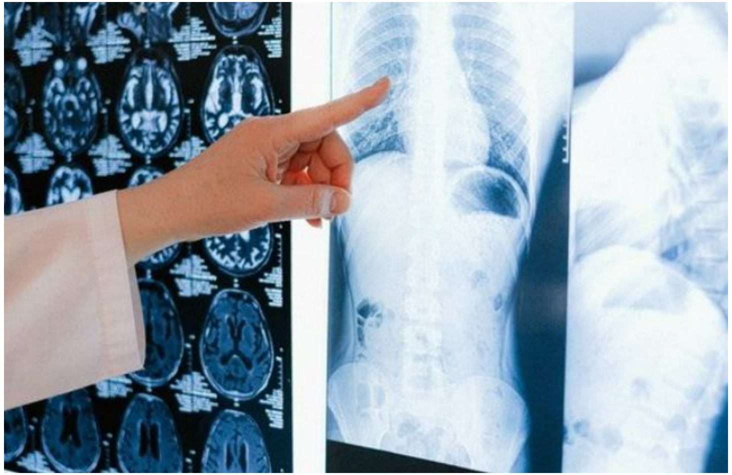
Pneumonia Detection and Comparison Using Different Deep Learning and Machine Learning Techniques

ECEN 5002-001 Deep Learning

How AI/ML/DL can help in disease diagnosis?

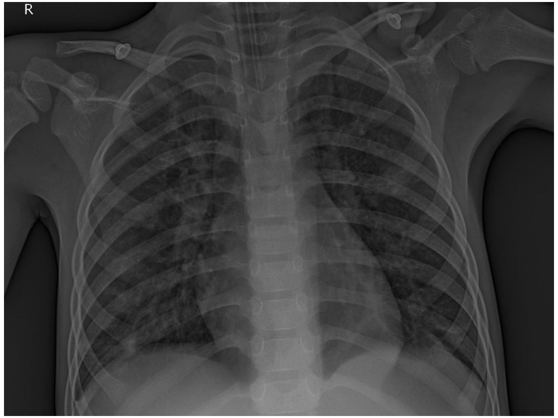
- Artificial intelligence can assist providers in a variety of patient care and intelligent health systems.
- Researchers use various AI-based techniques to detect the diseases such as skin, liver, heart, etc. that need to be diagnosed early.
- The Researchers have used techniques like Boltzmann machine, K nearest neighbour (kNN), support vector machine (SVM), decision tree, and artificial neural network to diagnose the diseases are presented along with their accuracies. [3]



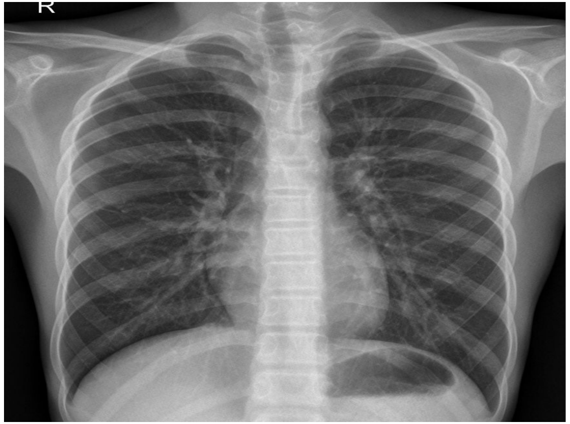
Motivation

- Pneumonia causes more than a million hospitalizations and more than 50,000 deaths each year. [4]
- With COVID-19 taking over the world in the last few years, a disease which aggressively attacks a person's respiratory system and can lead to severe pneumonia, it has become increasingly important to be able to diagnose cases of pneumonia as early as possible.

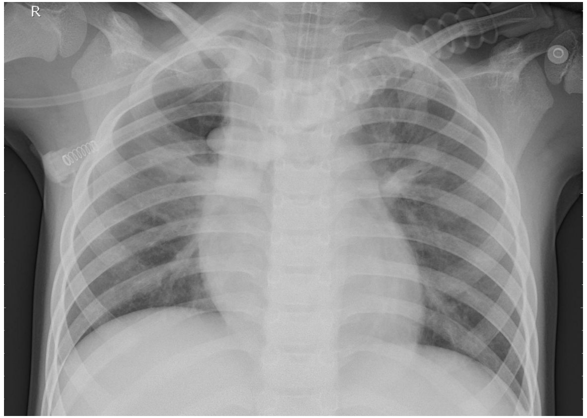
<u>How?</u>



PNEUMONIC LUNGS



NORMAL LUNGS



PNEUMONIC LUNGS

Dataset and Data Preprocessing

- We have used a dataset provided by Kaggle, which includes a collection of 5,232 labelled chest X-ray images.

Dataset:-

Pneumonic Images:- 3,883

Normal Images:- 1,349

Data was Preprocessed by doing the following:-

a) Resize
b) Rotate
c) Normalize
d) Imagefolder

Data was Preprocessed and fed to the system:-

a) Train
b) Test
c) Validation

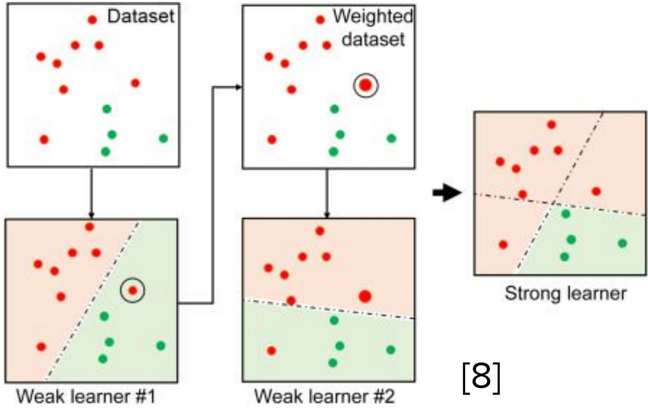
Techniques Used?

1. Boosting Algorithm
2. Fully Connected Algorithm
3. EfficientNet Algorithm

1. Boosting Algorithm

- Systematically combine results of several weak learners
- Algorithm choice: AdaBoost
    - Find p(xly) and p(y) from subset of training data based on assigned weight
    - Get weighted errors
    - Calculate performance
    - Update weights based on performance (increase for misclassified) and normalize
    - Iterate for given number of iterations
    - Combine results for final prediction

$$\alpha_j = \frac{1}{2}\ln\left(\frac{1-e_j}{e_j}\right)$$



[8]

- Weak Learners: Gaussian Naive Bayes, Multinomial Naive Bayes

$$p(y) = \frac{m_y}{m} \quad p(\mathbf{x}|y) = \prod_{i=1}^{d} p(x_i|y) \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma}^2) \quad p(x_i|y) = \frac{N_{yi} + \alpha}{N_y + \alpha m}$$

Results:- Boosting Algorithm

Result for 40x40 Images			
Iterations = 100	Accuracy(%)		
	Training	Validation	Test
Single Gaussian Naive Bayes	85	56	71
AdaBoost Gaussian Naive Bayes	68	56	72
Single Multinomial Naive Bayes	86	75	75
AdaBoost Multinomial Naive Bayes	91	81	79
Adaboost Decision Tree (scikit)	99	75	75

Results:- Boosting Algorithm

Result for 100x100 Images			
Iterations = 100	Accuracy(%)		
	Training	Validation	Test
Single Gaussian Naive Bayes	85	69	72
AdaBoost Gaussian Naive Bayes	67	69	71
Single Multinomial Naive Bayes	85	75	74
AdaBoost Multinomial Naive Bayes	90	81	76
Adaboost Decision Tree (scikit)	99	69	73

Observations

- Gaussian model is a poor guess
- Multinomial works relatively better
- Naive Bayesian classifier is a stable classifier with low variance and high bias [7]
- Decision tree is a classifier with high variance and low bias
- Decision tree is a better classifier than naive bayes (for boosting)
- The dimension of features had little impact
- AdaBoost immune to overfitting

2. Fully Connected Model

- Neural networks - a set of linear layers, with nonlinear activation functions interleaving the linear layers.
- Each function is a neuron (or a perceptron).
- A single neuron in a fully connected linear layer performs a dot product operation between a weight vector (w) and the output vector of the preceding nonlinear layer. [5]
- In the equation below,

$$y_{jk}(x) = f\left(\sum_{i=1}^{n_H} w_{jk}x_i + w_{j0}\right)$$

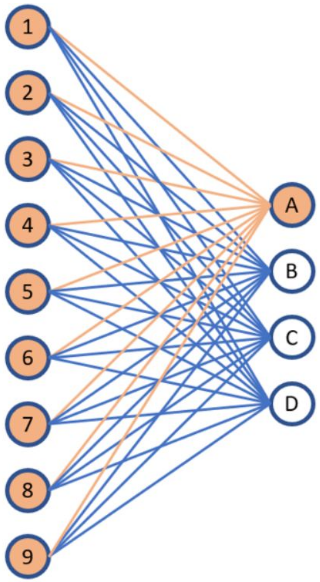


Fig:- An Example of FCL

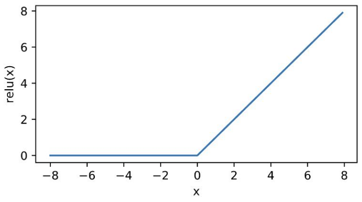


Fig:- ReLU Activation Function

2. Fully Connected Model

- We started with a basic model consisting of only 2 linear layers
- The optimizer used is Stochastic Gradient Descent
- The loss function used is Cross-Entropy Loss

(0)Linear

in_features=1024,
out_features=256,
bias=True

(1)ReLU

Activation Function.

(2)Linear

in_features=256,
out_features=2,
bias=True

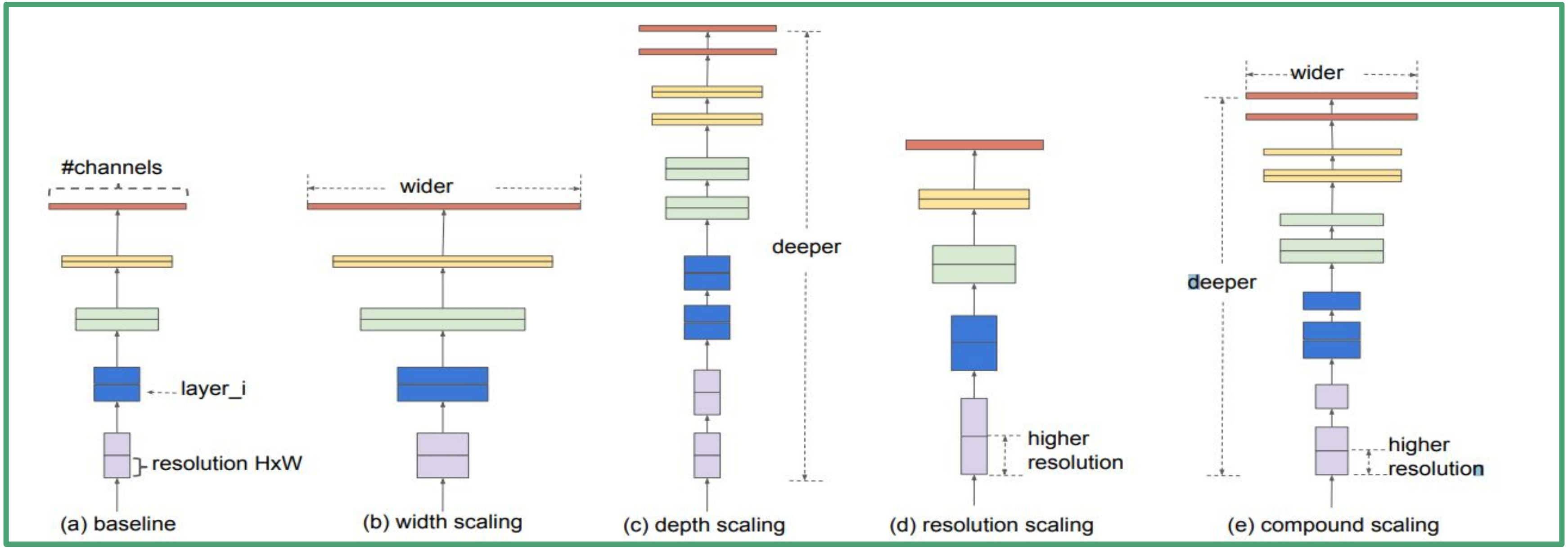
Results:- Fully Connected Layer Algorithm

Result for 32x32 Images		
Iterations = 1000		
	Loss	Test
Two Fully Connected Layers	0.46	76.2
Convolutional layers	0.66	65.2

3. State-Of-the-Art - EfficientNetV2

- State of the art model proposed by google AI
- Rethinks the way CNN scaled up
- Systematic ,principled scaling of depth,width and resolution
- Depth scaling- Increase layers
- Width scaling- Increase the feature maps
- Resolution Scaling - Increase the resolution of images to be fed
- Increase among single dimension- slow, less accurate, saturate
- Interdependent and systemic increase of all three.

- Compound scaling- Combination of width,depth and resolution
- α - Depth
  β - Width
  γ - Resolution
  ϕ -Scaling parameter



Training:-

- Training done in 2 stages
- Stage one- Fix ϕ =1
  - $\alpha.\beta^2.\gamma^2 \cong 2$
  - α>=1, β>=1, γ>=1 as the constraint ,obtain α,β and γ values for the simple model with gridsearch
  - α=1.2 ,β=1.1 ,γ= 1.15 for the V2 model, this is B0

- Stage 2- Change ϕ form 2-7
  - For every increase in the scaling parameter there is an increase in accuracy

- Progressive learning- Adjust regularization as image size increases

Results:- EfficientNetV2:-

Epoch	Training loss	Validation loss	Training accuracy	Validation accuracy
4	0.190376	0.824156	0.921396	0.562500
5	0.181656	0.558388	0.928298	0.687500
6	0.168729	0.526825	0.933857	0.687500

Results:- EfficientNetV2:-

	Epoch	Loss	Accuracy
Training	10	0.1601	0.9355
Validation	10	0.5000	0.7500
Testing	1	0.2873	0.8814

Comparison:-

	Parameters		
Model Name	Speed	Accuracy	Complexity
Boosting	25min	0.76	Medium
Fully Connected	15min	0.7612	Low
EfficientNet	2.5hours	0.8814	High

Conclusion:-

- Boosting Model - LOTs of feature engineering

- Fully Connected Layer Model- More simpler model worked well rather than a complicated convolutional layer model.

- EfficientNet Model- Most accurate but also most complex. State of the art.

Future expansion:-

- Testing boosting with other weak learners: K-nearest neighbors, logistic regression
- Weighted approach that is by adding more layers
- Pneumothorax and Tuberculosis detection
- Lung cancer detection

References:-

[1] Proceedings.mlr.press. 2022. [online] Available at: <http://proceedings.mlr.press/v139/tan21a/tan21a.pdf> [Accessed 28 April 2022].

[2] Kim, S., Rim, B., Choi, S., Lee, A., Min, S. and Hong, M., 2022. Deep Learning in Multi-Class Lung Diseases' Classification on Chest X-ray Images. *Diagnostics*, 12(4), p.915.

[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8754556/

[4][https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/five-facts-you-should-know#:~:text=Pneumonia%20is%20more%20common%20than,cause%20problems%20with%20oxygen%20exchange.

[5] https://towardsdatascience.com/convolutional-layers-vs-fully-connected-layers-364f05ab460b

[6] Chest X-Ray Images (Pneumonia) | Kaggle

[7] Ting, K.M. and Zheng, Z. (2003), A Study of AdaBoost with Naive Bayesian Classifiers: Weakness and Improvement. Computational Intelligence, 19: 186-200. https://doi.org/10.1111/1467-8640.00219

[8] Misra, Siddharth, Hao Li, and J. He. "Noninvasive fracture characterization based on the classification of sonic wave travel times." Machine Learning for Subsurface Characterization (2020): 243-287.

Thank you!