Multilabel chest X-ray classification using PCA+FCN and EfficientNeXtV2

Shitanshu Bhushan, Kuang-Yang Cheng, Rana Kamh, Andre Mixan
University of Michigan, Department of EECS
Github Link with Code : https://github.com/rana-rk1/pneumonias/tree/main

Abstract

Pneumonia detection presents a challenge as it requires expert radiologists or high computational cost. Here we propose using Principal Component Analysis and Fully Connected Networks to reduce computation time and increase the accuracy of current DenseNet 121 pneumonia detection models. Additionally, we develop a model that applies a variation of EfficientNetV2 for pneumonia detection to improve training accuracy. We use a subset of the ChestX-ray14 dataset to demonstrate how our proposed methods yield an increase in accuracy and time.

1. Introduction

Pneumonia is an inflammatory lung disease caused by infections of bacteria or viruses. According to the CDC, approximately 1.4 million people are diagnosed with Pneumonia in hospital emergency departments in the US every year, of which over 41 thousand die [1]. One of the best methods available to diagnose this disease is the analysis of chest X-rays, but this can be challenging as it heavily relies on the availability of expert radiologists.

Pneumonia detection using deep learning has been explored with models such as the CheXt-net [2] where a 121 layered DenseNet [3] is used to localize pneumonia found in chest X-rays. DenseNet121 connects each layer to one another to retain features over multiple layers. This method achieves a detection accuracy of 87% with a dataset composed of 112,120 images from ChestX-ray14 [4], complemented with labels of 14 diseases. While the DenseNet-121 model yields a high accuracy, the computational cost required to run this Convolutional Neural Network nessicitates a high capacity and a long training time to achieve high accuracy. To reduce runtime and computational cost associated with the DenseNet 121, we introduce the following:

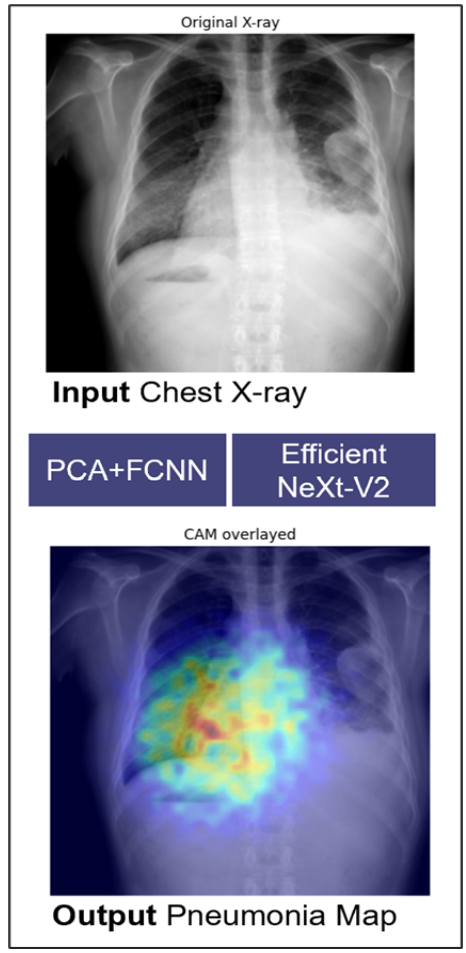

Fig 1 Input and output to our Efficient NeXt

1) We use Principal Component Analysis on our training images as input to a Fully Connected Neural Networks to significantly decrease training time of the model and improve the accuracy over a small dataset compared to the previously used Chext-Net
2) Our second approach uses a EfficientNet [5] and EfficientNetV2 [6] hybrid with a Gaussian error linear unit (GELU) as an activation function. This also improves accuracy compared to the original DenseNet 121 and allows us to run a large dataset in a significantly shorter time and less

computational power.

2. PCAF + FCNN

We developed a hybrid approach where we first reduce dimensionality using Principal Component Analysis (PCA) before feeding the data into a Fully Connected Neural Network (FCNN) to improve computing performance. To observe the impact on the results, we conducted an experiment using varying numbers of components and sample sizes.

2.1 Background and Model Architecture

Deep learning is a rapidly growing field of research, driven by the increase in computational power and advancements in model architecture in recent years. But complex models often take a long time to train. FCNNs are a fundamental type of neural network architecture, characterized by the way their neurons are connected. Due to their general structure, they can be applied to a wide range of problems, from basic classification tasks to more complex nonlinear problems. FCNNs have a straightforward and easily understandable architecture.

Training deep learning models typically requires a significant amount of time due to factors like model complexity and hardware limitations. Another way to enhance computing performance is by reducing the dimensionality of images. PCA is a widely used statistical technique in data science, machine learning, and statistics. It is used for dimensionality reduction while preserving as much variability in the data as possible.

2.2 Experiments and results

To enhance computing performance, we applied PCA to our datasets, as shown in figure 2. Next, we used DenseNet as a reference and employed a simple FCNN model to reduce model complexity. In our experiments, we use 100, 300, and 500 features with FCNN, which improves the runtime by nearly half compared to DenseNet (figure 3). However, only 300 and 500 features can still maintain high accuracy. And in order to investigate the relationship between data size and PCA, we chose to use 300 features for further experimentation. Figure 4. shows that the accuracy of PCA decreases more as the number of data images increases. It is important to note that as the amount of data

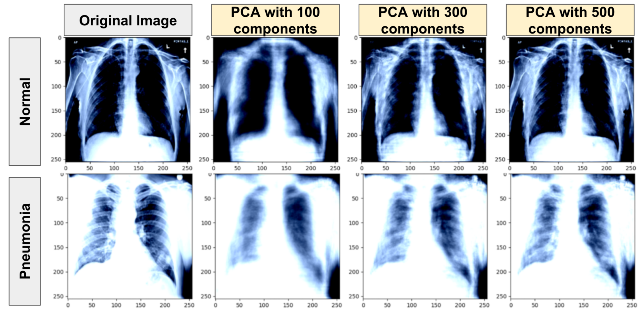

Fig 2 : PCA images appear slightly blurred but still retain the main components of the original images.

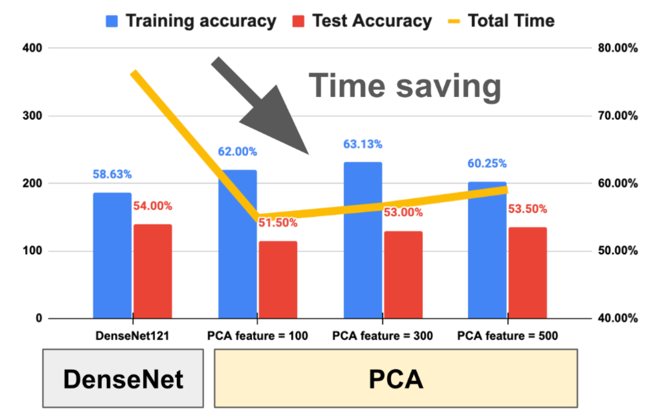

Fig 3 : PCA reduced the time by nearly half while maintaining the same accuracy.

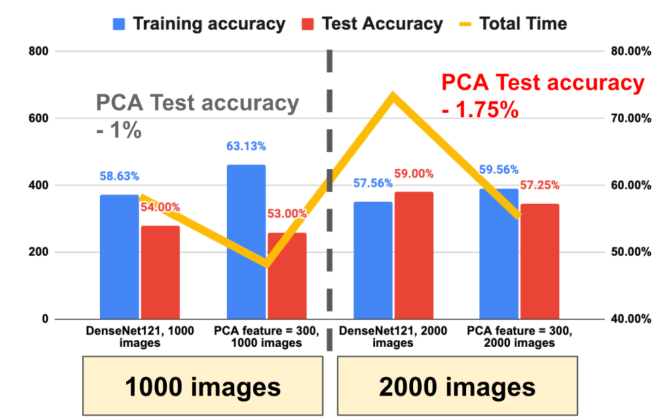

Fig 4 : The accuracy of PCA decreases more than DenseNet as the number of data images increases from 1000 to 2000.

increases, the difference in accuracy between PCA and DenseNet becomes larger. Maintaining accuracy on large data sets poses a challenge for PCA. Therefore, there is a trade-off between accuracy and time when considering these two approaches.

3. EfficientNeXtV2

We develop a new hybrid model called EfficientNeXtV2 derived by combining ideas from EfficientNet [5], EfficientNetV2 [6] & A ConvNet for the 2020s [7]. We started with a base implementation of EfficientNet-b3 and then experimented with techniques used by Zhuang, L. et al. (2022) [7] and saw the highest improvement using GELU as activation function over SiLU. Finally we incorporated the FusedMBConv block and new expand ratios used by Tan, M., & Le Q. (2021) [6]. The final model performs better than our base of EfficientNet-b3 while also taking less time to train.

3.1 Background

EfficientNets were introduced in 2019 where the authors presented a compound model scaling which uses a compound coefficient φ to uniformly scale network width, depth, and resolution in a principled way:

$$depth: d = \alpha^{\varphi}$$
$$width: w = \beta^{\varphi}$$
$$resolution: r = \gamma^{\varphi}$$
$$\alpha . \beta^2 . \gamma^2 \approx 2$$
$$\alpha \geq 1, \ \beta \geq 1, \ \gamma \geq 1$$

EfficientNetV2 was introduced in 2021 as a follow-up to the first EfficientNet, authors introduced new training routines and model changes to increase training speeds and have better parameter efficiency. Published in 2022, A *ConvNet for the 2020s* went over ways to improve the performance of ConvNets to match that of vision transformers.

3.2 Model Architecture

The model architecture resembles a hybrid between EfficientNet and EfficientNetV2 and uses GELU as an activation function.

Stage i	Operator $\hat{\mathcal{F}}_i$	Stride	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	1	40	1
2	FusedMBConv1, k3x3	2	24	2
3	FusedMBConv4, k3x3	2	32	3
4	FusedMBConv4, k3x3	2	48	3
5	MBConv6, k3x3	1	96	5
6	MBConv6, k5x5	1	136	5
7	MBConv6, k5x5	2	232	6
8	MBConv6, k3x3	1	384	2
9	Conv1x1 & Pooling	-	1536	1

Table 1 : EfficientNeXtV2 network architecture

The MBConv block is from MobileNetV2 [8] and they utilize depth wise separable convolutions and inverted residuals to efficiently capture features in a neural network. The MBConv block also has a squeeze-and-excitation [9] block which is basically used to adaptively scale channel wise features by calculating weights associated with each channel. The FusedMBConv block was introduced by EfficientNet-EdgeTPU [10] by google research and it replaces the expand conv1x1 and depthwise conv3x3 in MBConv with a regular conv3x3. The layers represent how many times each stage repeats.

3.3 Training and Results

We used SGD with momentum as our optimizer and as we have an imbalanced data ratio, we use Focal loss [11] as our loss function as it gives higher importance to hard-to-classify problems while training and reduces the impact of easy-to-classify data.
The images were resized and cropped to 300x300 and we used a batch size of 20 and trained on two different sizes of the NIH chest X-rays dataset [4].

Sample Size	Batch Size	Epoch	Accuracy
5606	20	24	56.557%
112120	20	6	54.156%

Table 2 : Accuracy of our model trained on NiH chest X-ray data. Sample Size is the number of images the model was trained on

We were able to achieve higher accuracy compared to a DenseNet121 and also had a quicker training time.

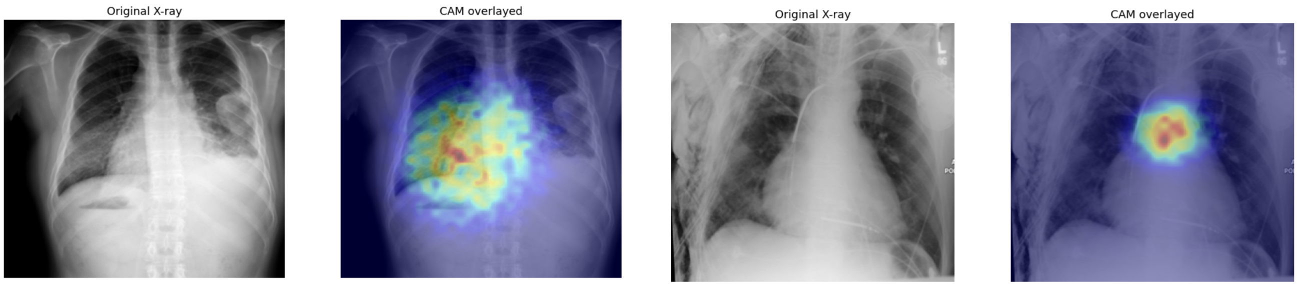

Fig 5 : EfficientNeXtV2 visualized using Class Activation Maps, which highlight the areas of the X-ray that are most important for making a particular pathology classification

4. Conclusion

The results we obtained show that existing deep learning models for the detection of Pneumonia in chest X-rays can be improved significantly through image pre-processing and tuning of the NN learning parameters, and this can be achieved with resources available at no cost. Our work using PCA and FCNN demonstrates a reduction in training time and improved accuracy over a small training dataset. Additionally, our EfficientNeXtV2 improves efficiency in comparison to the DenseNet121 model. However, our work can be further improved through more advanced image pre-processing and further training of the NN with a larger amount of epochs.

The use of PCA with FCNN is very promising in scenarios where the data is large and has very prominent features.

EfficientNeXtV2 could be trained to have similar accuracy to that of ViT [12] classifiers while having a much simpler training and deployment

5. References

[1] Centers for Disease Control and Prevention (2023). *Pneumonia*. Centers for Disease Control and Prevention. https://www.cdc.gov/nchs/fastats/pneumonia.htm

[2] Rajpurkar, P. et al. (2017). "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning." *arXiv preprint arXiv:1711.05225*.

[3] Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[4] Wang, X., et al. (2017). "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[5] Tan, M., & Le Q. (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks."*International conference on machine learning*. PMLR, 2019.

[6] Tan, M., & Le Q. (2021). "Efficientnetv2: Smaller models and faster training." *International conference on machine learning*. PMLR, 2021.

[7] Zhuang, L. et al. (2022). "A convnet for the 2020s." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

[8] Sandler, M., et al. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[9] Hu, J., Li S., & Gang S. (2017). "Squeeze-and-excitation networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[10] Gupta S., & Tan M. (2019). *EfficientNet-EdgeTPU: Creating Accelerator-Optimized Neural Networks with AutoML*. Google. https://blog.research.google/2019/08/efficientnet-edgetpu-creating.html

[11] Lin, T., et al. (2017). "Focal loss for dense object detection." *Proceedings of the IEEE international conference on computer vision*. 2017.

[12] Dosovitskiy, A., et al. (2020). "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020)