A Transfer Learning Approach to DICOM Slice Selection

Kennan LeJeune David Blincoe Sam Jenkins Chris Toomey Arthur Xin

Case Western Reserve University, Cleveland, OH
Department of Computer and Data Sciences
{kennan, drb133, soj3, ctt16, sxx132}@case.edu

Abstract

Lung tumor identification and classification is a challenging task which typically requires a trained medical professional to choose the best slices of a scan and accurately classify the chosen slices. With medical data privacy restrictions and regulations, it is difficult to collect sufficient data to construct a typical Convolutional Neural Network to choose and classify DICOM slices. We propose an inductive transfer learning approach which applies hidden layer image representations from a residual neural network to our lung nodule classifier to classify groups of slices and provide recommendations to a user as to what groups may contain possible benign or malignant nodules.

1. Introduction

Traditionally, Machine Learning problems rely on the assumption that the training data and future data domains are in the same feature space and have the same distribution (Pan, Yang). However, a large number of real world applications perform poorly when these assumptions are not met. A common instance of this is a classification task on a restricted domain of interest (e.g. classifying malignant brain tumors) with minimal or unlabeled training data, but we have sufficient training data in another domain of interest (e.g. classifying malignant lung carcinomas). Transfer Learning aims to solve this problem by adapting the knowledge acquired from a training task or training domain for use in a related domain or task. Intuitively, we can apply a solution to a problem to a different but related problem in a human-like matter.

1.1. Formal Transfer Learning Definition

Given source and target domains D_S, D_T and learning tasks T_S on the source domain and T_T on the target domain, where we aim to improve the learning of a target predictive function $f_T(\cdot) \in D_T$ using the knowledge in D_S or T_S, where $D_S \neq D_T$ or $T_S \neq T_T$. Note that in the case where both domains and tasks are equivalent, this is analogous to a standard machine learning problem. We can characterize the nature of nearly all Transfer Learning problems by considering three primary cases:

1.2. Types of Transfer Learning

1. $D_S = D_T, T_S \neq T_T$ (Inductive Transfer, 1.2.1)
2. $D_S \neq D_T, T_S = T_T$ (Transductive Transfer, 1.2.2)
3. $T_S \neq T_T$ and D_S, D_T are unlabeled (Unsupervised Transfer, 1.2.3)

1.2.1 Inductive Transfer Learning

In Inductive Transfer Learning, the domain of the source and target tasks is the same. In this case, labeled data from the target task must be available in order to induce the predictive function that we want the classifier to learn. There are two situations in inductive transfer learning: one where labeled source data is available, and one where it is not. The first scenario is similar to something known as self-taught learning, in which the classifier hopes to learn basic patterns from random unlabeled data. The second scenario is similar to multi task learning, where the classifier attempts to learn several classification tasks at the same time, except in this case we only care about the performance on a single target task, and are hoping to use knowledge learned from the other tasks in order to improve performance on said target task.

1.2.2 Transductive Transfer Learning

Transductive Transfer Learning focuses on an area in which the source and target tasks are the same, but the domains differ. The data used in the target domain is unlabeled, whereas there is an abundance of labeled data in the source domain. Transductive transfer learning can be broken down even further, into two specific scenarios: the first, in which the target and source domains have a different feature space; and the

second, where the feature space is the same, but the input data have different marginal probability distributions.

1.2.3 Unsupervised Transfer Learning

Unsupervised Transfer Learning focuses on a setting where there is no labeled data for either the source and target domains, and the target task is different from the source task. This situation is common in areas such as clustering and dimensionality reduction.

1.3. Classification Problem

Lung cancer is one of the most lethal diseases worldwide. A patient's probability of survival decreases the longer it takes to identify a cancerous tumor. Thus, successful lung cancer screenings can be used to save countless lives, making them extremely valuable. Manually inspecting cancer screenings can be very time and cost intensive, and is not necessarily error free, as classifying tumors requires advanced radiological knowledge. Computer programs, and deep learning specifically, can therefore help enhance the quality and decrease the costs associated with analyzing screenings. Given the scarcity of tumors in cancer screenings, however, and their limited availability in general, it is difficult to train models. Therefore, we can use transfer learning to learn the feature representation of images and transfer that knowledge to cancer screenings, so it does not have to learn the representation itself and can focus specifically on classifying benign or malignant tumors.

2. Related Works

2.1. Machine Learning vs Deep Learning Algorithms

In the paper by [4], the authors sought out to systematically compare the results and algorithms of several works of literature regarding the detection of lung modules in the LIDC-IDRI database. Out of the initial 1972 publications found in their search, the authors assessed the eligibility of 180 papers and found only 41 of which satisfied their for data requirements and were thus considered for their discussion. The two primary groups of algorithms are machine learning (ML) and deep learning (DL).

Amongst the algorithms found in the feature-based framework in the machine learning (ML) category, the most popular type was the Support-vector Machines (SVM). The algorithms that implemented some forms of the SVM classifier reached accuracy range of 68.4%-99.0%, sensitivity range of 55.0%-98.6%, specificity range of 87.5%-98.2%, and and AUC of 0.905-0.998.

Amongst the algorithms that were used in the deep learning category, the convolution neural network (CNN) was the most frequently used type. The algorithms that implemented a CNN architecture had an accuracy range of 82.2%-97.6%, sensitivity range of 83.1%-96.6%, specificity range of 71.4%-98.2%, and an AUC range of 0.877-0.984.

While there is a greater range in the predictions between the SVM algorithms, both categories of DL and ML algorithms contain significantly high levels of accuracy, sensitivity, and specificity. The researchers concluded that while deep learning algorithms reduce the need for feature construction of the images, there appears that some machine learning algorithms are still very useful for efficient classification. However, there appears to be no consensus on which clinical settings in which ML algorithms would be more appropriately used.

2.2. ResNet: Deep Residual Networks

In the paper [3], the authors described both the advantages and disadvantages of using CNNs to classify malignant lung nodules, and how using a deep residual network addresses its disadvantages. While a CNNs help remove the need for any hand-crafted feature engineering from images by learning discriminative features from the data directly, they work the best handling classification classes that are more visually distinct while nodule classification tasks are more subtle. Moreover, the publicly available datasets are small compared to most other image classification datasets (thousands vs millions). Therefore the researchers attempted to address the following questions in their work:

1. Does increasing the depth of neural networks help with the task of lung nodule classification?

2. How can we leverage transfer learning to achieve higher accuracy?

3. Can accuracy be improved by using a training curriculum?

The proposed solution to this problem is through using a deep residual network (ResNet). To address the first question, the researchers designed a ResNet architecture that was much deeper than the lung nodule classification systems using CNNs in their cited sources. The increased depth came as a result of advancements such as batch normalization and residual learning with ResNets. To address the second question of using transfer learning, the researchers have pre-trained on a generated column parameters using the CIFAR-10 (a large image classification dataset) before training on the smaller nodule dataset. To address the third question, the researchers described a way to quantify the difficulty of examples for the task of nodule classification and use it to create their training curriculum. In addition, they presented two main sets of results: the first of which is a report on the positive influences that deeper networks, transfer learning, and curriculum learning have on the accuracy of the classification, and the second

of which is the comparison of their system against implementations of existing models under the same training and testing conditions.

The final comparison of results showed that the deepest custom CNN found in their related works yielded a sensitivity of 89.21%, specificity of 84.96%, precision of 86.31%, AUROC of 0.9342, and accuracy of 87.18%, while their model yielded a sensitivity of 91.07%, specificity of 88.64%, precision of 89.35%, AUROC of 0.9459, and accuracy of 89.90%. This showed that by overcoming the over-fitting discrepancy between their model and existing CNN architectures, the deeper and more complex learning model with the increased representational power of having a pre-trained model allowed them to create a highly accurate model for classifying CT scans.

2.3. Nodule Malignancy Classification

This paper [1] introduces another systematics approach for lung cancer malignancy classification called NoduleX. This specific approach is based on a CNN architecture, and it takes into input a 3-D image volume and processed through a successive 2-D multi-channel convolution and max pooling layers to produce spacial 1-D "feature vector", and then processed into a final classification layer in which a softmax function predicts its output. In addition to a CNN architecture, the authors used a radiological quantitative image features (QIF) model to integrate the CNN features with radiological features. The predictive model then extracted from the CNN feature vectors concatenated with the features from QIF are then used as inputs to a trained random forest classifier.

The dataset they trained on was the LIDC-IDRI. They processed on 1065 nodules with different malignancy scores ranging from 1-5, where 1 means highly unlikely to be malignant and 5 meaning highly malignant. They then denoted the datasets as S1, S2, S3, S4, and S5, respectively. They then tested on two designs: S1 vs S45 and S12 vs S45 with 80% of the data for training and the other 20% for testing.

While this method is using a combined ML and DL algorithms, it performed significantly well. For the design of S1 vs S45, they achieving a result of 96.2% accuracy, 96.2% sensitivity, 96.2% specificity, and AUC of 0.989 for a CNN of 21 x 21 x 5 with the QIF features. For the design of S12 vs S45, the same model has found a result of 93.2% accuracy, 87.9% sensitivity, 98.5% specificity, and AUC of 0.985. This shows that although there are hand constructed features that combines the feature vectors of the CNN and that of QIF, the classification was still highly accurate in its predictions of malignant nodules.

3. Datasets

For standard transfer learning, both source and target data is needed. In this case the project utilizes ResNet 50's weights of the source images. The data that the weights were trained on comprises source domain, D_s.

LIDC-IDRI, 3.1, is used to train the output segment of the the network. This is our target domain, D_t.

3.1. LIDC-IDRI Dataset

The Lung Image Database Consortium image collection (LIDC-IDRI) dataset is a very popular cancer classification dataset that focuses on tumors located in the lungs. The lung scans are CT images of the upper torso. The entire dataset consists of 1018 cases that each contain thoracic radiologists annotations of tumor segments These tumors annotations each contain 9 different descriptors such as malignancy, calcification, and lobulation. The descriptor this paper is interested in is the malignancy of each nodule.

The malignancy is rated on a 1-5 scale. 1 being 'Highly Unlikely' of malignancy and 5 being 'Highly Suspicious' of malignancy. Using these ratings each slice in a chest CT was rated as either malignant, benign, or non-nodule. A slice was considered non-nodule if there was no nodule annotations found within the slice. To split nodules into malignant and benign labels, the annotations performed on a specific node were averaged and for malignancy values ≥ 3, the node was considered malignant and for malignancy values < 3, the node was considered benign, based upon the 4 radiologists predictions.

3.1.1 PyLIDC

To assist in the extraction of data from the DICOM image files, a python library was utilized to read to XML files which contained the annotation information for each nodule. [2]

3.1.2 Processing LIDC-IDRI Data

DICOM files, (Digital Imaging and Communications in Medicine), are the standard method for transferring and communicating image data. The structures of these files are extremely robust and offer many access in the form of 'Tags'. In the case of LIDC-IDRI, the dataset is composed entirely of CT images which must be processed by first transforming the image data along the HU (Hounsfield scale) given the transformation coefficients in the DICOM.

The vertical slice size must also be taken into account because CT scans can be ordered in a variety of ranging resolutions from (< 0.1mm to > 3mm). A scale of 1 mm per slice was chosen and the pixel data was transformed.

4. Project Structure

4.1. Residual Neural Network

The vanishing gradient has long been a problem when constructing neural networks with large architectures. Essentially, the back-propagating the gradient to earlier layers makes the gradient tend towards zero, so the performance can degrade in earlier layers. This can result in worse performance for deeper models than shallower ones, since earlier layers will perform worse for the deeper model. Residual neural networks attempt to solve this problem, by using "shortcuts" that jump layers in order to make sure the gradient does not become infinitesimally smaller. Figure 1 illustrates this concept. The input, X, is both passed to the next layer and skips ahead to the layer after, in order to add the effects of the input and the activation function and avoid the vanishing gradient problem.

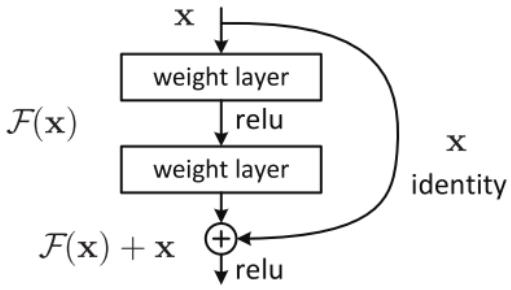

Figure 1. Residual Node

For our project, we are using the ResNet-50, 2,pre-trained model for Keras. The architecture of the model can be seen in the image below.

As a whole, the model has 5 stages, followed by a pooling and flattening step to output the data. The first stage puts the data through a convolutional layer that then runs ReLU and max pooling, and then the model consists of four stages convolutional blocks and identity blocks, with varying number of identity blocks. The convolutional blocks runs the input through two iterations of convolutional layers, followed by ReLU, and a final convolutional layer, and adding to this the input run through a convolutional layer, before running a final iteration of ReLU. The identity block does the same, but simply adds back the input in the end rather than a convolutional layer run on the input. The difference in the two is the shape of the input and output: if their dimensions line up, then an identity block is sufficient, otherwise the input must be re-shaped to match the output.

4.2. Model Sectors

A unique approach of our solution is to bin together specific locations of lung CT scans into a bucket. In other words, there exists a hyper-parameter called *sector_num*$= x$ that will train x independent ResNet models, 4.1 on separate physical portions of the CT scans. For example, every studies full chest CT scan was scaled to be just 65 slices. If the *sector_number* $= 13$, there will be 13 bins starting with bin 1 at the top of the chest and bin 13 at the bottom. Here, each bin will be comprised of 5 slices from each study.

A bucket is defined as all scans from a specific sector over all studies.

Once there are x independent CNN models trained, the output of the classifier given a study will be the slice numbers that should be looked at closely.

5. Experiments

Three primary ideas were tested in this project. One was to determine if this model generally performed well on the given data and if it performed well compared to the results from 2.1. The second goal was to analyze if shifting the *sector_num* as seen in 4.2, increased or decreased the performance of our model. The third was to see if down-sampling the given CT scans to a lower resolution contributed to a significant drop in accuracy.

Given the way our problem setup is designed, a significant amount of slices from the CT scans did not contain tumorous growths. This led to an extremely skewed label distribution. To fix this problem for testing, the number of slices from each sector that did not contain any tumors was limited so the label distribution was well distributed among the three classes (non-nodule, benign, and malignant).

5.1. General Performance

To evaluate the general performance of the model above, 4,the following parameters are used as a baseline.

Number of Sectors: 5
Scan size: 128×128

The above figure 3 shows the accuracy and loss of the general baseline model over 10 epochs for the first sector.

The baseline model for each sector was run on 2010 scans. The whole baseline model was trained on $10,050$ scans.

5.2. Varying Sector Number

5.2.1 1 Sector Model

In this model, there are 1 sectors used to divide the chest CT. This is a decrease of 4 sectors from 5.1.

Number of Sectors: 1
Scan size: 128×128

The above figure 4 shows the accuracy and loss of the 1 sector model over 10 epochs.

This model was run on $10,050$ scans.

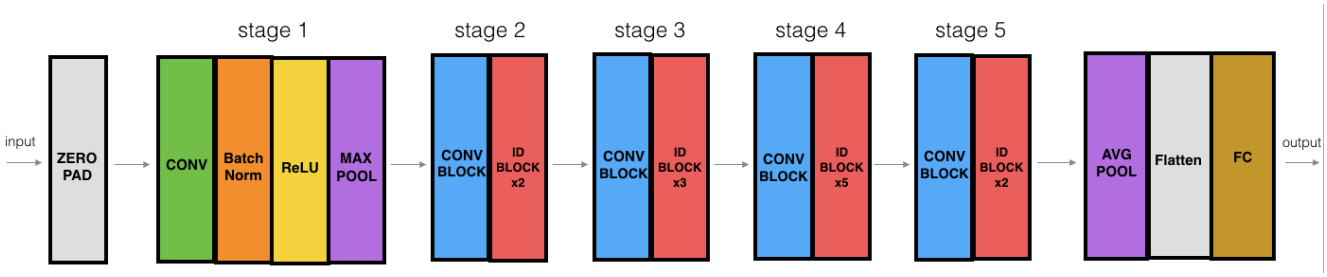

Figure 2. Architecture of ResNet-5

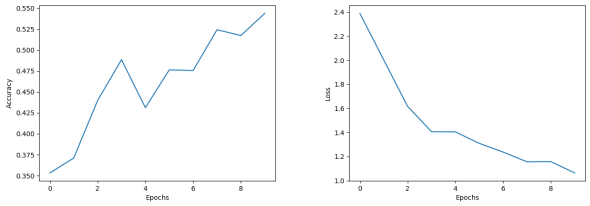

Figure 3. General Baseline Accuracy and Loss(Sector 1)

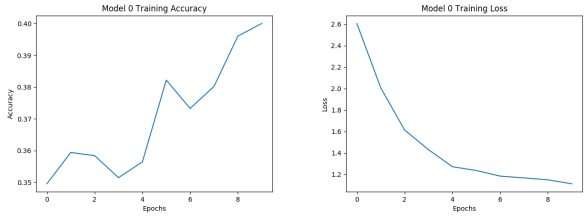

Figure 4. 1 Sector Model Accuracy and Loss (Sector 1)

5.2.2 13 Sector Model

In this model, there are 13 sectors used to divide the chest CT. This is an increase of 7 sectors from 5.1.

Number of Sectors: 13
Scan size: 128×128

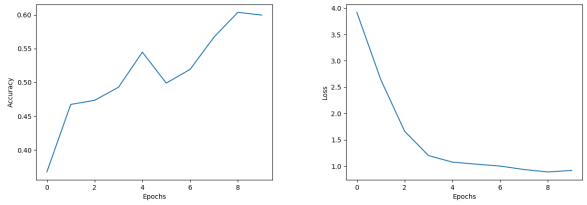

Figure 5. 13 Sector Model Accuracy and Loss (Sector 1)

The above figure 5 shows the accuracy and loss of the 5 sector model over 10 epochs for the first sector.

The baseline model for each sector was run on 774 scans. The whole baseline model was trained on $10,062$ scans.

5.3. Varying Image Resolution

In this model, the number of pixels is reduced by half. This is a from 128×128 to 64×64 sectors from 5.1.

Number of Sectors: 5
Scan size: 64×64

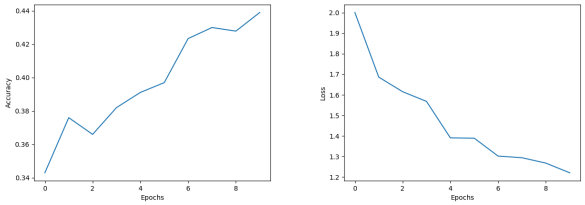

Figure 6. 64x64 Model Accuracy and Loss (Sector 1)

The above figure 6 shows the accuracy and loss of the 5 sector model over 10 epochs for the first sector.

The model for each sector was run on 2010 scans. The whole baseline model was trained on $10,050$ scans.

6. Results

6.1. General Performance

In the first epoch, the accuracy is essentially randomly choosing between the three labels, but it incrementally climbs until on the 10th and final epoch, it reaches about 55%. A final accuracy of 55% does not seem very high, but given this a multi-class problem, this is still significantly higher than the expected value if the classifier randomly chose a label.

6.2. Varying Sector Number

6.2.1 1 Sector Model

While the 1 sector started out essentally guessing at epoch 1, the accuracy did not decrease as quickly as the baseline mode. After 10 epochs, the accuracy ended up at $\approx 40\%$. This is roughly 15% lower than the baseline model. This means that overall, increasing the sector number, increases the accuracy.

6.2.2 13 Sector Model

Similar to the base model, which had 5 sectors, the first epoch of the 13 sector model ends with an accuracy approximately the same as random guessing. However, the accuracy of the 13 sector model actually climbs slightly higher, to a final accuracy of about 60%. This supports the claim that dividing the image into more sectors allows for more accuracy, though it takes longer to run.

6.3. Varying Image Resolution

Changing the image resolution decreased the accuracy significantly. It climbs only slightly, from initially around random to a final of about 40%, which finishes 15% below that of the base model. This is likely because the image is less clear and harder to differentiate between what is and what is not a tumor.

7. Conclusion

Generally, the classifier improves as we increase the number of sectors and increase the image resolution. As far as the number of sectors, this could be for a couple reasons. It may be because of the high amount of variability between sectors with lots of slices, since slices in different areas of a lung can be extremely different from each other.

A second reason is that since we are dividing a single classification problem into n classification problems, each separate problem can converge to its own local minimum, which when combined, are more likely to be higher than the local minimum of a single classification.

Increasing the image resolution allows the classifier to better represent features that differentiate malignant and benign tumors. Since it has more details to work with, it makes sense that as the resolution increases, the accuracy of the model increases as well.

7.1. Further Investigation

7.1.1 Identifying Tumors

One approach to tumor identification is the use of autoencoders rather than CNNs or residual neural networks. In the autoencoder context, 2.1 gives some context on how autoencoders have been used in this area. In one such example, lung cancer screenings with tumors are input into the autoencoder, and initially, the output is the same size as the input and identifies the tumor, with the rest of the picture blank (i.e. it only identifies the tumor in the output). This process is repeated and the autoencoder learns how to identify the varying shapes and sizes of tumors within lungs. This extension could be paired with another algorithm, like ours, to identify whether a tumor is malignant or benign.

7.1.2 Classifying Tumors

Once we can identify a tumor in a lung, the problem becomes whether that tumor can be harmful. 2.2 uses ResNet in a slightly different way than our implementation. Rather than scanning a full lung, they use an image of the tumor itself instead, and with this image, they create three images from this original image, each representing the different axes of the image (coronal, sagittal, and axial). Each of these is then input into a corresponding ResNet, setting up three different models. In order to both identify and diagnose a tumor, we could use a similar method to that used in 2.2. First, in order to identify the tumor, we could run along the coronal plane of a scan. Then, when we have identified the slices of the lung that have a tumor (if it exists), we can use all three planes (coronal, sagital, and axial) in order to diagnose the tumor. This would allow us a full dimension worth of data more than we are currently using, so it likely would help with accuracy.

References

[1] J. Causey, J. Zhang, S. Ma, B. Jiang, J. Qualls, D. Politte, F. Prior, S. Zhang, and X. Huang. Highly accurate model for prediction of lung nodule malignancy with ct scans. *Scientific Reports*, 8, 02 2018.

[2] M. Hancock. Pylidc. `https://github.com/notmatthancock/pylidc`, 2018.

[3] A. Nibali, Z. He, and D. Wollersheim. Pulmonary nodule classification with deep residual networks. *International journal of computer assisted radiology and surgery*, 12, 05 2017.

[4] L. Pehrson, M. B. Nielsen, and C. Lauridsen. Automatic pulmonary nodule detection applying deep learning or machine learning algorithms to the lidc-idri database: A systematic review. *Diagnostics*, 9:29, 03 2019.