Classification of COVID-19 from CT scans with Multi-Source Transfer Learning

Alejandro R. Martinez
Dartmouth College
alejandro.r.martinez.20@dartmouth.edu

***Abstract*—Since December of 2019, the novel Coronavirus COVID-19 has spread around the world infecting millions of people. One of the main reasons for its high rate of infection is due to the unreliability and lack of RT-PCR testing. As an alternative, recent research has investigated the use of Convolutional Neural Networks for the classification of COVID-19 from CT scans with relative success. Because there is an inherent lack of available COVID-19 CT data, researchers are forced to resort to methods of Transfer Learning. Transfer Learning has proven to improve model performance on tasks with relatively small amounts of data, as long as the Source feature space is not too different from the Target feature space. Unfortunately, this difference is often encountered in the classification of medical images as publicly available Source datasets usually lack the visual features found in medical images. In this study, we propose the use of Multi-Source Transfer Learning (MSTL) to improve upon traditional Transfer Learning for the classification of COVID-19 from CT scans. With our multi-source fine-tuning approach, our models outperformed baseline models fine-tuned with ImageNet. We additionally, propose an unsupervised label creation process, which enhances the performance of our Deep Residual Networks. Our best performing model was able to achieve an accuracy of 0.893 and a Recall score of 0.897, outperforming its baseline Recall score by 9.3%.**

Keywords: *COVID-19, Transfer Learning, Convolutional Neural Networks, CT*

I. Introduction & Related Work

On March 11th 2020, the World Health Organization (WHO) proclaimed the novel coronavirus COVID-19 a global pandemic. Originating in the Hubei Province of China in late 2019, COVID-19 has spread across 185 countries, infecting nearly 6 million people and causing over 350,000 deaths [1], [2]. One of the main reasons for its unprecedented growth is due to the unreliability and lack of testing [3].

The most widely employed test kits are reverse transcription polymerase chain reaction (RT-PCR) tests, which check for the detection of nucleic acid from SARS-CoV-2 in respiratory specimens [4]. While RT-PCR tests are commonly used, they are reported to yield poor sensitivity in early stages of infection and require a lengthy processing time [3], [5]. In addition to these issues, RT-PCR tests face severely limited supply, causing many symptomatic people to be left untested[6].

In light of these constraints, Computed Tomography (CT) imaging has been explored as a possible alternative diagnostic tool for COVID-19 [3], [5], [7]. Prominent features of the virus, such as bilateral ground-glass opacities, have been identified in the chest CT scans of patients with COVID-19. These visual features have a potential to act as regions of interest in the detection of the virus [3], [5]. CT imaging also produces much faster results in comparison to RT-PCR tests and is widely available with roughly 6,000-7,000 scanners present in the United States [8].

The use of CT imaging as a diagnostic tool would require Deep Learning and Computer Vision technologies. These computational tools have been successfully applied to CT and other medical imaging classification tasks with low rates of error [9], [10], [11]. The most commonly applied algorithm for image classification is the Convolutional Neural Network (CNN). Researchers have used CNNs for classification of lung diseases in chest CT with radiologist level accuracy [12]. Recently, CNNs have been applied to COVID-19 chest CT classification and have shown promising results.He et al (2020) and Xu et al (2020) have both explored the use of CNNs for distinguishing COVID-19 from other types of pneumonia or normal chest CTs and managed to achieve overall accuracies of 86.0% and 86.7%, respectively [13], [14]. While recent research has seemed hopeful, there still remain limitations.

With all Deep Learning problems, the amount of collected data largely determines the success of the system. As COVID-19 is a novel virus, there is an inherent lack of available datasets to construct a robust Deep Learning classifier capable of producing reliable results. To compensate for this issue, both He et al (2020) and Xu et al (2020) exploit the use of transfer learning: a method by which a network is pretrained on a large Source task and then retrained on a smaller Target task. Transfer learning provides a network with a deep understanding of generic features from a Source dataset, so it does not require much data to learn the idiosyncrasies of the Target dataset [15]. The problem with applying this method to medical imaging, however, is that the Source dataset the network is trained on (i.e. ImageNet) usually contains very dissimilar feature spaces to those in the Target dataset [10]. This leaves the network with sub par performance, compared to what it could achieve if pretrained on a an additional dataset of medical images.

In this study we aim to provide a highly sensitive classification model for the detection of COVID-19 by exploiting a Multi-Source Deep Transfer Learning process to distinguish COVID-19 from normal CT scans.

We start with the collection of multiple Deep CNNs pretrained on ImageNet, provided by TensorFlow, Google's open source Machine Learning Library [16]. We then collect two datasets: the first is comprised of 22,238 lung CT slices from the SPIE-AAPM Lung CT Challenge provided by the Cancer Imaging Archive; the second is comprised of 349 COVID-19 and 397 normal CT scans provided by the UCSD Department of Engineering [17], [18]. The first dataset will be used to teach our pretrained ImageNet models to extract relevant features from chest CT scans, and the second dataset will be used to further fine-tune the models on distinguishing COVID-19 from normal chest CT scans.

II. Methods

In this section we describe our Multi-Source Transfer Learning approach for the classification of COVID-19 from normal chest CT scans. Our methodology begins with the collection of multiple chest CT datasets, followed by data preprocessing, model selection, and ultimately MSTL.

A. Data description

Our study utilizes three separate datasets as part of our multi-source fine-tuning paradigm: a source, a transition, and a target.

Source dataset: Being that our selected models are pretrained on an ImageNet subset known as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset, this will serve as our Source dataset. The ILSVRC dataset comprises of 1.2 million images spanning 1,000 unique classes. Each class in the dataset corresponds to a distinct synonym set, or a synset, defined by WordNet, a large lexical database that retains a semantic hierarchy between concepts through the construction of synsets [19]. This structure ensures that ILSVRC classes hold unique feature representations, making the dataset conducive to generalization. All images are labeled by human annotators via Amazon's Mechanical Turk, a crowd sourcing marketplace [19], [20].

Transition dataset: Our Transition dataset comes from the 2015 SPIE-AAPM-NCI Lung Nodule Classification Challenge, made available through The Cancer Imaging Archive (TCIA) and sponsored by the SPIE, NCI/NIH, AAPM and The University of Chicago. The dataset contains 22,489 CT scan slices from 70 patients (28 males, 42 females: median age: 61 years), containing 42 benign and 41 malignant lung nodules in total. All scans were acquired on Philips Brilliance 16, 16P, and 64 scanners, and stored as 3-dimensional DICOM files with resolution of 512x512 per slice. All protected health information was removed from DICOM headers [17].

Target dataset: Our Target dataset is collected from an open access COVID-19 CT image repository provided by the UCSD Department of Engineering [18]. The dataset contains 349 COVID-19 and 397 nonCOVID-19 or normal CT scans. The normal CT scans were collected from MedPix, an open-access online database of medical images. The COVID-19 scans were manually selected from 760 preprints on COVID-19 from medRxiv and bioRxiv, published from January 19th to March 25th. Of the scans collected 137 contain gender information and 169 contain age information. From the available metadata, the mean age of patients is calculated to be roughly 45 years old and the gender distribution is 86 males to 51 females. Most cases were reported to be from East Asia, with an overwhelming majority from Wuhan, China. It should be noted that the creators of the dataset claim the quality of CT images is well-preserved.

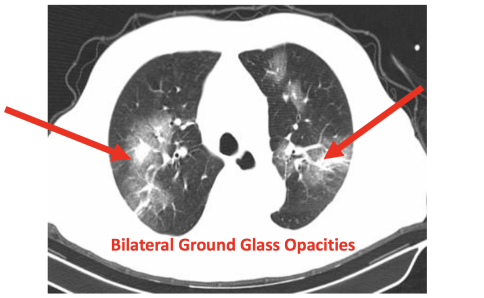

Fig. 1: Regions of interest exhibiting key identifiers of COVID-19

B. Data preprocessing

The purpose of the Transition step is to improve the learning transfer from the Source task to Target task. It does this by teaching our network to extract low to mid-level features that are more prevalent in our Target feature space than in our Source feature space. Therefore, it is essential that the Transition dataset is thoroughly filtered of images that may be too dissimilar from images in our Target dataset, so the model only learns relevant features during the Transition step.

Scans from the Target dataset contain only 2-dimensional slices, focusing on areas of the lungs displaying distinguishable COVID-19 symptoms. This region of interest captures a clear peripheral view of lung lobes that would exhibit key identifiers of COVID-19 such as bilateral ground glass opacities, as depicted in Fig.1. Because our Transition dataset comprises of 3-dimensional CT scans, containing many axial slices per scan, we excluded any slices that did not display the same regions of interest as shown in the Target slices. We then exported the resulting 10,176 Transition slices as JPG files to process them as image arrays in our pretrained networks. We conducted all preprocessing on Transition data with Horos.

To diminish the variability in image processing, we ensured that each input image was reshaped to a standard dimensions of (224,224,3).

C. Model Selection

We selected four pretrained models from Tensorflow's Keras API: ResNet50V2, ResNet101V2, DenseNet121, and DenseNet169 [21], [22]. See Figure 2 for a comparison of architectures.

ResNet: The ResNet is a Deep Residual Network, a type of CNN often employed in the field of computer vision since

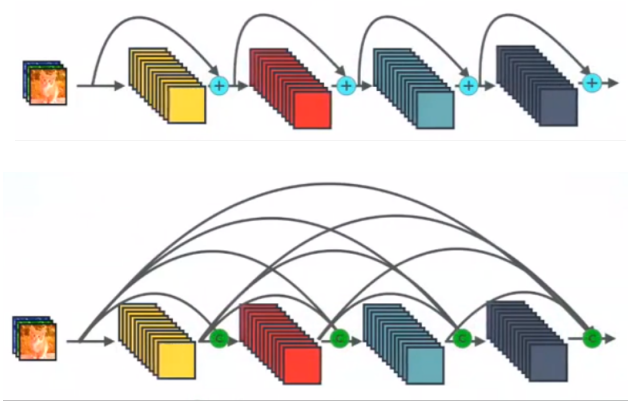

Fig. 2: ResNet (top) & DenseNet (bottom) Architectures

gaining recognition from wining the 2015 ILSVRC[21]. Deep Residual Networks act similarly to deep CNNs except they implement a residual connection between any given layer and its following layer's output. This entails that any given layer will receive feature maps as input from its two preceding layers. The residual connections improve upon regular neural networks in two ways: they mitigate the vanishing gradient problem by allowing the use of an alternative paths for gradient flow, and they allow the model to learn referenced functions which ensures deeper layers will perform either better or at least as good as shallower layers.

DenseNet: The DenseNet is very similar to the ResNet, however, instead of retaining a single residual connection between any given layer and its following layer's output, any given layer in a DenseNet retains a residual connection from each of its preceding layers. This entails that any n^{th} layer will take feature maps as input from n_{-1} preceding layers. The Dense connectivity of this network is then compacted by grouping dense layers into *dense blocks* and introducing transition layers, which apply convolutions and pooling operations to the dense blocks' output feature maps, which reduces the depth of these feature maps by a compression factor of θ [22]. This compression process allows DenseNets to hold less parameters than ResNets, as depicted in Table 1.

Model	# of Parameters
ResNet50V2	25,613,800
ResNet101V2	44,675,560
DenseNet121	8,062,504
DenseNet169	14,307,880

TABLE I: Parameters of each model

D. Multi-Source Transfer Learning

As illustrated in Figure 3, our Multi-Source Transfer Learning process involves a three step process:

1) **Source step:** a randomly initialized model learns a Source task T_s on a Source domain D_s
2) **Transition step:** the model is then fine-tuned on a Transition task T_t on a Transition domain D_t
3) **Target step:** the model is ultimately fine-tuned on a Target task T_{targ} on a Target domain D_{targ}

Fig. 3: Multi-Source Transfer Learning paradigm

In theory, the Transition step can be a sequence of steps, however, in this study we limit the total number of Transition steps to 1 for computational simplicity. The Multi-Source process allows for increasingly positive transfer of knowledge at each step assuming that at each step, i, the domain D_i is a better approximation of Target domain D_{targ} than its preceding step, i-1.

Therefore:

$$|D_{i\text{-}1} - D_{targ}| > |D_i - D_{targ}| \quad (1)$$

,where the difference in similarity between a and b is represented as:

$$|a - b| \quad (2)$$

Source step: As indicated in section 2.3, our loaded models are pretrained on the ILSVRC dataset. This indicates that our models have already learned our Source task T_s on our Source domain D_s.

Transition step: For our Transition step we decided to explore the use of two alternate Transition tasks via different labeling strategies: Soft Labeling and Hard Labeling.

- *Hard Labeling:* hard labels can be though of as ground truth labels. In other words, for the Hard Labeling strategy we utilize the original labels that were provided to us in the Transition dataset: malignant (1) or benign (0). This binary labeling strategy makes our Hard Labeling Transition task $T_{t\text{-}hard}$ a binary classification task.
- *Soft Labeling:* soft labels can be thought of as a sort of unsupervised labeling strategy that can be applied to unlabeled data. Previous attempts at soft labeling employ the use of an auxiliary task, which teaches a model how to learn unlabeled data when knowledge of the domain is known[15]. Alternatively, we explore a soft labeling strategy in which knowledge of the domain is either not known or ignored.

  As shown in Figure 4, our soft labeling strategy begins by first feeding the Transition data through a pretrained InceptionV3 convolutional base, provided by Tensorflow [23]. We utilize the InceptionV3 base as a trained feature

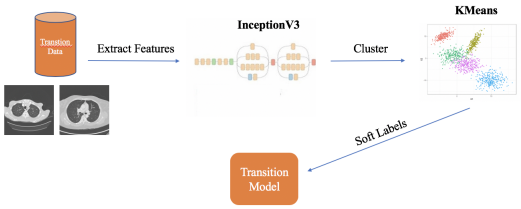

Fig. 4: Unsupervised creation of Soft Labels

extractor, which converts our original input images of dimensions (224,224,3) into feature maps of dimensions (5, 5, 2048). The resulting feature maps are then flattened and fed into a KMeans clustering algorithm, provided by scikit-learn, which clusters the data into 16 distinct feature groups [24]. We then applied corresponding labels to each cluster resulting in 16 labels: making the Transition task $T_{t\text{-}soft}$ a 16-class classification task.To select the number of clusters, we performed a grid search with cross validation of 10, evaluating clusters sizes $\{2,4,8,16\}$.

After acquiring our hard and soft labels, we configured our models to fit the Transition tasks. Because our pretrained models are loaded as convolutional bases we added randomly initialized pooling and fully connected layers, appropriate to each model. For the ResNet models we appended a 2-dimensional average pooling layer, a flatten layer, and a dense layer of 1000 nodes. For the DenseNet models we appended a 2-dimensional global average pooling layer and a dense layer of 1000 nodes. For each model the output layer either consisted of a SoftMax activation function followed by a 16 node output layer for Transition task $T_{t\text{-}soft}$ or a Sigmoid activation function followed by a single node output layer for Transition task $T_{t\text{-}hard}$.

We further configured our models by freezing the shallower half of all convolutional layers. Freezing layers is a common strategy performed when fine-tuning models for two main reasons: the first is to mitigate overfitting by reducing the total number of trainable parameters in the network; the second is to avoid redundant learning of low-level generic features that are already learned in our Source model[15].

After our models were configured for the Transition step. We split the Transition data into 80:20 Train and Validation sets, respectively. We then fine-tuned each model with Stochastic Gradient Descent and a batch size of 32, saving the weights that yielded the highest validation accuracy. Sparse-Categorical Cross Entropy loss was utilized for the Transition task $T_{t\text{-}soft}$ and Binary Cross Entropy loss was utilized for the Transition task $T_{t\text{-}hard}$.

Target step: To begin the Target step we once again randomly initialized the pooling and fully connected layers of each model. This re-initialization ensures that we are fine-tuning the fully connected layers only on the Target task. We also, added a dropout layer of 0.5 prior to the output layer of each model, to further mitigate overfitting.

As shown in Table 2, the Target dataset was split into 60:15:25 Train, Validation and Test sets, respectively. We then fine-tuned each model with Stochastic Gradient Descent and a momentum of 0.9 for 60 epochs and a batch size of 32, saving the weights that yielded the highest validation accuracy.

Dataset	COVID-19	Normal
Train	196	251
Validation	56	56
Test	97	90

TABLE II: Division of Target dataset

III. Results

In this section we present the results of our experiment outlined in the Methods section. As this paper aims to enhance the process of Single-Source Transfer Learning with a Multi-Source process, we compare the performances of our MSTL models against that of their baseline models pretrained on the ILSVRC dataset and fine-tuned on the Target dataset, referred to as the 'ImageNet' model. The performances of our finalized models will be compared against their baselines to assess the magnitude of positive or negative transfer.

After considering our two alternate labeling strategies used on our four models, we are left with 8 finalized models: ResNet50V2: Soft Labels, ResNet101V2: Soft Labels, DenseNet121: Soft Labels, DenseNet169: Soft Labels, ResNet50V2: Hard Labels, ResNet101V2: Hard Labels, DenseNet121: Hard Labels, and DenseNet169: Hard Labels.

We first plotted the Receiver Operating Characteristic (ROC), of our finalized models against their baseline 'ImageNet' models isolated by model architecture. The ROC curve plots true positive rate against the false positive rate of samples predicted from the Test set. As shown in Figure 5, when using a Hard Labeling strategy both ResNet50V2 and ResNet101V2 models outperformed its baseline AUCs by 1.8% and 5.4%, respectively. However, their performances were not as exceptional when utilizing a Soft Labeling strategy, as only the ResNet101V2 outperforms its baseline AUC by 3.4%, while the ResNet50V2 under performs by 2.9%. For the DenseNets, when using a Hard Labeling strategy, the deeper DenseNet169 outperforms its baseline AUC by 0.8%, however, the shallower DenseNet121 under performs by 0.2%. When using a Soft Labeling strategy, both DenseNet121 and DenseNet169 under perform by 1.2% and 3.0%, respectively.

We then plotted the ROC of our finalized models isolated by labeling strategy. As shown in Figure 6, our DenseNet models outperformed our ResNet models by a margin of at least 0.7% when employing a Hard Labeling strategy, and our ResNet models outperformed our DenseNet models by a margin of at least 0.9% when employing a Soft Labeling strategy. When using Hard Labels the DenseNet169 model achieved a superior AUC of 0.965, followed by the DenseNet121 with an AUC of 0.947. When using Soft Labels the ResNet101V2

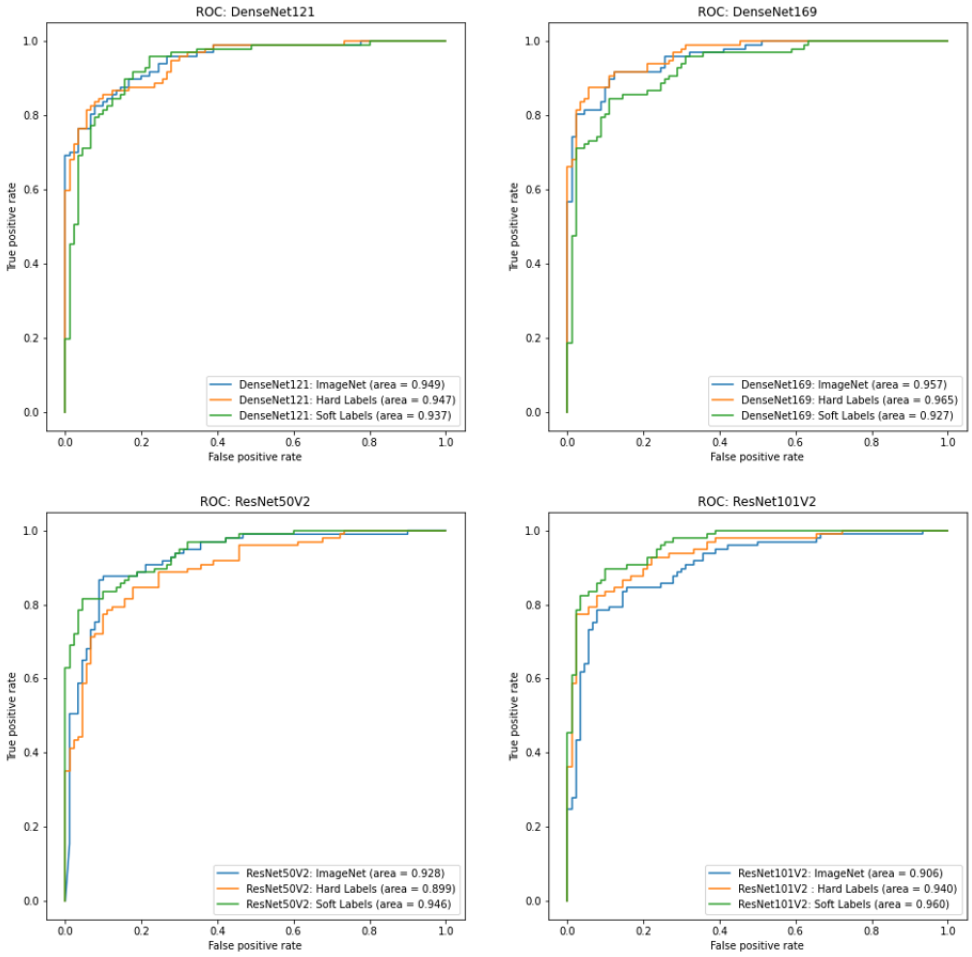

Fig. 5: ROC Curve per model architecture. True positive rate is plotted on y-axis, while false positive rate is plotted on x-axis.

model achieved a superior AUC of 0.960, followed by the ResNet50V2 with an AUC of 0.946.

F1, Accuracy, Precision and Recall scores were then calculated between the models to further evaluate the models performances. As depicted in Table 3, the DenseNet169: Hard Labels model achieves superior F1, Accuracy, and Precision scores of 0.903, 0.904, and 0.944 while the ResNet101V2: Soft Labels model achieves superior a Recall score of 0.897.

IV. Discussion

When evaluating the performances of our models it is essential that we are reminded of our research objective. As stated in the Introduction section, our purpose in this study is to develop a highly sensitive classification model for the detection of COVID-19. We seek to emphasize the use of 'sensitive', because the sensitivity of the model takes precedence over all other metrics in the case of a medical imaging diagnosis. In other words, the cost of a False Negative greatly surpasses the cost of a False Positive, especially in the diagnosis of an infectious disease. In the case of a False Positive, the worst outcome would be that a individual without COVID-19 is told to self-isolate for two weeks. In the case of a False Negative, however, the worst outcome would be that an individual with COVID-19 continues to spread the infection after receiving a negative diagnosis.

Therefore, the model with the highest Recall score is our most desirable model as it has the smallest chance of producing a False Negative. The classification results between our two highest performing models are visualized in Figure 7. As shown in the confusion matrices, although the DenseNet169: Hard Labels model has total fewer misclassifications than the ResNet101V2: Soft Labels model, the ResNet101V2: Soft Labels model has 3 fewer False Negative predictions, and thus is more desirable.

It should be noted that a significant finding of this study was the variance in performance between our DenseNets and ResNets with respect to the labeling strategy. As stated in the Methods section, the DenseNets seemed to outperform the ResNets when utilizing a Hard Labeling strategy, while the ResNets outperformed the DenseNets when utilizing a Soft Labeling strategy. We attribute these discrepancies in

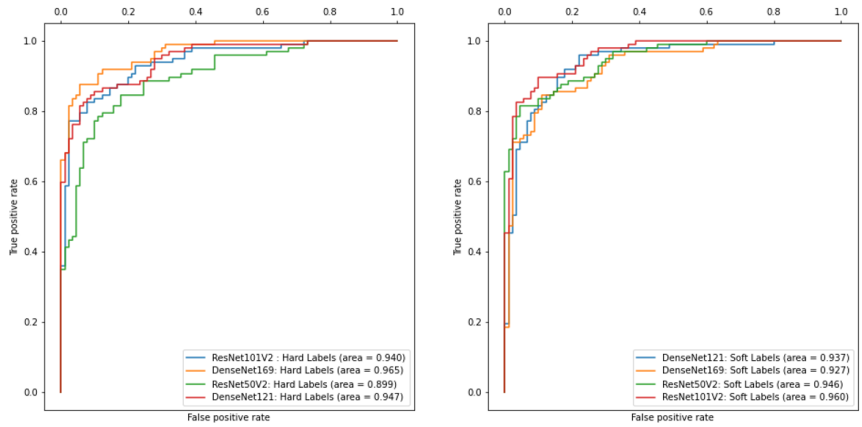

Fig. 6: ROC Curve per labeling method. True positive rate is plotted on y-axis, while false positive rate is plotted on x-axis.

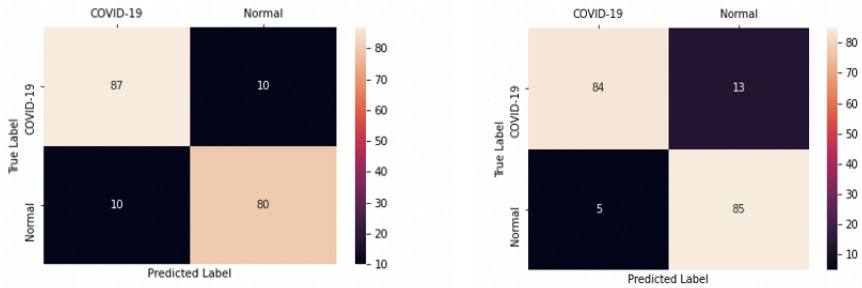

Fig. 7: ResNet101V2: Soft Labels (left) DenseNet169: Hard Labels (right)

performance largely to the number of parameters in our models. As shown in Table 1, the ResNets are much more complex than DenseNets as they retain a larger number of total parameters. This higher complexity most likely caused the ResNets to overfit more to the binary classification Transition task $T_{t\text{-}hard}$ than to the multi-class classification task $T_{t\text{-}soft}$. However, to conclude these hypotheses, further analysis is required to assess the relationship between the number of classes in $T_{t\text{-}soft}$ and the performances of our models.

A. Limitations & Future Work

Although our results appropriately reflected the aspirations of our research objective, certain limitations apply to this study.

The first limitation to this study is that CT scans are usually stored as 3-dimensional DICOM files, while our study requires an input of a 2-dimensional slice. This issue was exhibited in section 2.2, when our Transition dataset of 3-dimensional scans needed to be manually decomposed into relevant 2-dimensional slices. This preprocessing step was very computationally expensive, as it required manual selection of 10,176 CT slices displaying the regions of interest. To prevent this limitation, our study can be improved in the following ways: by employing a multi-channel CNN, or by automating the slice selection process.

If we converted our model architecture into multiple channels, then we would be able to handle 3-dimensional CT scans by processing each axial slice in parallel, eliminating the need for manual selection. While converting our pretrained models to multiple parallel channels is feasible, we would still need to acquire a Target dataset of 3-dimensional COVID-19 CT scans. This data is inherently more difficult to acquire, as DICOM files retain sensitive patient metadata and are not made as publicly available as 2-dimensional image slices. Additionally, since the model would be split into multiple channels, the number of parameters in the convolutional layers increases with each parallelization, requiring us to upgrade our computational resources appropriately. We could alleviate this complexity by sharing parameters across channels and updating them as part of an averaged loss function. This would parallelize learning, but we would also suffer loss of individualized feature extraction at each axial slice channel.

Conversely, we could attempt to eliminate manual selection by automating the selection process entirely. In this scenario,

Model	F1	Accuracy	Precision	Recall
DenseNet121 ImageNet	0.865	0.861	0.874	0.856
DenseNet121 Hard Labels	0.870	0.866	0.875	0.866
DenseNet121 Soft Labels	0.856	0.850	0.856	0.856
DenseNet169 ImageNet	0.878	0.877	0.902	0.856
DenseNet169 Hard Labels	**0.903**	**0.904**	**0.944**	0.866
DenseNet169 Soft Labels	0.833	0.840	0.904	0.773
ResNet50V2 ImageNet	0.865	0.866	0.909	0.825
ResNet50V2 Hard Labels	0.824	0.824	0.855	0.794
ResNet50V2 Soft Labels	0.866	0.866	0.900	0.835
ResNet101V2 ImageNet	0.830	0.829	0.857	0.804
ResNet101V2 Hard Labels	0.860	0.866	0.939	0.794
ResNet101V2 Soft Labels	0.897	0.893	0.897	**0.897**

TABLE III: F1, Accuracy, Precision and Recall scores

we could retain our original 2-dimensional architecture and Target dataset. The only expense would be training a independent classifier to assess if a given slice retains the region of interest we seek.

A second limitation to this study is that confounding diseases can disrupt the performances of our models. As our models were trained to distinguish COVID-19 from normal CT scans, it runs the risk of classifying other lung diseases as False Positive. Although this presents an issue, we can diminish the risk of this type of missclassification by including other diseases in our Target dataset.

V. Conclusion

In this study we presented an updated Transfer Learning approach for the classification of COVID-19 from CT scans. By learning to classify an additional dataset of images more closely related to the Target domain, our models were able to outperform baseline models fine-tuned with traditional methods. We additionally proposed an unsupervised label creation process, which further improved the performance of our Deep Residual Networks. The results of this study show the following: Transfer Learning can be improved by bridging the gap between the Source domain and the Target domain with a target-related Transition domain; unsupervised label creation has the potential to improve the performance of certain Deep CNNs; and with limited data, the application of Computer Vision for the detection of COVID-19 from CT scans exhibits high sensitivity and should be investigated further.

References

[1] "Who timeline - covid-19," World Health Organization.

[2] "Maps trends," Johns Hopkins Coronavirus Resource Center.

[3] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun, and L. Xia, "Correlation of chest ct and rt-pcr testing in coronavirus disease 2019 (covid-19) in china: a report of 1014 cases," *Radiology*, p. 200642, 2020.

[4] C. for Devices and R. Health, "Emergency use authorizations."

[5] Y. Fang, H. Zhang, J. Xie, M. Lin, L. Ying, P. Pang, and W. Ji, "Sensitivity of chest ct for covid-19: comparison to rt-pcr," *Radiology*, p. 200432, 2020.

[6] E. J. Emanuel, G. Persad, R. Upshur, B. Thome, M. Parker, A. Glickman, C. Zhang, C. Boyle, M. Smith, J. P. Phillips, and et al., "Fair allocation of scarce medical resources in the time of covid-19," *New England Journal of Medicine*, vol. 382, no. 21, p. 2049–2055, 2020.

[7] M. D. Hope, C. A. Raptis, and T. S. Henry, "Chest computed tomography for detection of coronavirus disease 2019 (covid-19): don't rush the science," 2020.

[8] M. Castillo, "The industry of ct scanning," *American Journal of Neuroradiology*, vol. 33, no. 4, p. 583–585, 2011.

[9] K. Suzuki, "Overview of deep learning in medical imaging," *Radiological physics and technology*, vol. 10, no. 3, pp. 257–273, 2017.

[10] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[11] Q. Song, L. Zhao, X. Luo, and X. Dou, "Using deep learning for classification of lung nodules on computed tomography images," *Journal of healthcare engineering*, vol. 2017, 2017.

[12] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[13] X. He, X. Yang, S. Zhang, J. Zhao, Y. Zhang, E. Xing, and P. Xie, "Sample-efficient deep learning for covid-19 diagnosis based on ct scans," *medRxiv*, 2020.

[14] C. Butt, J. Gill, D. Chun, and B. A. Babu, "Deep learning system to screen coronavirus disease 2019 pneumonia," *Applied Intelligence*, p. 1, 2020.

[15] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[17] "Spie-aapm lung ct challenge - the cancer imaging archive (tcia) public access - cancer imaging archive wiki," Cancer Imaging Archive.

[18] J. Zhao, Y. Zhang, X. He, and P. Xie, "Covid-ct-dataset: a ct scan dataset about covid-19," *arXiv preprint arXiv:2003.13865*, 2020.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[20] A. Sorokin and D. Forsyth, "Utility data annotation with amazon mechanical turk," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8, IEEE, 2008.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.