

Capstone Project Phase B

Idiopathic pulmonary fibrosis classification using optimized Convolutional Neural Network

Alex Baboshin 310926415

Asaf Shnaider 316468636

(24-2-R-9)

Supervisor:

Prof. Miri Weiss Cohen

Table Of Content

[Abstract 2](#_Toc196889274)

[Introduction 3](#_Toc196889275)

[Usual interstitial pneumonia 3](#_Toc196889276)

[Idiopathic pulmonary fibrosis 3](#_Toc196889277)

[CT scan 5](#_Toc196889278)

[Modalities of CT scan 6](#_Toc196889279)

[Background and Related Work 7](#_Toc196889280)

[Background 7](#_Toc196889281)

[Data augmentation 7](#_Toc196889282)

[Convolutional Neural Network 7](#_Toc196889283)

[Convolutional layer 8](#_Toc196889284)

[Activation function 8](#_Toc196889285)

[Max pooling 9](#_Toc196889286)

[Flatten layer 10](#_Toc196889287)

[Dense layer 10](#_Toc196889288)

[Dropout layer 11](#_Toc196889289)

[Related work 11](#_Toc196889290)

[Proposed approach 12](#_Toc196889291)

[Network architecture 12](#_Toc196889292)

[Hyperparameters 12](#_Toc196889293)

[Flow 21](#_Toc196889294)

[Database 21](#_Toc196889295)

[Research Process 22](#_Toc196889296)

[Process 22](#_Toc196889297)

[Milestones 22](#_Toc196889298)

[Challenges and Achievements 23](#_Toc196889299)

[Results 24](#_Toc196889300)

[Conclusions 25](#_Toc196889301)

[References 26](#_Toc196889302)

[Citations 26](#_Toc196889303)

[Figures 27](#_Toc196889304)

# Abstract

Idiopathic pulmonary fibrosis (IPF) is a progressive interstitial lung disease characterized by irreversible scarring of the pulmonary interstitium, leading to respiratory failure and a median survival of approximately four years post-diagnosis. Early and accurate detection of IPF on high-resolution computed tomography (HRCT) scans is critical for timely intervention, yet manual interpretation remains time-consuming and subject to inter-observer variability. In this study, we propose an optimized convolutional neural network (CNN) architecture to automatically classify CT lung slices as IPF-positive or IPF-negative.

Our approach employs a CNN composed of four convolutional/max-pooling blocks, followed by a flattening stage, a fully connected dense layer with dropout regularization, and a final softmax output layer. We performed extensive hyperparameter tuning—evaluating learning rates (5×10⁻⁴ to 5×10⁻⁶), batch sizes (32, 64), dropout rates (0.3–0.5) and epochs (50–150)—to identify the configuration that maximizes generalization while minimizing overfitting. The model was trained on a combined dataset of 4,391 grayscale CT slices (2,269 IPF-negative; 2,122 IPF-positive), with 70% used for training, 5% for testing, and 25% for validation. Data augmentation (flips, brightness change, contrast change) further enriched the training set.

We assessed performance via accuracy, precision, recall, F1-score, and confusion matrices. The optimal configuration—batch size 64, 150 epochs, learning rate 5×10⁻⁶, dropout 0.45—yielded balanced training and validation curves, with minimal divergence and stable convergence. On the held-out test set, the model achieved approximately **95.45% accuracy**, precision of 0.9652, recall of 0.9487, and an F1-score of 0.9569, indicating strong discriminative capability and low false-positive rate.

These results demonstrate that an appropriately regularized CNN, combined with a sufficiently large and diverse dataset, can accurately identify IPF from CT scan images. Our optimized model offers a promising tool to assist radiologists in early IPF detection, potentially improving patient outcomes through faster diagnosis and treatment planning.

# Introduction

## Usual interstitial pneumonia

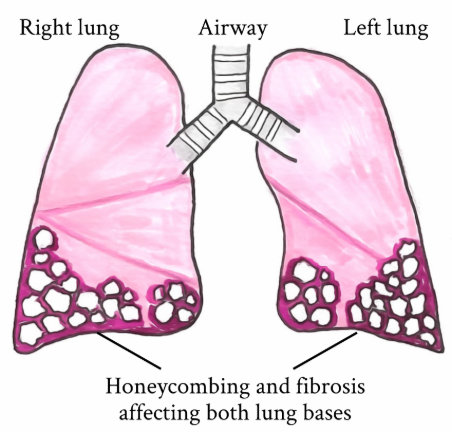
Usual interstitial pneumonia (UIP) is a type of lung condition marked by the gradual development of scarring in both lungs. [[1]](#one) This scarring, known as pulmonary fibrosis, affects the lung's interstitium—the tissue that supports the lung structure. Because of this, UIP falls under the category of interstitial lung diseases.

The word "usual" highlights that UIP is the most frequently encountered type of interstitial fibrosis. Despite the term "pneumonia" suggesting an infection, in this context it broadly refers to any abnormal lung condition, including fibrosis and inflammation.

Common symptoms of UIP include a gradual onset of breathlessness and a persistent cough lasting for several months. For some individuals, the disease is only identified after a sudden worsening of symptoms leads them to seek medical care.

The exact cause of scarring in UIP may be identifiable in some cases, though it is more often unknown. When the cause is unclear, the condition is referred to as idiopathic pulmonary fibrosis (IPF) [[2]](#two), with "idiopathic" meaning "of unknown origin. Known causes of UIP can include autoimmune diseases like rheumatoid arthritis, drug reactions, chronic hypersensitivity pneumonitis, exposure to asbestos, and genetic conditions such as Hermansky–Pudlak syndrome.

Diagnosis of UIP may be made either through imaging techniques, such as a chest CT scan performed by a radiologist, or through analysis of lung tissue obtained by biopsy, examined by a pathologist.



[Figure 1](#Figure1)

## Idiopathic pulmonary fibrosis

Idiopathic pulmonary fibrosis (IPF), also known as cryptogenic fibrosing alveolitis, [[3]](#three) is a rare and progressive respiratory disease. It is marked by the thickening and stiffening of lung tissue due to the development of scar tissue (fibrosis). IPF is a form of chronic pulmonary fibrosis that leads to a gradual and irreversible decline in lung function.[[4]](#four)

The lung tissue becomes thickened and stiff, particularly impacting the areas surrounding the air sacs (alveoli). his condition usually manifests as gradually worsening shortness of breath and a dry cough. Other symptoms may include persistent fatigue and nail clubbing, where the fingernails and toenails become abnormally large, and dome shaped. In some cases, complications such as pulmonary hypertension, heart failure, pneumonia, or pulmonary embolism may develop.[[5]](#five)

The precise cause remains unknown—hence the term "idiopathic".[[5]](#five) However, several risk factors have been identified, including cigarette smoking, gastroesophageal reflux disease, certain viral infections, and genetic predisposition.[[5]](#five) The condition is driven by the scarring of lung tissue, which disrupts normal function.[[5]](#five) Diagnosis is primarily based on excluding other potential causes, and it is often supported by high-resolution CT scans or lung biopsy findings that reveal a pattern of usual interstitial pneumonia. This condition falls under the same category of interstitial lung diseases [[4]](#four) (large group of diseases that cause scarring of the lungs).

Pulmonary rehabilitation and supplemental oxygen often provide significant benefits. Medications such as pirfenidone and nintedanib may help slow the progression of the disease, and lung transplantation can be considered as an option for certain patients.[[5]](#five)[[6]](#six)

Globally, around 5 million people are affected by this disease.[[7]](#seven) Its incidence is approximately 12 cases per 100,000 individuals each year. The condition most frequently occurs in peoples in their 60s and 70s, and it is more common in males than in females.[[8]](#eight) On average, the life expectancy following diagnosis is about four years.[[5]](#five)

Various artificial intelligence technologies have been developed to assist in the diagnosis of the disease. One deep learning algorithm designed to categorize high-resolution CT images demonstrated high accuracy.[[9]](#nine) In addition, a research project led by Nagoya University Graduate School of Medicine and Riken utilized a combination of deep learning and machine learning algorithms to achieve accurate diagnostic outcomes.[[10]](#ten)

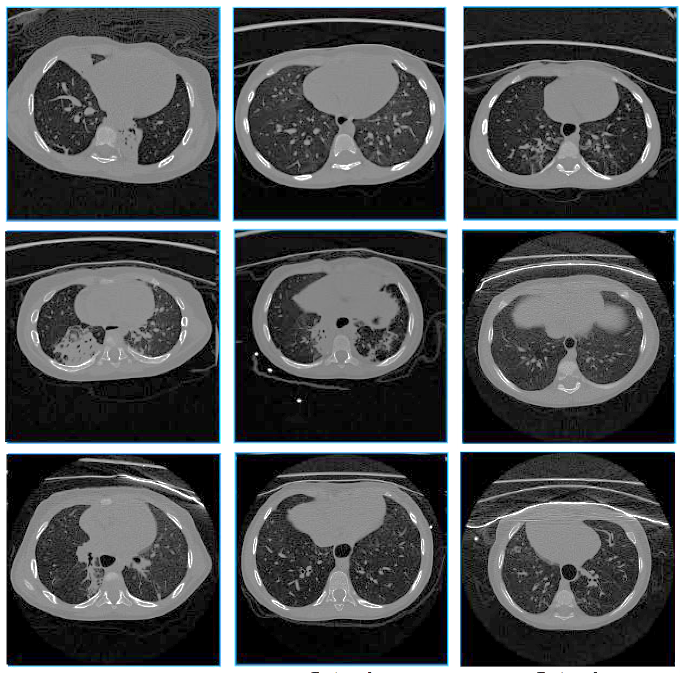
## CT scan

A computed tomography scan (CT scan) is a medical imaging technique that provides detailed internal images of the body.[[11]](#eleven) The professionals who operate CT scanners are known as radiographers or radiology technologists.[[12]](#twelve)[[13]](#thirteen) CT scanners work by using a rotating X-ray tube paired with a row of detectors arranged in a gantry to measure how X-rays are absorbed by different tissues. The X-ray measurements from various angles are processed using tomographic reconstruction algorithms, resulting in cross-sectional (virtual "slice") images of the body. Additionally, CT scans are particularly useful for patients with metallic implants or pacemakers, for whom magnetic resonance imaging (MRI) may be contraindicated.

The detectors in a CT scanner do not directly produce an image. Instead, they measure the transmission of a thin beam of X-rays (typically 1-10 mm thick) as it passes through the body. Images of a particular section are captured from different angles, which allows the system to reconstruct depth-information in the third dimension.

A CT image is digital, represented as a square matrix of pixels where each pixel corresponds to a voxel (volume element) of the patient's tissue.

Typically, a CT image consists of 512 rows with 512 pixels per row, creating a matrix of   
512 x 512 = 262,144 pixels (and voxels). During image processing, the attenuation coefficient is calculated for each voxel corresponding to these pixels. [[14]](#fourteen)



[Figure 2](#Figure2)

## Modalities of CT scan

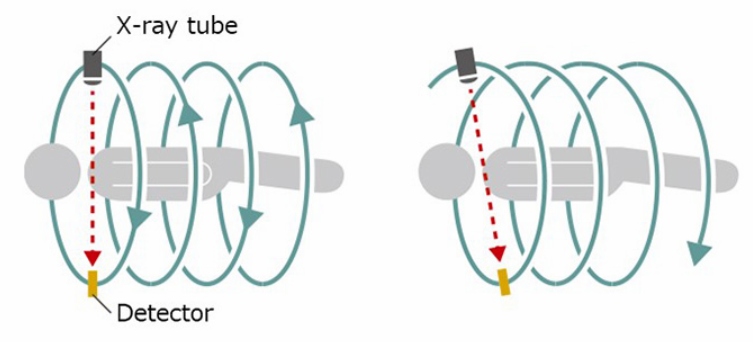
The most common and conventional CT scan imaging sequences:

* **Sequential CT** image acquisition, also known as "scan-move-scan" or "step and shoot," was the conventional method in computed tomography before the introduction of helical CT. In this technique, the patient is advanced along the scanner's longitudinal axis and paused at intervals to acquire trans-axial images, as the gantry’s rotation is limited by high-tension cabling rather than modern slip ring technology. This approach yields only axial slices, has longer acquisition times, and is more susceptible to motion artifacts. Although helical CT now predominates, modern scanners still offer sequential (axial) acquisition—especially for head and gated cardiac exams—to minimize helical artifacts and reduce radiation dose by avoiding over-ranging.[[15]](#fifteen)
* **Helical (or spiral) CT** image acquisition marks a significant improvement over earlier sequential ("step and shoot") methods by enabling a continuous, three-dimensional data set to be acquired rapidly—as the patient moves through a rotating x-ray beam and detector, the resulting images can be reconstructed from a helical path, often within a single breath-hold. This continuous acquisition minimizes misregistration from patient movement or breathing, which is why most modern CT protocols favor helical scanning when large examination volumes are required. The overall radiation dose during helical acquisition depends on the patient’s speed through the scanner (pitch), and while the approach offers increased speed and improved image quality, it also poses challenges like complex reconstruction demands, potential over-ranging leading to higher doses, and slight motion artifacts due to table movement.[[16]](#sixteen)

**Sequential CT** excels in high-precision axial imaging (e.g., brain), while **Helical CT** is faster, more efficient, and preferred for most body imaging applications.

**Sequential CT**

**Helical/Spiral CT**



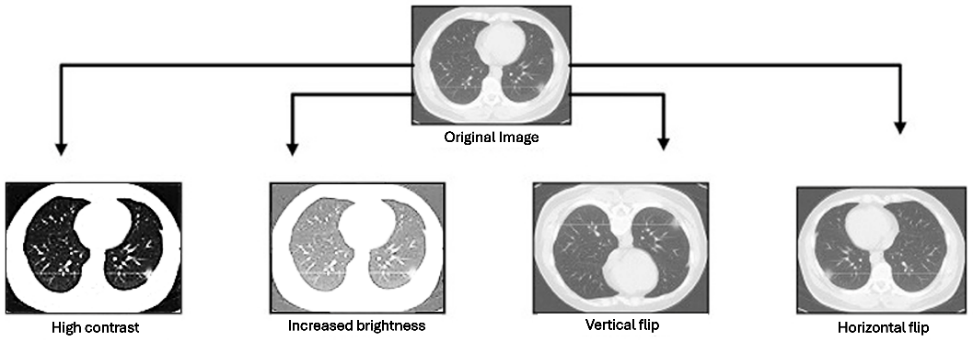
[Figure 3](#Figure3)

# Background and Related Work

## Background

### Data augmentation

Data augmentation is a statistical approach that facilitates maximum likelihood estimation from incomplete datasets.[[17]](#seventeen)[[18]](#eighteen) It serves as a fundamental tool and is extensively employed in machine learning to restrain overfitting – which occurs when a neural network model learns not only the general patterns in the training data but also the noise and specific details that do not generalize to unseen data, which leads to high accuracy on training data but poor accuracy on test or validation data. This works by expanding the training set with multiple, subtly altered versions of the original data.[[19]](#nineteen)

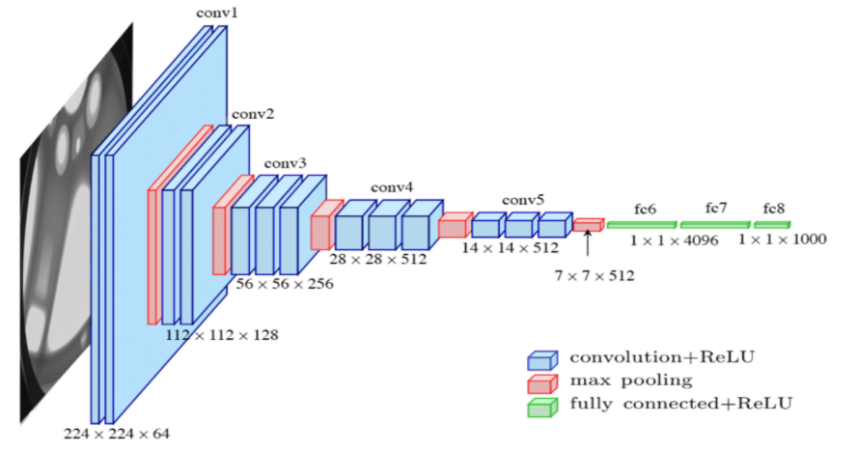


[Figure 4](#Figure4)

### Convolutional Neural Network

A convolutional neural network (CNN) is a kind of neural network which learns to find useful patterns in data by using filters. It’s a popular deep learning method used to understand and make predictions from different types of data, like text, pictures, and sound.[[20]](#twenty) CNNs have been the main choice for tasks involving images and computer vision.[[21]](#twenty_one)

A convolutional neural network has three main parts: an input layer, hidden layers, and an output layer. The hidden layers include one or more convolution layers. In a convolution layer, a small filter  
(called a kernel) slides over the input data. At each position, the filter and the matching patch of input are multiplied element‑by‑element and then summed (a dot product), and that sum is passed through a ReLU activation. As the filter moves across the input, it builds a feature map, which becomes the input for the next layer. After the convolution layers, the network usually adds pooling layers to shrink the data, fully connected layers to mix all the features, and normalization layers to keep the values stable.[[22]](#twenty_two)

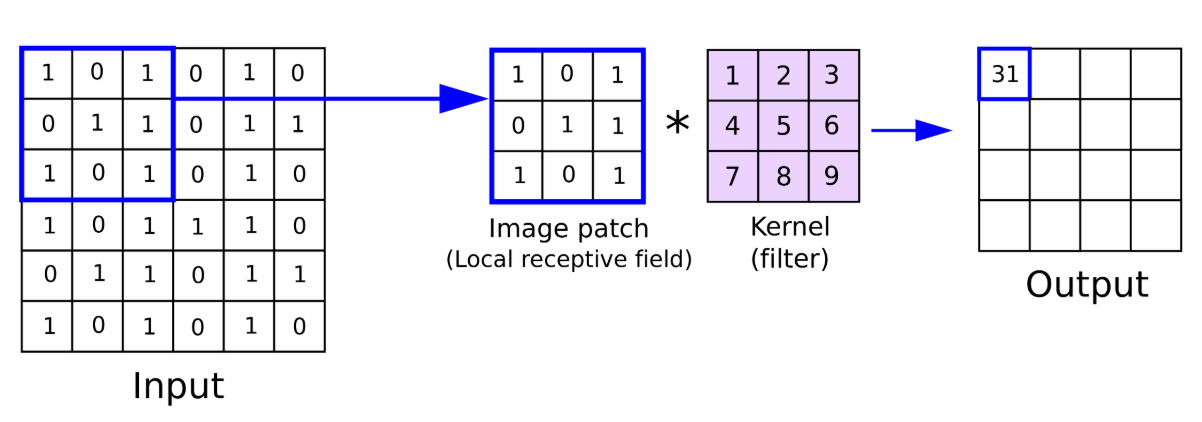


[Figure 5](#Figure5)

### Convolutional layer

In a convolutional layer, also called Conv2D layer, the convolution operation includes moving a small window, referred to as a kernel or filter, across the entire input dataset. At each position, the kernel overlays a segment of the input, and a dot product is computed between the kernel’s values and the corresponding input values. This calculation yields a single output value for that location, and by repeating this process across the input, a new matrix known as the feature map is formed.

This feature map acts as a distilled representation of the input, emphasizing specific patterns or features such as edges, textures, or other notable characteristics. Essentially, the convolution operation serves both to extract these important features and to introduce a degree of spatial invariance, meaning that the network can recognize the same patterns regardless of their location within the input. This makes convolutional layers especially powerful for processing images, videos, and other structured data.[[23]](#twenty_three)[[24]](#twenty_four)

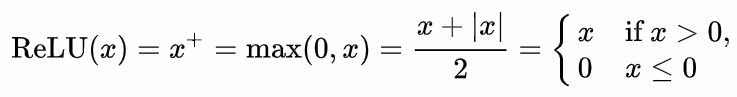


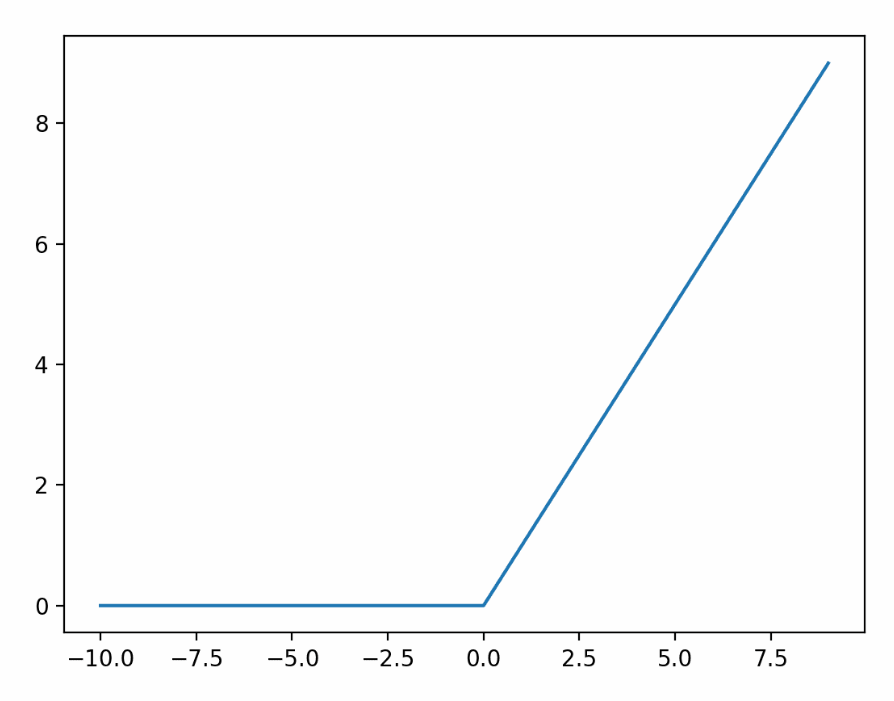
[Figure 6](#Figure6)

### Activation function

An activation function in an artificial neural network is a mathematical function that determines the output of a node by processing its weighted inputs. When the activation function is nonlinear, even a network with only a few nodes can effectively solve complex problems.[[25]](#twenty_five)

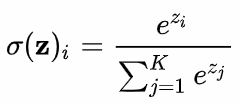
* ReLU (**rectified linear unit**)[[26]](#twenty_six) is one of the most popular activation functions for artificial neural networks, defined as the non-negative part of its argument. The [ramp function](https://en.wikipedia.org/wiki/Ramp_function), where x is the input to a neuron, is:

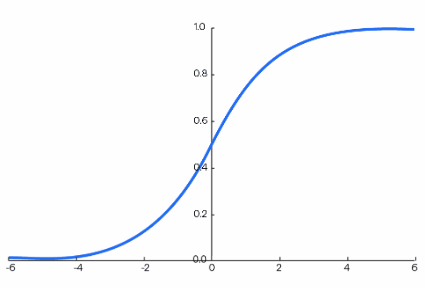




[Figure 7](#Figure7)

* SoftMax[[27]](#twenty_seven) is an activation function which transforms a vector of K real values into a probability distribution across K possible outcomes. It extends the logistic function to multiple dimensions and is commonly applied in multinomial logistic regression. In neural networks, SoftMax is typically used as the final activation function to convert the output into a probability distribution over the predicted classes.





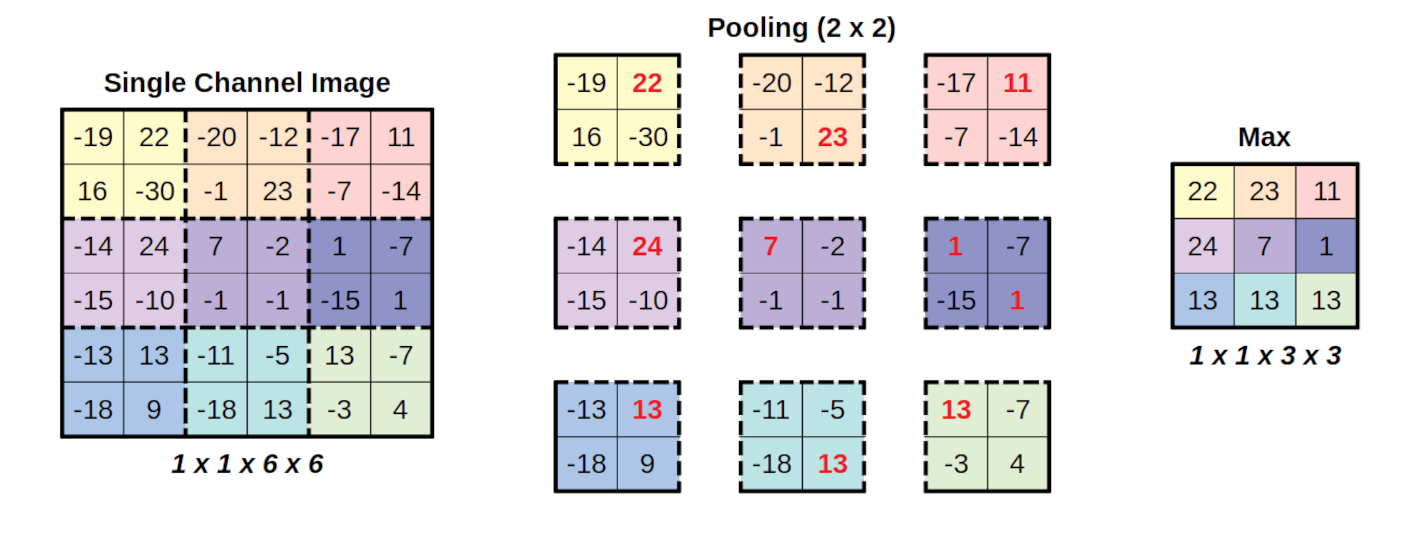
[Figure 8](#Figure8)

The SoftMax function transforms a vector of real numbers into a vector of probabilities. Each probability in the result is in the range 0...1, and the sum of the probabilities is 1.

(In our case, K=2 – IPF Positive or Negative)

### Max pooling

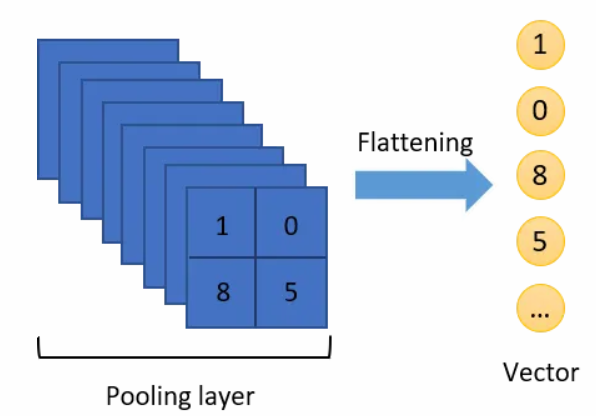
Pooling layers offer a method for downsampling feature maps by summarizing the presence of features within local regions. Max pooling is generally the most effective pooling method at capturing distinct and informative spatial features compared to other pooling methods, like average pooling, as it highlights the most prominent activations within each region.



[Figure 9](#Figure9)

### Flatten layer

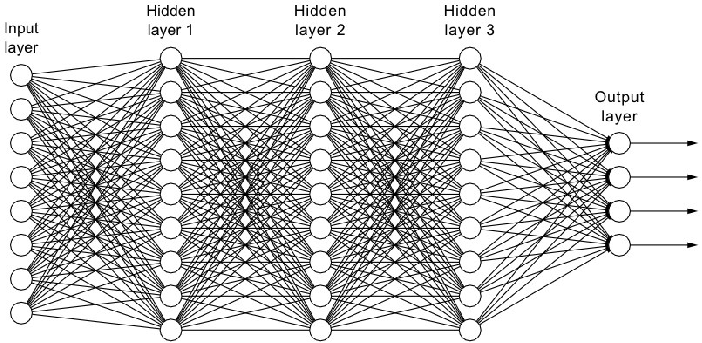
The Flatten layer[[28]](#twenty_eight) takes a multi‑dimensional tensor and transforms it into a single, continuous one‑dimensional vector. In other words, it removes the spatial dimensions of the data, performed by convolutional/pooling layers, converting the original multi‑dimensional array into a flat, linear tensor, for the fully connected (Dense) layers.



[Figure 10](#Figure10)

### Dense layer

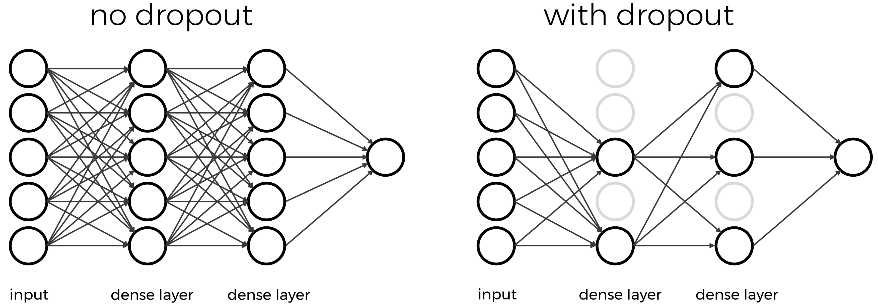
A dense layer—also known as a fully connected layer—creates abstract representations of its inputs by linking each of its neurons to every neuron in the previous layer. In multilayer perception, which is a feedforward neural network [[29]](#twenty_nine), as well as in CNN, several of these dense layers are stacked one after another.[[30]](#thirty)



[Figure 11](#Figure11)

### Dropout layer

Dropout is a straightforward yet powerful regularization method. During training, it randomly “turns off” a portion of neurons—often between 20% and 50%—in each layer. In practice, any given neuron may be temporarily set to zero at a particular training step. This forces the network to rely on multiple neurons to represent features, rather than depending on a small subset of neurons.[[31]](#thirty_one)



[*Figure 12*](#Figure12)

## Related work

Automated lung disease detection, with and without relation to IPF, has been acknowledged in previous studies and remains a relevant and interesting research topic. Here is a summary of the main contributions:

In the department of computer science engineering at Sri Sairam Engineering College in Chenni, India, research was performed on predicting severity of lung function decline in IPF based on LSTM (a type of recurrent neural network designed to learn long-term dependencies and retain information over longer sequences).   
The paper presents a predictive framework that learns from each patient’s time‑series data—namely high‑resolution CT (HRCT) scans, serial forced vital capacity (FVC) measurements, and basic demographics (e.g. sex, smoking status)—to forecast future FVC values. By doing so, it aims to quantify and predict the severity of lung function decline in idiopathic pulmonary fibrosis, enabling earlier intervention and better management of this progressive, currently incurable disease. [[32]](#thirty_two)

In a Heliyon research from 2024[[33]](#thirty_three), a study proposes three progressively broader deep‑learning models for classifying pulmonary diseases from chest X‑ray images —CovCXR‑Net, MDCXR3‑Net, and MDCXR4‑Net—for the automatic detection and classification of pulmonary diseases (COVID‑19, pneumonia, and pulmonary opacity) from chest X‑ray images, achieving state‑of‑the‑art accuracy while using far fewer parameters and computational resources.

In a 2018 paper “Lung Cancer Detection using CT Scan Images”[[34]](#thirty_four), the authors evaluate and compare existing computer‑aided lung cancer detection methods on CT images, identify their key limitations (notably in preprocessing and malignancy classification), and then propose a novel model—incorporating median and Gaussian noise filtering, enhanced feature extraction, marker‑controlled watershed segmentation, and SVM‑based benign vs. malignant classification—to achieve higher overall detection accuracy and provide diagnostic classification in a single pipeline.

Our contribution in this field is to try detecting IPF Positive lungs using CNN by exploring different hyperparameters and defining the most suitable set of parameters for our datasets and model.

# Proposed approach

## Network architecture

Our network architecture is a CNN with 4 pairs of Conv2D and MaxPooling2D layers, following flattening layer to convert the output into a vector and then passing it through a fully connected dense layer with a dropout of 20-50 percent to prevent overfitting, and finally arriving to a layer of 2 possible outputs – IPF Positive and IPF Negative.  
The model receives at each step of the training a batch size of 32 or 64 images where each image as a size of 256x256 in a gray scale and with data augmentation to increase our training dataset with more variety of images.  
For batch size, epochs amount, learning rate and dropout rate we tested several options mentioned in the hyperparameters section.

## Hyperparameters

In our project, we tried out different settings and checked how they affect the model's performance to detect IPF from the dataset of CT images. The goal was to find the best combination that gives good overall results and avoids overfitting. We have tested the results of 4 different performance measurements: **Accuracy, Precision,** **Recall (Sensitivity)** and **F1 Score.**

While researching similar health related projects, we discovered that there is a main emphasis on F1 Score which is particularly important in medical applications where both false positives and false negatives have critical implications.

The Hyperparameters we have tested are:

**Learning rates**: 5e-4, 5e-5, 5e-6.

Affects how fast the model learns.

Lower values were more stable but slower, also required more computational resources which made it harder to evaluate our current system.

**Batch sizes**: 32 vs 64.

We come to realize that bigger batch sizes results in faster training, but also sometimes worse outcomes.

**Dropout rates**: 0.3 to 0.5.

Helped to reduce overfitting. The higher the dropout rate the better results we got.

**Epoch sizes:** range from 30 to 150.

More training rounds (Epochs) helped the model to learn more, but too many caused overfitting, expecially for our limited dataset.

The table below shows the runs and hyperparameters we chose:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Run # | Batch Size | Epochs | Learning rate | Dropout | Augmentation |
| 1 | 32 | 70 | 5e-5 | 0.3 | No |
| 2 | 32 | 50 | 5e-4 | 0.3 | No |
| 3 | 64 | 150 | 5e-5 | 0.5 | No |
| 4 | 64 | 100 | 5e-6 | 0.5 | Yes |
| 5 | 64 | 150 | 5e-6 | 0.5 | Yes |
| 6 | 64 | 150 | 5e-5 | 0.5 | Yes |
| 7 | 64 | 150 | 5e-6 | 0.45 | yes |

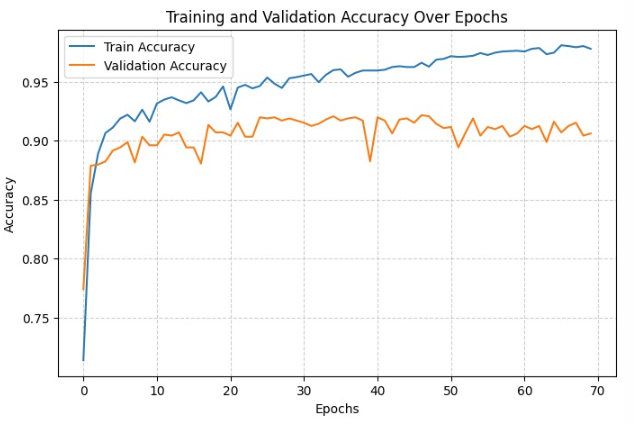
#1 run

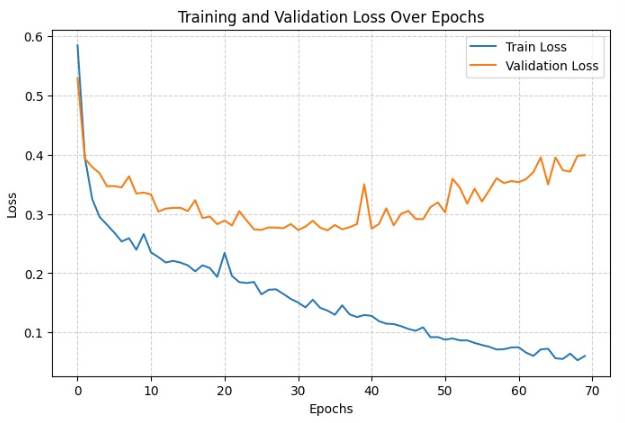
BATCH\_SIZE = 32

EPOCHS = 70

LEARNING\_RATE = 5e-5

DROPOUT\_RATE = 0.3





A blue squares with white text

AI-generated content may be incorrect.

🏁 Final Results:

📈 Final Test Accuracy : 0.8318

📉 Final Test Loss : 0.4860

Accuracy : 0.8318

Precision: 0.8175

Recall : 0.8803

F1 Score : 0.8477

Augmentation False

Model underfitted: low recall, low precision.

Very low LR (5e-5), no augmentation led to underfitting, missclassified 37 cases this is the worst scores.

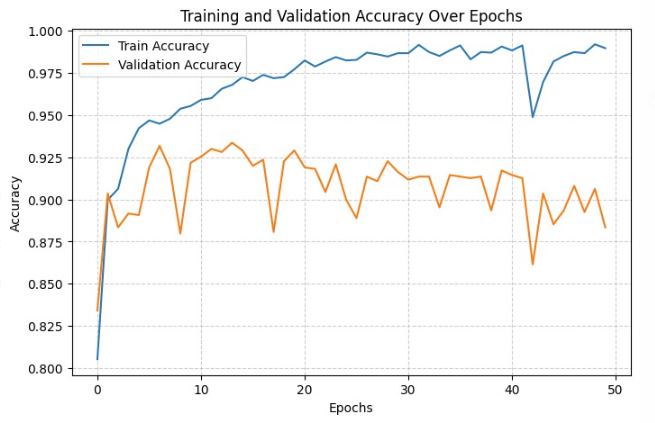
#2 run

BATCH\_SIZE = 32

EPOCHS = 50

LEARNING\_RATE = 5e-4

DROPOUT\_RATE = 0.3



A graph showing the loss of a train

AI-generated content may be incorrect.

A diagram of a diagram

AI-generated content may be incorrect.

🏁 Final Results:

📈 Final Test Accuracy : 0.8682

📉 Final Test Loss : 0.6690

Accuracy : 0.8682

Precision: 0.8438

Recall : 0.9231

F1 Score : 0.8816

Augmentation False

Instability between train/validation loss ,until epoch 10 the model is learning useful patterns, after that training loss keeps dropping meaning the model memorizing the training data and validation loss increases, model fails to generalize unseen validation data anymore.

Batch size 32 caused more noise. Dropout was low (0.3).

#3 run

BATCH\_SIZE = 64

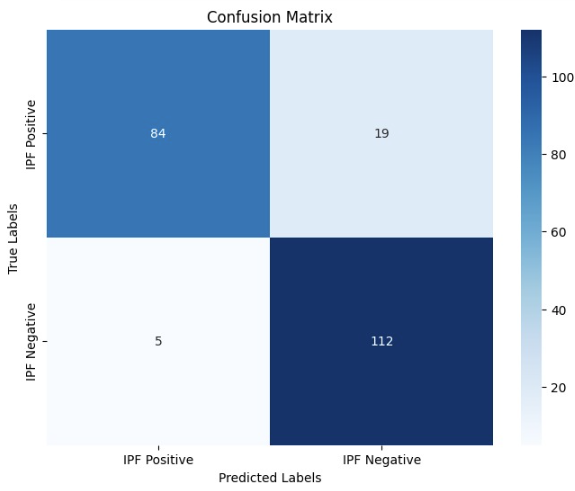
EPOCHS = 150

LEARNING\_RATE = 5e-5

DROPOUT\_RATE = 0.5

A graph showing the results of training and validation

AI-generated content may be incorrect.

🏁 Final Results:

📈 Final Test Accuracy : 0.8909

📉 Final Test Loss : 0.5199

Accuracy : 0.8909

Precision: 0.8550

Recall : 0.9573

F1 Score : 0.9032

Augmentation False

Overfitting after 150 epochs, validation loss grew at around epoch 60.

High epochs without augmentation aren't useful. Needed augmentation or early stopping.

#4 run

BATCH\_SIZE = 64

EPOCHS = 100

LEARNING\_RATE = 5e-6

DROPOUT\_RATE = 0.5

A graph of a graph showing the value of training

AI-generated content may be incorrect.

A diagram of a blue box

AI-generated content may be incorrect.🏁 Final Results:

📈 Final Test Accuracy : 0.9091

📉 Final Test Loss : 0.3345

Accuracy : 0.9091

Precision: 0.9619

Recall : 0.8632

F1 Score : 0.9099

Augmentation True

Augmentation helped, but LR was very small (5e-6).

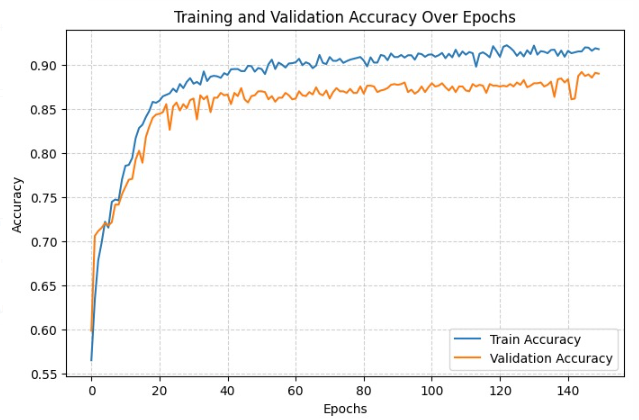
Better to balance LR and augmentation together, try to increase epochs to see if results improve.

#5 run

BATCH\_SIZE = 64

EPOCHS = 150

LEARNING\_RATE = 5e-6

DROPOUT\_RATE = 0.5

A graph showing the loss of a graph

AI-generated content may be incorrect.

A diagram of a confusion matrix

AI-generated content may be incorrect.🏁 Final Results:

📈 Final Test Accuracy : 0.9227

📉 Final Test Loss : 0.2923

Accuracy : 0.9227

Precision: 0.9808

Recall : 0.8718

F1 Score : 0.9231

Augmentation True

Running these parameters for 150 epochs definitely improved results, Precision very high, recall a bit lower.

Very strong augmentation helped raise precision but slight recall sacrifice.

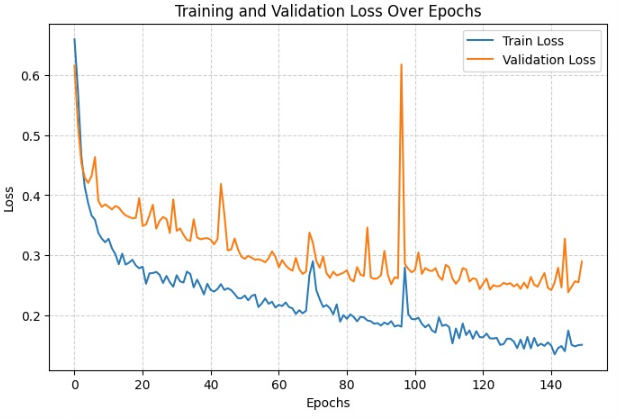
#6 run

BATCH\_SIZE = 64

EPOCHS = 150

LEARNING\_RATE = 5e-5

DROPOUT\_RATE = 0.5



A diagram of a diagram

AI-generated content may be incorrect.🏁 Final Results:

📈 Final Test Accuracy : 0.9364

📉 Final Test Loss : 0.2504

Accuracy : 0.9364

Precision: 0.9256

Recall : 0.9573

F1 Score : 0.9412

Augmentation True

Best results: highest F1, great accuracy.

Augmentation + tuned hyperparameters gave the best model overall.

The model missclassifed in total just 14 cases out of total 220 which is the lowest of all our runs, considering the limited dataset we used of 4391 CT scans , these results are the best we got.

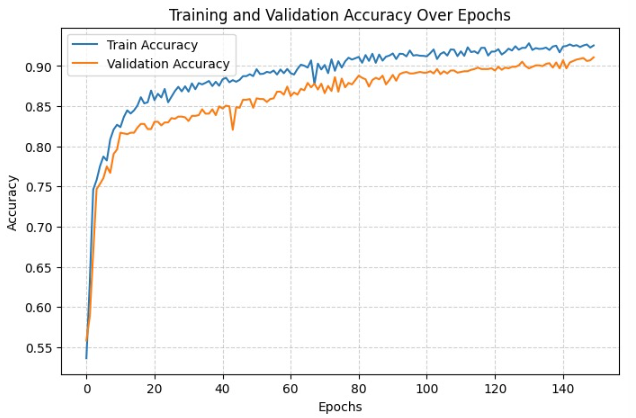
Both training and validation accuracy are increasing together over epochs, there are 3 points where they touch but after touching they keep increasing together and the gap remains small, some random spikes likely duo to noise in the batch smaples which is acceptable.

#7 run

BATCH\_SIZE = 64

EPOCHS = 150

LEARNING\_RATE = 5e-6

DROPOUT\_RATE = 0.45

A graph of a graph showing the loss of a number of people

AI-generated content may be incorrect.

A diagram of a diagram

AI-generated content may be incorrect.🏁 Final Results:

📈 Final Test Accuracy : 0.9545

📉 Final Test Loss : 0.1964

Accuracy : 0.9545

Precision: 0.9652

Recall : 0.9487

F1 Score : 0.9569

Augmentation True

Accuracy curve has smooth convergence unlike the previous run, the training and validation accuracy having small gap indicating good sign of generalization. Loss curve is also steady and smooth, no spikes meaning the model is stable. Only 10 misclassifications out of 220 test samples,best result we got , FP=4, FN =6 , balanced preformance for both classes (no bias toward one label).

considering the limited dataset we used of 4391 CT scans, this configuration is optimal and balanced.

F1 score above 0.95 considered excellent in medical classification tasks.

## Flow

תמונה שמכילה טקסט, צילום מסך, קו, גופן

תוכן שנוצר על-ידי בינה מלאכותית עשוי להיות שגוי.

## Database

Train Set: 3073 images

- IPF Positive: 1631 images (53.08%)

- IPF Negative: 1442 images (46.92%)

Test Set: 220 images

- IPF Positive: 103 images (46.82%)

- IPF Negative: 117 images (53.18%)

Validation Set: 1098 images

- IPF Positive: 535 images (48.72%)

- IPF Negative: 563 images (51.28%)

Grand Total: 4391 images

- Training: 69.98%

- Testing : 5.01%

- Validation: 25.01%

# Research Process

## Process

### Milestones

* Our project started by researching idiopathic pulmonary fibrosis, focusing on its radiographic manifestations, particularly via computed tomography scanning, and the prevailing industry-standard imaging modalities.
* We selected Python as the development language because of its extensive scientific ecosystem, particularly libraries such as NumPy for calculation and computation, as well as TensorFlow with Keras which provided us with all the necessary frameworks for constructing and training our convolutional neural network model.
* To address the significant computational requirements of model training, we used a Google Colab Pro+ environment, which provides enhanced CPU and GPU capabilities to execute calculations and achieve network convergence within a practical timeframe.
* During our initial experiments, we ingested the raw CT dataset without any preprocessing and quickly encountered quality issues: several slices contained predominant bone structures with minimal lung tissue. To address this, we supplemented our collection with additional non IPF-lung CT scans drawn from other Kaggle repositories, ensuring a more representative balance of IPF positive and negative images.
* Despite expanding the dataset, training still collapsed into severe overfitting. Validation accuracy jumped wildly, even after simple augmentation strategies to synthetically enlarge the dataset, prompting a deeper investigation that revealed our images were loaded in RGB rather than their true grayscale colors. Correcting the color channels alone proved insufficient, as the overall sample size remained too small for robust generalization.
* Finally, with an enlarged, correctly formatted dataset in hand, we introduced conservative augmentations (e.g. flips, brightness and contrast adjustments) without adding new images.

Through iterative adjustments, we identified optimal hyperparameter combinations that consistently yielded strong performance on both training and validation sets. These refinements transformed the loss and accuracy curves into stable, convergent training dynamics.

* For each hyperparameter configuration, whether adjusting learning rate, batch size, dropout rate, or number of epochs, we executed a dedicated training run. During each run, we monitored both training and validation performance, computing the following metrics at every training – Accuracy and Loss function graphs, a table with Accuracy, Precision, Recall and F₁-Score, and finally a Confusion Matrix (displayed graphically to inspect true positives, true negatives, false positives, and false negatives).
* After each training process, we logged all these statistics for both the training and validation sets. This comprehensive logging allowed us to pinpoint features specific learning that cause overfitting, or convergence issues as they came up.
* Once all individual hyperparameter runs were complete, we conducted a comparative analysis of their logs, plotting metrics and examining confusion matrices, to identify which settings yielded the most balanced improvements across Accuracy, Precision, Recall and F₁-score. Finally, we performed a conclusive “meta‐run” that combined the best values for each hyperparameter, resulting in our optimized model configuration as seen in our results section.

### Challenges and Achievements

Throughout the project we ran to both some technical challenges as well as physical challenges that we managed to find solutions for:

* We found our first dataset from a Kaggle dataset[[35]](#thirty_five) which included 3D CT scans with 2D slices of lungs with IPF diagnosed patients only. The images were loaded as .dcm format so we've built a python script to convert that dataset to .png format.
* We started building our model to work with the dataset without modifying it first. We then realized that the dataset was missing crucial parts like labels (IPF and Non-IPF) as well as that the dataset did not contain at all images of Non-IPF CT scans of lungs.
* It took us a long time to get the right dataset because there was no specific public dataset which had both IPF CT scans of lungs as well as Non-IPF CT scans.  
  Moreover, the datasets we found at the beginning , were quite lacking for our needs, so to overcome this challenge we first tried to use data augmentation to expand our small dataset, by creating 3-4 duplicates with tweaks (like rotation, zoom, crop, etc.,) for each original image, but it took too much RAM and also made the training phase crash several times until we found out the cause. Eventually, we ended up building our dataset by gathering CT images of healthy lungs (IPF negative) from several Kaggle sources[[36]](#thirty_six) and finally managed to build our dataset.
* At first, we decided to use Google Colab on standard version because the size and amount of the images in the dataset we ended up with made it hard to run locally. Later we found out that the training phase was stopped before finishing due to not having enough RAM, so we upgraded to Google Colab Pro+ which fixed that issue.
* One of our biggest challenges was overfitting: with only about 650 training and 100 validation images, the model simply memorized the training set, achieving 100% accuracy with minimal loss, while validation accuracy remained very low, and loss stayed high. We tried reducing the number of epochs to avoid over-training, standardizing image sizes to limit input complexity, and adding data augmentation (random flips, rotations, and shifts) to expose the network to more variability, yet validation performance was still lacking.
* Finally, we discovered that our grayscale images were being loaded as three-channel RGB arrays, tripling the input dimensionality and giving the model redundant color information to exploit. After fixing the data loader to use single-channel grayscale images, the validation accuracy and loss curves immediately began to track the training curves much more closely, reflecting a clear improvement in generalization.
* Yet we still encountered high jumps in the validations, and the training was still converging too quickly to high accuracy even when using only 5-10 epochs. Those results indicated that the model was still in overfitting. At that point we enlarged our dataset severely, and introduced more hyperparameters (learning rate, dropout rate, batch size, increased number of epochs) which significantly improved the training process.
* Our greatest challenge was scheduling meetings between us due to Alex being called to the reserve duty.

# Results

Augmentation clearly improved generalization, the best runs used it.

Learning rate around 5e-5 seems to be optimal for our task.

Batch size 64 worked better than 32.

Dropout 0.4-0.5 helped to prevent overfitting

| **Run #** | **Batch Size** | **Epochs** | **Learning Rate** | **Dropout** | **Augmentation** | **Test Accuracy** | **Precision** | **Recall** | **F1 Score** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 32 | 70 | 5e-5 | 0.3 | No | 0.8318 | 0.8175 | 0.8803 | 0.8477 |
| 2 | 32 | 50 | 5e-4 | 0.3 | No | 0.8682 | 0.8438 | 0.9231 | 0.8816 |
| 3 | 64 | 150 | 5e-5 | 0.5 | No | 0.8909 | 0.8550 | 0.9573 | 0.9032 |
| 4 | 64 | 100 | 5e-6 | 0.5 | Yes | 0.9091 | 0.9619 | 0.8632 | 0.9099 |
| 5 | 64 | 150 | 5e-6 | 0.5 | Yes | 0.9227 | 0.9808 | 0.8718 | 0.9231 |
| 6 | 64 | 150 | 5e-5 | 0.5 | Yes | 0.9364 | 0.9256 | 0.9573 | 0.9412 |
| 7 | 64 | 150 | 5e-6 | 0.45 | Yes | 0.9545 | 0.9652 | 0.9487 | 0.9569 |

**Worst results** came from small batch size (32), higher learning rate (5e-4), and no augmentation.

**Improvements** came when using a **larger batch size (64)**, **lower learning rates (5e-5 or 5e-6)**, **higher epochs (150)**, and **applying data augmentation**.

**Best results** (Run 7) came with:

Batch Size 64

Learning Rate 5e-6 very low but led to stable convergence.

150 Epochs was sufficient for training.

Dropout 0.45 proved ideal for regularization.

Data Augmentation likely helped to improve generalization especially for large epoch sizes.

Final Accuracy = **95.45%**

F1 Score = **95.69%**

# Conclusions

This study looked for new hyperparameter settings to improve how well a CNN can segment idiopathic pulmonary fibrosis. We tested different hyperparameters and saw that having a large, varied dataset is very important.

At first, our model couldn’t learn because the dataset was too small. When we added more images, the model began to overfit. We fixed this by converting images to grayscale, using data augmentation, and lowering the dropout rate so the model avoids learning too specific features.

Through step-by-step improvements in batch size, learning rate, dropout, training epochs, and enabling data augmentation, the model's generalization ability was significantly improved.  
The progressive changes highlight the importance of tuning hyperparameters and applying augmentation techniques in deep learning workflows, especially when working with moderately sized datasets (~4,400 images).

After these changes, our CNN can now tell if a lung CT scan is IPF-positive or IPF-negative with 93.64% accuracy.

# References

## Citations

1. Travis, W. D., King, T. E., Bateman, E. D., Lynch, D. A., Capron, F., Colby, T. V., ... & McLoud, T. C. (2002). American Thoracic Society/European Respiratory Society international multidisciplinary consensus classification of the idiopathic interstitial pneumonias. *American journal of respiratory and critical care medicine*, *165*(2), 277-304.‏
2. Wuyts, W. A., Cavazza, A., Rossi, G., Bonella, F., Sverzellati, N., & Spagnolo, P. (2014). Differential diagnosis of usual interstitial pneumonia: when is it truly idiopathic? European Respiratory Review, 23(133), 308-319.‏
3. NIfHaCE, N. I. C. E. *Idiopathic pulmonary fibrosis in adults: diagnosis and management*.‏
4. Raghu, G., Collard, H. R., Egan, J. J., Martinez, F. J., Behr, J., Brown, K. K., ... & Schunemann, H. J. (2011). An official ATS/ERS/JRS/ALAT statement: idiopathic pulmonary fibrosis: evidence-based guidelines for diagnosis and management. *American journal of respiratory and critical care medicine*, *183*(6), 788-824.
5. "Idiopathic Pulmonary Fibrosis". NHLBI, NIH. Retrieved 5 December 2020.
6. Raghu, G., Rochwerg, B., Zhang, Y., Garcia, C. A. C., Azuma, A., Behr, J., ... & Schünemann, H. J. (2015). An official ATS/ERS/JRS/ALAT clinical practice guideline: treatment of idiopathic pulmonary fibrosis. An update of the 2011 clinical practice guideline. *American journal of respiratory and critical care medicine*, *192*(2), e3-e19.‏
7. Meltzer, E. B., & Noble, P. W. (2008). Idiopathic pulmonary fibrosis. *Orphanet journal of rare diseases*, *3*, 1-15.‏
8. Ferri, F. F. (2017). *Ferri's clinical advisor 2018 E-Book: 5 Books in 1*. Elsevier Health Sciences.‏
9. Lynch, D. A., Sverzellati, N., Travis, W. D., Brown, K. K., Colby, T. V., Galvin, J. R., ... & Wells, A. U. (2018). Diagnostic criteria for idiopathic pulmonary fibrosis: a Fleischner Society White Paper. *The Lancet respiratory medicine*, *6*(2), 138-153.‏
10. Furukawa, T., Oyama, S., Yokota, H., Kondoh, Y., Kataoka, K., Johkoh, T., ... & Hasegawa, Y. (2022). A comprehensible machine learning tool to differentially diagnose idiopathic pulmonary fibrosis from other chronic interstitial lung diseases. *Respirology*, *27*(9), 739-746.‏
11. Hermena, S., & Young, M. (2023). CT-scan image production procedures. *StatPearls*.‏
12. Marhoon, Z. M. H., Abbas, S. B. Z. A. A., & Sobhi, T. S. A. (2024). Role of Multidetector Ct in Characterization of Renal Mass. *Current Clinical and Medical Education*, *2*(5), 314-339.‏
13. Luharia, A., Mishra, G., Haran, N., & Dhoble, S. J. (2022). Photodiagnostic techniques. In *Photophysics and Nanophysics in Therapeutics* (pp. 115-138). Elsevier.
14. Nadrljanski, M., & Foster, T. (2022). Computed tomography. *Reference article, Radiopaedia. org (pristupljeno 03. ožujka 2023.) https://doi. org/10.53347/rID-9027*.
15. Vajuhudeen, Z., Bell, D., Murphy, A., et al. (n.d.). *Sequential CT image acquisition*. Radiopaedia.org. Retrieved April 12, 2025.
16. ‏Morgan, M., Feger, J., Chen, W., et al. (n.d.). *Helical CT image acquisition*. Radiopaedia.org. Retrieved April 12, 2025.
17. Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)*, *39*(1), 1-22.‏
18. Rubin, D. B. (1987). The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association*, *82*(398), 543-546.‏
19. Plompen, A. J., Cabellos, O., de Saint Jean, C., Fleming, M., Algora, A., Angelone, M., ... & Žerovnik, G. (2020). The joint evaluated fission and fusion nuclear data library, JEFF-3.3. *The European Physical Journal A*, *56*, 1-108.‏
20. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.‏
21. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551.‏
22. Stanković, L., & Mandic, D. (2023). Convolutional neural networks demystified: A matched filtering perspective-based tutorial. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *53*(6), 3614-3628.
23. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1, No. 2). Cambridge: MIT press.‏
24. Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into deep learning*. Cambridge University Press.‏
25. Hinkelmann, K. (2018). Neural Networks, p. 7. *PDF). University of Applied Sciences Northwestern Switzerland*.‏
26. Brownlee, J. (2019). A gentle introduction to the rectified linear unit (ReLU). *Machine learning mastery*, *6*.‏
27. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1, No. 2). Cambridge: MIT press.‏
28. Lippert, C., Shahriar, M. H., Chowdhury, M. S. H., Junger, A., Rasul, M. G., & Yakub, M. Kuzushiji Classification.‏
29. Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, *2*(4), 303-314.‏
30. CS231n: Convolutional neural networks for visual recognition. (2016, May 10). *CS231n Convolutional Neural Networks for Visual Recognition*. Retrieved April 27, 2021.

1. <https://medium.com/@vishnuam/dropout-in-convolutional-neural-networks-cnn-422a4a17da41>
2. Venkatesh, D., Valarmathi, R., & Uma, R. (2022, May). An LSTM-based approach for predicting idiopathic pulmonary fibrosis progression. In AIP Conference Proceedings (Vol. 2464, No. 1). AIP Publishing.‏
3. Bennour, A., Aoun, N. B., Khalaf, O. I., Ghabban, F., Wong, W. K., & Algburi, S. (2024). Contribution to pulmonary diseases diagnostic from X-ray images using innovative deep learning models. *Heliyon*, *10*(9).‏
4. Bonelli, F. S., Hartman, T. E., Swensen, S. J., & Sherrick, A. (1998). Accuracy of high-resolution CT in diagnosing lung diseases. AJR. American journal of roentgenology, 170(6), 1507-1512.‏

1. <https://www.kaggle.com/competitions/osic-pulmonary-fibrosis-progression/data>

1. <https://www.kaggle.com/datasets/mehradaria/covid19-lung-ct-scans/data>

## Figures

1. <https://www.mypathologyreport.ca/diagnosis-library/usual-interstitial-pneumonia-lung/>
2. Azam, S., Montaha, S., Rafid, A. R. H., Karim, A., Jonkman, M., De Boer, F., ... & Chang, A. (2023). An automated broncho-arterial (ba) pair segmentation process and assessment of ba ratios in children with bronchiectasis using lung hrct scans: A pilot study. *Biomedicines*, *11*(7), 1874.‏

1. <https://global.canon/en/technology/ct2021s.html>
2. Loey, M., Manogaran, G., & Khalifa, N. E. M. (2020). A deep transfer learning model with classical data augmentation and CGAN to detect COVID-19 from chest CT radiography digital images. *Neural Computing and Applications*, 1-13.‏

1. <https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/>

1. <https://www.baeldung.com/cs/neural-networks-conv-fc-layers>
2. Brownlee, J. (2019). A gentle introduction to the rectified linear unit (ReLU). *Machine learning mastery*, *6*.‏

1. <https://botpenguin.com/glossary/softmax-function>

1. <https://en.wikipedia.org/wiki/Pooling_layer>

1. <https://blog.gopenai.com/understanding-autoencoders-part-iv-the-fully-connected-layer-and-flattening-566f246fa1a>
2. Da Wang, Y., Blunt, M. J., Armstrong, R. T., & Mostaghimi, P. (2021). Deep learning in pore scale imaging and modeling. *Earth-Science Reviews*, *215*, 103555.‏

1. <https://carpentries-incubator.github.io/intro-image-classification-cnn/instructor/04-fit-cnn.html>