Performance of Traditional Image Processing and Convolutional Neural Network Techniques in Classifying Interstitial Lung Disease

Jackson Meixensperger, Spencer Perry, Dr. Gurman Gill

Department of Computer Science, Sonoma State University, Rohnert Park

Introduction

Chest computed tomography (CT) scans are used by radiologists to detect and classify Interstitial Lung Disease (ILD), however the use of computer-aided diagnostic (CAD) systems can reduce the time taken for these decisions with less intervention from radiologists. We explore the performance of both traditional image processing and convolutional neural networks (CNN) techniques in classifying ILD. Traditional image processing has long been successful at recognizing and classifying images into defined groups, but some may consider the technology outdated. On the other hand, CNN's possess very powerful computing capabilities but require large image sets to train and evaluate. We hope to bypass this issue using "transfer learning", where input images are trained and tested on pre-existing model weights.

Data Sets

ImageNet Dataset (used to pre-train the CNN)

- > 14 Million Images
- 100,000 Categories
- Sample Images:

Talisman Test Suite Data Set

- 19321 Images from 96 Patients
- 5 Categories (disease types)
- Sample Images:

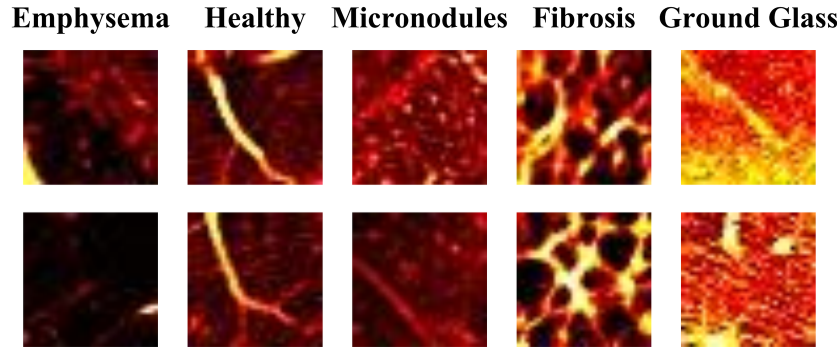
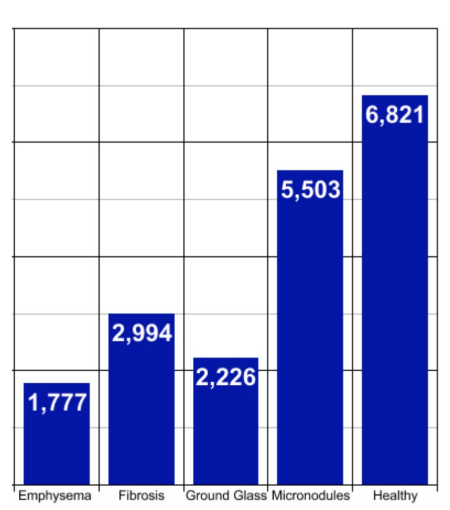


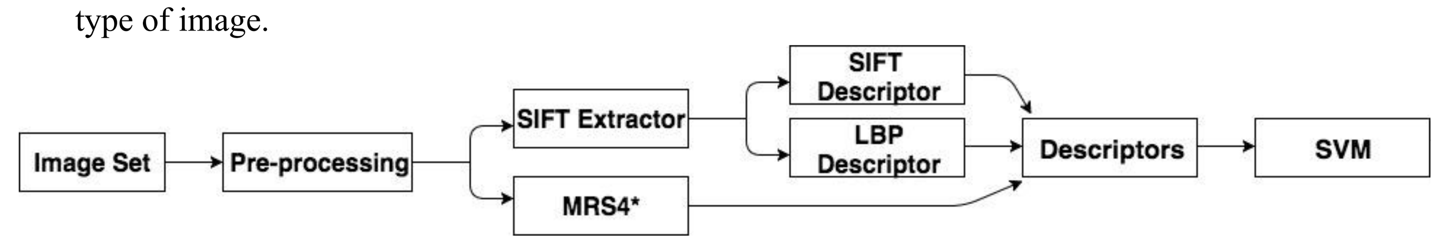
Image Count per Disease



Brief Introduction to CNNs & Traditional Image Processing

Traditional Image Processing

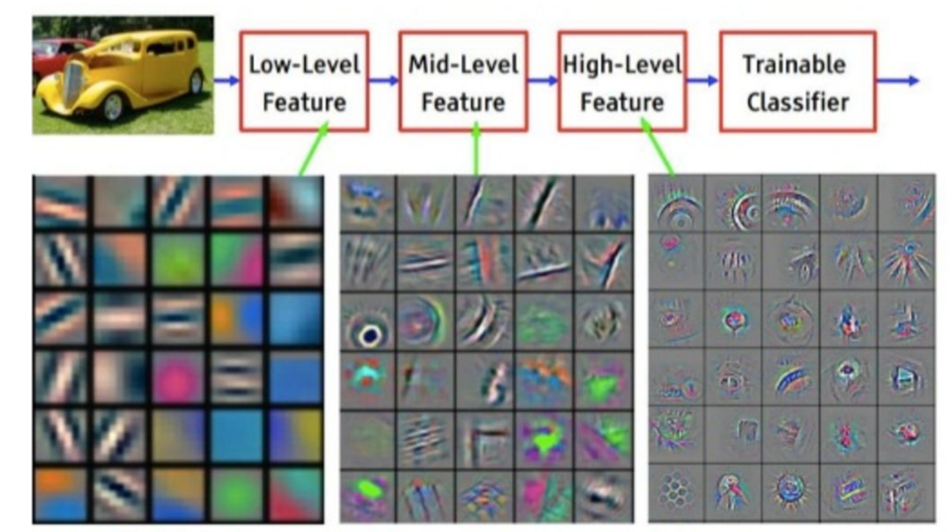
- A traditional approach to learning and recognizing classes of images utilizes a feature extractor, feature descriptor, and classifier.
- A binary Support Vector Machine (SVM) can classify the presence or absence of 1 class. This *binary* comparison determines whether the image is of our class, or if it is some other undefined type of image.



Traditional Pipeline with several options of feature extraction/description

Convolutional Neural Networks

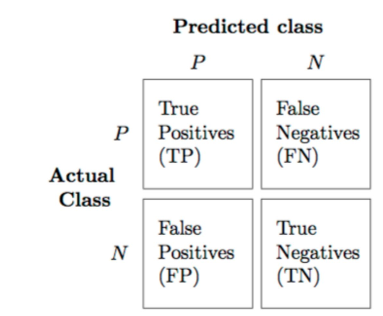
- Uses layers composed of multiple convolutional kernels
- Each kernel uses 2D matrices as input, and outputs some new 2D matrix of features.
- When training, a CNN takes in images and optimizes parameters to minimize a loss function.
- Most CNN's increase in complexity as the layers progress. Beginning layers may find constructs like color and edges, while later layers detect complex ones like faces.



Evaluation Measures

Confusion Matrix: contingency table showing the actual class versus the predicted class for all images.These values can be normalized to obtain a rate for each value.

F1 Score: an average of precision and recall. Scores exist in values from 0 to 1, where 1 represents the best possible score.



$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + T_n}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

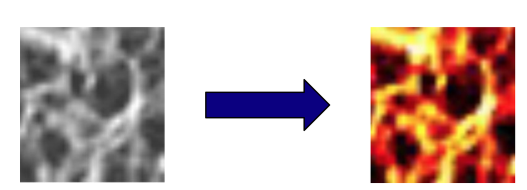
Methodology

HU to RGB

Each pixel of a CT scan holds a Hounsfield unit (HU) value, representing the density of material at the pixel. However, the methods we're using expect color channel values. RGB images consist of 3 color channels that have pixel values between 0 and 255. We linearly mapped:

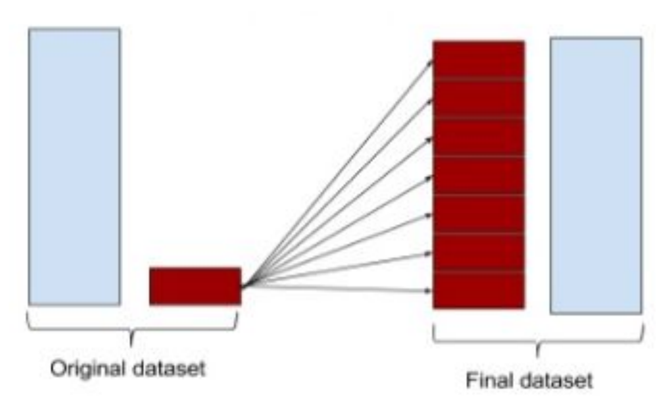
- HU range (-1000, -600) to the Red Channel
- HU range (-601, -200) to the Green Channel
- HU range (-201, 200) to the Blue Channel

Each HU range looks for different anatomic features in CT-Scans, such as air, blood vessels, bone, etc. Values outside of the HU ranges of each channel were mapped to either 0 or 255.



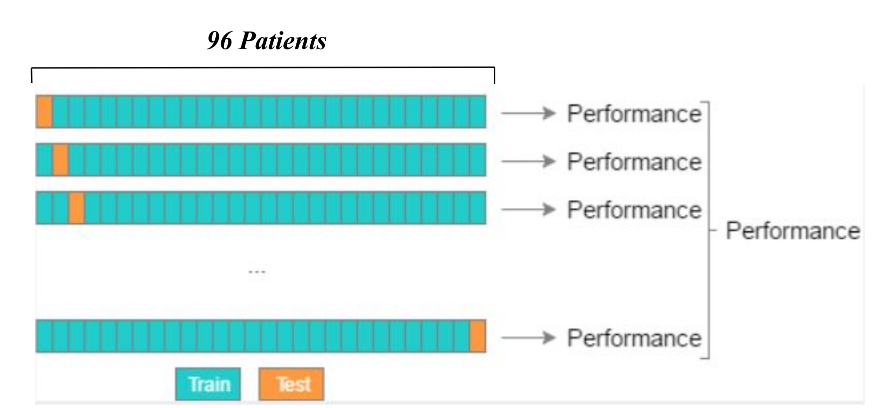
Data Imbalance

When class sizes are imbalanced, any CNN results relating to accuracy will be skewed. To solve this, we oversample classes with fewer images, taking random samples with replacement until the class sizes are equal. This ensures all test results are consistent and unbiased by class imbalance.



Leave One Patient Out Cross Validation

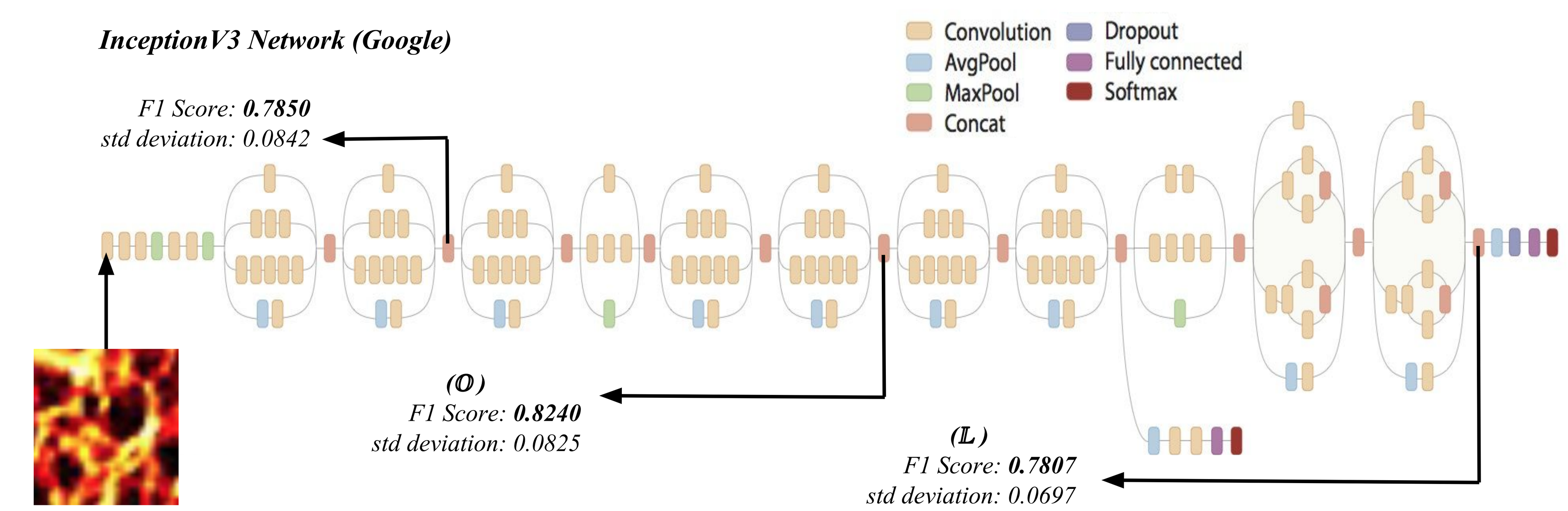
By removing all images belonging to a single patient, we can be sure that the training data and the subsequent model won't be biased by the single patient's images. This patient's images can then be tested using our new model. Doing this for each patient allows us to combine our results and make an estimate of model performance for any patients.



One cell represents a single patient, each comprising of several images

Transfer Learning

Using weights pre-trained on the ImageNet dataset, the network takes RGB images in its input layer. We first pull features from the last feature layer in the network (𝕃) and append our own softmax layer to classify the images. The process is repeated on earlier, intermediate layers of the network to find an optimal layer (𝕆) which produces the highest F1 Score. With a smaller number of layers, the time spent traversing them will reduce drastically, and the features extracted will be simpler constructs.



Experimental Set-Up & Results

Binary Comparison

To measure the performance and Traditional Image Processing and CNN's in similar fashions, we used a binary classifier for both methods. This classifier determines only whether an image is *healthy* or *diseased* (any class of ILD). From the traditional pipeline (introduced above), the SIFT extractor/descriptor performed best for our task. We used it as a baseline for traditional processing and collected various Keras CNN models to test against it. For our binary measurements, transfer learning was used on each model at the last feature layer (𝕃).

Multi-Class Comparison

CNN models were then tested with a multi-class classifier to see how their performance would be affected. This new classifier defines an image as: *healthy, fibrosis, emphysema, ground glass,* or, *micronodules*. Since the models appeared to perform much worse here, we applied transfer learning further to improve scores. We measured 2 scores for each model tested: 𝕃, the score from using Transfer Learning on the last layer; and 𝕆, the score from using Transfer Learning on the optimal layer.

Traditional Processing

Binary F1 Score: 0.7918

TensorFlow Keras

CNN's

Model Name	Binary F1 Score	𝕃 F1 Score	𝕆 F1 Score
InceptionV3	0.8739	0.7807	0.8240
InceptionResnetV2	0.8513	0.7934	0.8051
ResNet50	0.8754	0.7894	0.8033
VGG16	0.8611	0.7619	0.7960
VGG19	0.8579	0.7702	0.7702
Xception	0.8486	0.7747	0.8211

Highest score for given test (green): InceptionV3 𝕆 0.8240; InceptionResnetV2 𝕃 0.7934; ResNet50 Binary 0.8754

No improvement from previous test (orange): VGG19 𝕃 0.7702; VGG19 𝕆 0.7702

CNN Performance: Last Layer(𝕃) vs Optimal Layer(𝕆)

CNN's far outperformed our traditional image processing techniques when it came to binary classification, however there was a substantial drop off for multi-class classification. Further application of transfer learning proved to be a viable solution to this problem, as it increased the performance of almost every model that was tested. It also showed that features found after the optimal layer won't be beneficial to our classification when using the pre-trained ImageNet weights.

The following normalized confusion matrix represents the results for the optimal layer (𝕆) in the InceptionV3 network (pictured in "Transfer Learning"):

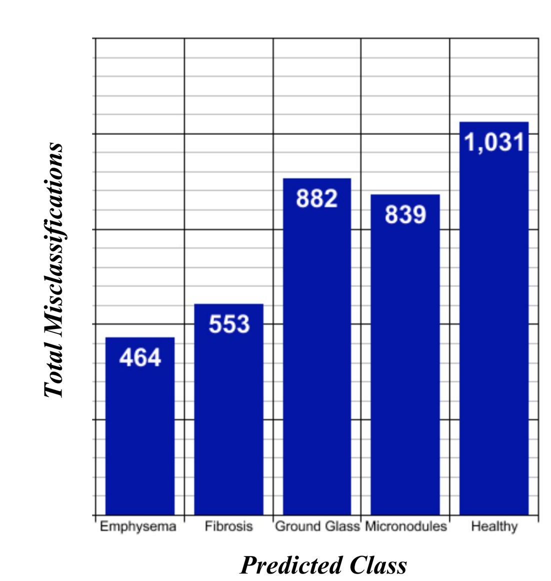
(Predicted)

(Actual)	E	F	G	H	M
E	0.78	0.04	0.02	0.13	0.03
F	0.02	0.88	0.07	0.01	0.02
G	0.02	0.11	0.7	0.10	0.07
H	0.03	0.01	0.10	0.76	0.11
M	0.03	0.03	0.03	0.09	0.83

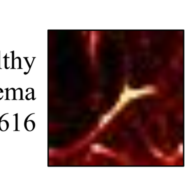
Misclassifications

After analyzing commonly misclassified images by their true labels, predicted labels, and confidence scores (**c**), we found:

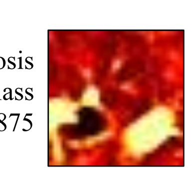
- Dark images are usually classified with high confidence scores as emphysema.
- Incorrectly predicting healthy made up the largest portion of our total misclassifications.
- Higher values in the confusion matrix corresponded to more common misclassifications (examples below)



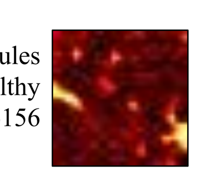
Predicted: Healthy
Actual: Emphysema
(c): 0.8616



Predicted: Fibrosis
Actual: Ground Glass
(c): 0.6875



Predicted: Micronodules
Actual: Healthy
(c): 0.6156



Conclusions

- CNN's are more feasible than traditional image processing techniques for classifying ILD.
- Transfer learning is a useful tool to customize a CNN model for ILD classification.
- Performing "Data Augmentation" for emphysema images or using a more established database of images may yield higher scores.
- Future work involves "fine-tuning" later layers in networks, based on the observed optimal transfer layers in each network.

Work done from funding by the Koret Scholar Foundation, Data provided by Adrien Depeursinge