{VAR} is a hyperparameter that balances between the continuity and completeness, and the reconstruction accuracy.

$$\mathcal{L} = \mathcal{L}_{BCE} + \beta_{VAL} \cdot \mathcal{L}_{KL} \tag{9}$$

$$\mathcal{L}_{BCE} = -\sum_{i=1}^{n}[y_i \cdot \log(\sigma(x_)) + (1-y_i) \cdot \log(1-\sigma(x_i))] \tag{10}$$

$$\mathcal{L}_{KL} = -\frac{1}{2}\sum_{i=1}^{n}[\sigma_i + \mu_i^2 - 1 - \log(\sigma_i)] \tag{11}$$

4. Evaluation

We use the Covid-CT-Dataset [17] to evaluate our model performance, which is one of the most widely used CT datasets for Covid-19 detection in recent literature. The Covid-CT-Dataset has 349 CT images containing clinical findings of Covid-19 from 216 patients and 463 non-COVID-19 CTs. We follow the default setup given by the data producer to split the data into training, validation and test dataset. We apply three metrics to evaluate the model performance, accuracy (ACC), F1-score and AUC.

4.1. Fine-tuning Method

For fine-tuning method, we used different pre-trained models including ResNet50, ResNet18, DenseNet, AlexNet, VGG and SqueezeNet. We trained each of them for 20 epochs with learning rate of 0.002. We choose an SGD optimizer with a momentum setting to 0.9. We also implemented a standard data augmentation process including random cropping, horizontal flipping, and brightness and contrast adjustment. To enhance the reproducibility of our work, we ran each experiment 5 times and we report the mean scores with the associated standard deviations in table 1.

Model	ACC (%)	F1-score (%)	AUC (%)
ResNet50	75.38 ± 3.16	**76.69** ± 2.51	83.36 ± 3.38
ResNet18	75.57 ± 2.93	76.22 ± 3.19	81.34 ± 3.13
VGG	**75.83** ± 3.93	76.68 ± 2.91	**83.95** ± 1.58
DenseNet	74.38 ± 2.84	75.74 ± 2.60	82.74 ± 2.92
AlexNet	65.62 ± 4.52	69.86 ± 3.70	69.86 ± 3.70
SqueezeNet	56.85 ± 4.68	48.42 ± 26.16	63.02 ± 10.01

Table 1. Performance of Fine-tuning Method

Among all models we have tested, ResNet50, ResNet18, VGG and DenseNet have comparable performance, with the difference in the three metrics only $\sim 2\%$. VGG performs the best in terms of accuracy and AUC, achieving an average accuracy of 75.83% and an average AUC of 83.95% while ResNet50 performs the best in terms of F1-score, achieving an average F1-score of 76.69%.

4.2. Mapping-based Method

To conduct our mapping-based method, we first find and process a source dataset from another domain. We use a large, lung CT scans dataset for COVID-19 as our source dataset, which contains 7,593 COVID-19 images from 466 patients, 6,893 normal images from 604 patients [11]. The source dataset is collected from 7 different sources, with 1 source containing our Covid-CT-Dataset. Thus, we manually delete the 349 overlapping CT images from the source dataset to ensure the source data doesn't contain any information about our target dataset, and treat the remaining as our source dataset.

Based on the methods proposed in section 3.2, there are 5 different loss functions to be evaluated. Both $\mathcal{L}^*_{\text{rbf}}$ and $\mathcal{L}^*_{\text{linear}}$ follow the computation process in equation 3, but with the representation $\phi(\cdot)$ following RBF kernel and linear kernel respectively. And $\mathcal{L}^*_{\_}$, $\mathcal{L}^{**}$ and $\mathcal{L}^{**}_{\_}$ follow exactly the same definition and calculation process as in section 3.2. We also tested two different ResNet models, ResNet18 and ResNet50. We trained each of them for 40 epochs with a learning rate of 0.0001. As for the hyperparameter λ, we choose 0.25 for MMD and 2.5 for CORAL due to the difference in scale. We also implement the standard data augmentation process including random cropping, horizontal flip and adjustment on brightness and contrast. Similarly, we run each experiment 5 times and report the mean scores and the associated standard deviations in table 2 for better reproducibility.

Among all models with different loss functions we have tested, the ResNet50 model with $\mathcal{L}^*_{\text{rbf}}$ performed the best in terms of all three metrics, achieving an average accuracy of 80.00%, an average F1-score of 80.93%, and an aver-

Model	loss	ACC (%)	F1-score (%)	AUC (%)
ResNet50	$\mathcal{L}^*_{\text{rbf}}$	**80.00** $\pm$ 1.23	**80.93** $\pm$ 1.59	**86.29** $\pm$ 1.41
ResNet50	$\mathcal{L}^*_{\text{linear}}$	50.74 $\pm$ 1.94	55.74 $\pm$ 4.43	46.79 $\pm$ 4.01
ResNet50	$\mathcal{L}^*_{-}$	67.73 $\pm$ 0.85	71.49 $\pm$ 1.07	76.64 $\pm$1.08
ResNet50	$\mathcal{L}^{**}$	76.75 $\pm$ 3.76	77.63 $\pm$ 3.81	83.84 $\pm$ 2.30
ResNet50	$\mathcal{L}^{**}_{-}$	70.81 $\pm$ 2.29	72.22 $\pm$ 2.21	80.50 $\pm$ 1.90
ResNet18	$\mathcal{L}^*_{\text{rbf}}$	78.62 $\pm$ 1.70	80.33 $\pm$ 1.16	85.71 $\pm$ 1.42
ResNet18	$\mathcal{L}^*_{\text{linear}}$	52.22 $\pm$ 2.46	49.26 $\pm$ 4.72	48.74 $\pm$ 4.22
ResNet18	$\mathcal{L}^*_{-}$	65.19 $\pm$ 4.13	68.60 $\pm$ 1.65	77.41 $\pm$ 2.14
ResNet18	$\mathcal{L}^{**}$	75.96 $\pm$ 4.52	76.37 $\pm$ 5.26	82.72 $\pm$ 3.52
ResNet18	$\mathcal{L}^{**}_{-}$	66.87 $\pm$ 0.81	67.00 $\pm$ 0.73	72.88 $\pm$ 2.52

Table 2. Performance of Mapping-Based Method

age AUC of 86.29%. Both ResNet18 and ResNet50 models with $\mathcal{L}^*_{\text{rbf}}$ also outperform all of our fine-tuning methods in terms of all three metrics, showing that our adaptation layer learned through the designed joint loss $\mathcal{L}^*_{\text{rbf}}$ could successfully transfer to our target domain, and help further boost the performance of the fine-tuning method.

As for the comparison between different domain losses, we could see that the joint loss constructed with MMD performs better than that with CORAL, showing that MMD is a better metric to capture the distance of distributions between source and target domains and to implement the transfer learning procedure for the Covid-CT-Dataset. This is also validated during our training process, where the CORAL loss doesn't vary a lot from epoch to epoch. Furthermore, as for the choice of kernel in MMD, the linear kernel performs far worse than the RBF kernel. Changing from the RBF kernel to the linear kernel results in a loss of accuracy of more than 20%. This shows that MMD based on a linear kernel is not capable to learn a representation that is domain invariant for the task of Covid-CT detection.

In general, based on the results, we could conclude that the ResNet18 model performs worse than the ResNet50 model in our mapping-based method, showing the representations learned from ResNet50 are more effective for classifying the target dataset.

And our ablation study on the deficiency of labels shows that we would lose around 10% of accuracy without access to the labels of the training dataset in the target domain.

4.3. Generative Method

The generative method was implemented using the Covid-CT-Dataset discussed in section 4.

The Conditional Generative Adversarial Network (CGAN) model was trained for 1,200 epochs, with a learning rate of 0.0002 and a batch size of 128. We chose an Adam optimizer with momentum setting to (0.5, 0.999). We initialized the model weights to the normal distribution with mean of 0 and standard deviation of 0.02.

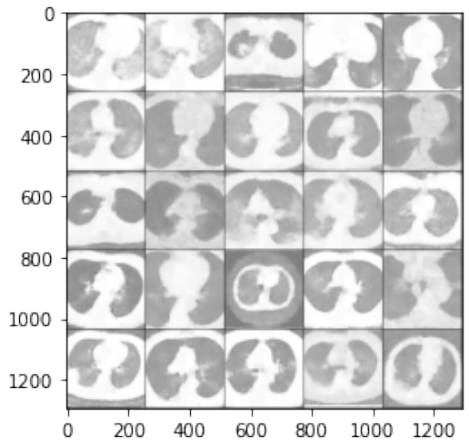

Figure 3. Synthetic Covid-19 Images using the VAE model

Despite the hours spent fine-tuning the CGAN model, the synthetic images generated by the CGAN did not meet the quality criteria to be included in the augmented dataset. Conducting performance analysis, we came to the conclusion that the CGAN model suffered from one most common GAN failure modes, the mode collapse anomaly [6], where in each iteration the generator over-optimizes for a given set of discriminator parameters. The discriminator learns to always recognize and always reject the particular sample output, never learning its way out of the local minimum. As a result, the generator learns to produce a small set of outputs that temporarily trick the discriminator, but do not have a meaningful interpretation.

The Variational Autoencoder (VAE) model was trained for 250 epochs, with a learning rate of 1e-3 and a batch size of 64. We chose an Adam optimizer, with weight decay of 1e-5. We chose the latent dimension of the mean and the standard deviation vectors to be (20,). The synthetic images generated by the VAE model are displayed in addition.

While qualitatively, we observe that the synthetic images meet the visual criteria of non-expert observers, our biggest concern was whether the VAE model learned to generate meaningfully distinct representations for the two image classes (Covid and NonCovid). In order to validate/reject our concern, we decided to use the fine-tuning methods on the augmented dataset and compare the results to the results obtained using the original dataset.

4.3.1 Fine-tuning Method on Augmented Dataset

In this section, we applied the set of pre-trained models and we used the set of hyperparameters described in section 4.1. However, in addition to the standard data augmentation process, including affine transformations of the input images, we used the VAE model to generate 576 Covid training images and 576 NonCovid training images. Similar to the ar-

Model	ACC (%)	F1-score (%)	AUC (%)
ResNet50	76.76 ± 4.93	**79.48** ± 4.48	85.59 ± 1.85
ResNet18	75.35 ± 0.70	76.82 ± 0.51	84.32 ± 1.89
VGG	**77.18** ± 1.38	78.71 ± 1.59	**86.86** ± 2.16
DenseNet	74.37 ± 6.00	75.83 ± 6.72	82.36 ± 4.88
AlexNet	67.61 ± 5.56	74.74 ± 3.08	68.26 ± 9.63
SqueezeNet	40.85 ± 0.00	0.00 ± 0.00	50.00 ± 0.00

Table 3. Performance of Fine-tuning Method of Augmented Dataset

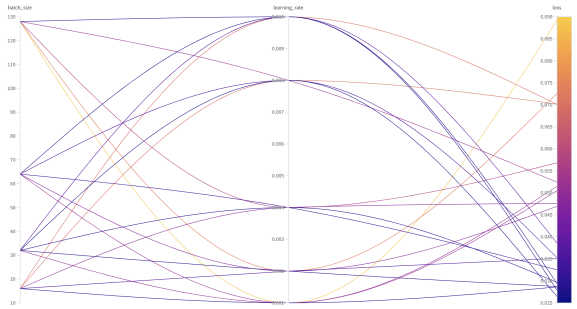

Figure 4. Hyperparameters Tuning on VGG

gument made in section 4.1., in order to enhance the reproducibility of our results we ran each experiment twice and we reported the mean and the standard deviation of each experiment in table 3.

The results obtained mimic the results obtained in section 4.1., with ResNet50, ResNet18, VGG and DenseNet having comparable performance. Again, VGG performs the best in terms of accuracy and AUC, achieving an average accuracy of 77.18% and an average AUC of 86.86%, while ResNet50 performs the best in terms of F1-score, achieving average F1-score of 79.48%. The general trend we observe when comparing the results of using the augmented dataset to the original dataset is that the metrics for the best performing models improve by $\sim$ 2% under the augmented dataset. Given the large variance of results between experiments, it is difficult to draw general conclusions, but it is safe to say that the synthetic images generated by the VAE also meet the quantitative criteria and contribute to a performance boost of the fine-tuning methods.

4.3.2 HP Tuning of VGG using Augmented Dataset

In order to maximize the performance of the fine-tuning method on the augmented dataset, we performed a grid hyperaparameters sweep, on the batch size and the learning rate using the validation loss as the accuracy metric. The results of the sweep are shown in figure 4. Based on the results obtained, we re-trained the VGG network 5 times with batch size of 32 and learning rate of 0.008.

Model	ACC (%)	F1-score (%)	AUC (%)
VGG	**78.31** ± 1.91	**80.45** ± 1.93	**85.39** ± 2.78

Table 4. Performance of HP-tuned VGG on Augmented Dataset

The HP-tuned VGG, has higher accuracy and higher F1-score than the original VGG, however it has a lower AUC score. However, given the class balance in our dataset, we claim that accuracy is the more relevant metric, as we are interested in the performance of the model for the particular threshold value. Thus, we can conclude that the HP-tuning process yielded an improved version of the original VGG trained on the augmented dataset.

5. Conclusion

In this paper, we present three different deep transfer learning methods for Covid-19 detection, which are fine-tuning, mapping-based and generative methods. The fine-tuning method serves as a baseline for comparison with our optimizations. Fine-tuning method with VGG architecture performs the best, achieving an average accuracy of 75.83%, an average F1-score of 76.68%, and an average AUC of 83.95%.

In the mapping-based method, we present a joint loss capable of learning domain invariant representations and balancing the classification loss between the source and target data. We then conduct experiments on different choices of representation in MMD and the metrics of domain loss function. We also conduct an ablation study to analyze the performance loss from the deficiency of target labels. Our best model, which uses MMD with RBF kernel and builds on the ResNet50 model, achieves an average accuracy of 80.00%, an average F1-score of 80.93%, and an average AUC of 86.29%, outperforming the baseline strongly.

In the generative method, we train a VAE architecture using a joint loss that balances between the continuity and completeness of the latent space, and the reconstruction accuracy of the model. We use the trained VAE to generate synthetic training data, alleviating therefore the problem of training data scarcity for the fine-tuning method. The best performing fine-tuned architecture, VGG, on the augmented dataset, after hyperparameters optimization achieves an average accuracy of 78.31%, an average F1-score of 80.45%, and an average AUC of 85.39%, outperforming the baseline fine-tuned model.

In summary, our mapping-based method benefits from the larger size of source data and the joint loss to maintain the domain invariance, while our generative method benefits from synthetic data generation. We credit the slightly better performance of our mapping-based method to the amount of source data and the wise choice of a loss metric.

6. Individual Contribution

In general, Xin and Tale contributed equally to the project. In detail, Xin conducted the literature review and proposed the modeling ideas of three methods. In the implementation, Xin took the responsibility of designing, coding, and analyzing the performance fine-tuning and mapping-based methods, and helped guide the modeling of generative method. In the writing, Xin contributed to the majority of the literature review, methods and evaluation of fine-tuning and mapping-based methods, and the conclusion.

References

[1] Harrison X Bai, Ben Hsieh, Zeng Xiong, Kasey Halsey, Ji Whae Choi, Thi My Linh Tran, Ian Pan, Lin-Bo Shi, Dong-Cui Wang, Ji Mei, et al. Performance of radiologists in differentiating covid-19 from non-covid-19 viral pneumonia at chest ct. *Radiology*, 296(2):E46–E54, 2020. 1

[2] S. Deepak and P.M. Ameer. Brain tumor classification using deep cnn features via transfer learning. *Computers in Biology and Medicine*, 111:103345, 2019. 2

[3] Bonnington CP Zhou J Gu Y, Ge Z. Progressive transfer learning and adversarial domain adaptation for cross-domain skin disease classification. *IEEE J Biomed Health Inform*, 24(5):1379-1393, 2020. 2

[4] Mohammadreza Iman, Khaled Rasheed, and Hamid R. Arabnia. A review of deep transfer learning and recent advancements. *CoRR*, abs/2201.09679, 2022. 1

[5] Singh D Kumar V Kaur M. Jaiswal A, Gianchandani N. Classification of the covid-19 infected patients using densenet201 based deep transfer learning. *J Biomol Struct Dyn*, 39(15):5682–5689, 2021. 2

[6] Qi Li, Long Mai, Michael A Alcorn, and Anh Nguyen. A cost-effective method for improving and re-purposing large, pre-trained gans by fine-tuning their class-embeddings. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 5

[7] Xiang Li, Wei Zhang, Qian Ding, and Xu Li. Diagnosing rotating machines with weakly supervised data using deep transfer learning. *IEEE Transactions on Industrial Informatics*, 16(3):1688–1697, 2020. 2

[8] Yafang Li, Lin Yao, Jiawei Li, Lei Chen, Yiyan Song, Zhifang Cai, and Chunhua Yang. Stability issues of rt-pcr testing of sars-cov-2 for hospitalized patients clinically diagnosed with covid-19. *Journal of medical virology*, 92(7):903–908, 2020. 1

[9] Manogaran G. Khalifa N.E.M. Loey, M. A deep transfer learning model with classical data augmentation and cgan to detect covid-19 from chest ct radiography digital images. *Neural Comput & Applic*, 2020. 2, 3

[10] Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. *CoRR*, abs/1602.04433, 2016. 2

[11] Law A.C Shen B. Zhou Y. Yazdi N. Maftouni, M. and Z.J. Kong. A robust ensemble-deep learning model for covid-19 diagnosis based on an integrated ct scan images database. *Proceedings of the 2021 Industrial and Systems Engineering Conference*, May 22-25, 2021. 4

[12] N. Narayan Das, N. Kumar, M. Kaur, V. Kumar, and D. Singh. Automated deep transfer learning-based approach for detection of covid-19 infection in chest x-rays. *IRBM*, 43(2):114–119, 2022. 2

[13] Tanu Singhal. A review of coronavirus disease-2019 (covid-19). *The indian journal of pediatrics*, 87(4):281–286, 2020. 1

[14] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. *CoRR*, abs/1612.01939, 2016. 3

[15] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. *CoRR*, abs/1808.01974, 2018. 1

[16] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014. 2

[17] Jinyu Zhao, Yichen Zhang, Xuehai He, and Pengtao Xie. Covid-ct-dataset: a ct scan dataset about covid-19. *arXiv preprint arXiv:2003.13865*, 2020. 4