Classification of Pneumonia Using Deep Autoencoder for Chest X-ray Images

Karim Kousa[a], Silvia Tavares[b], Sónia Ferreira[c], and Francisco Cuevo[d]
[a] up202102687@fe.up.pt, [b] up202204392@fe.up.pt, [c] up202200967@fe.up.pt, and [d] up202302145@fe.up.pt

Abstract—This project aims to enhance pneumonia detection from chest X-rays using an encoder-decoder for unsupervised feature learning. We will also integrate different architectures, including Variational Autoencoders (VAE), Convolutional Neural Networks (CNN), and Transfer Learning models like ResNet, VGG16, and DenseNet, to bolster our methodology. The goal is to improve pneumonia classification accuracy and contribute to medical image analysis through advanced deep-learning techniques. We got promising results with DenseNet and ResNet models across various metrics. At the same time, AutoEncoder and VAE show promising results for projects where the goal is prioritizing high recall rates.

Keywords—Pneumonia Classification, Chest X-ray imaging, Encoder-decoder, Variational Autoencoders, Convolutional Neural Networks, VGG16, Resnet, DenseNet, Deep Learning in Healthcare.

1 Introduction

Pneumonia hospitalizes 1 million adults and kills over 50,000 in the US (CDC, 2017). It causes coughing, fever, and respiratory distress by inflaming one or both lungs. For effective treatment and better patient outcomes, pneumonia must be identified quickly. Unfortunately, radiographic findings do not always confirm pneumonia because it is one of many lung disorders. Given current technology, radiographic criteria cannot distinguish pneumonia from other lung diseases. Therefore, with current technology, it is impossible to distinguish pneumonia from other lung diseases with certainty using radiological criteria.

Producing large amounts of accurately labeled data for pneumonia detection systems is difficult. Labeling pneumonia data is difficult because radiologists are needed, and labeled images are scarce. Deep learning, a subfield of artificial intelligence, can diagnose pneumonia from chest X-rays.

Deep-learning algorithms can identify pneumonia-related patterns in large chest X-ray datasets. CNNs' deep-learning architecture makes them ideal for image identification. CNNs can detect lung infection or inflammation in chest X-rays by analyzing pixel texture, shape, and intensity.

Deep-learning models can classify fresh chest X-rays as pneumonia or not after training. Performing this task in real time may help doctors diagnose and treat pneumonia. Deep-learning models can also help radiologists analyze chest X-rays, improving patient outcomes and reducing errors.

2 Methods

2.1 Dataset

The dataset for our project was sourced from Kaggle [1] and comprises 5,863 JPEG X-ray images from pediatric patients, categorized into Pneumonia and Normal classes. These images are divided into training, testing, and validation sets, with the images neatly organized within corresponding folders. They are from Guangzhou Women and Children's Medical Center, of pediatric patients one to five years old. Before inclusion, the images underwent a rigorous quality control process to exclude any that were of low quality or unreadable.

Fig. 1: Examples of X-Rays in Patients with Pneumonia

Figure 1, illustrates an example of these X-ray images. The left panel shows a normal chest X-ray with clear lungs. The middle image reveals bacterial pneumonia, indicated by focal consolidation in the right upper lobe (white arrows). The right panel displays viral pneumonia, characterized by a diffuse 'interstitial' pattern across both lungs.

2.2 Modeling

We aim to refine pneumonia classification from chest X-rays using deep learning. We develop a binary classifier leveraging both an Encoder-Decoder, Variational Autoencoders and CNN architecture to distinguish between the presence and absence of pneumonia. Additionally, we integrate pre-trained robust models like ResNet, VGG16, and DenseNet to bolster our system.

The dataset images were transformed, normalized, and resized to a uniform dimension of 256x256 pixels to ensure that all images have the same size. Additionally, the validation set was redefined by strategically selecting samples from the training dataset, as during data exploration, it was observed that the provided validation dataset wasn't representative of the diverse range of characteristics. This step is essential for assessing the generalization capabilities of our model and ensuring its reliability in real-world medical applications.

2.3 Encoder Decoder

The implementation of Deep Autoencoders [2] is part of an unsupervised feature learning approach which is the main

focus of this project. The goal is that the encoder part of the autoencoder compresses the input X-ray images into a lower-dimensional latent space.

The encoder module uses Torchvision's pre-trained EfficientNet-B4 model. To preserve convolutional feature map output, fully connected layers, average pooling, and dropout are excluded. The dataset's hierarchical features are preserved in this encoder's output.

The encoder's compressed output is transformed to match the transpose convolutional layers' input format. The decoder uses Transpose Convolutional Layers to upsample the encoded input and reconstruct it. Starting with a linear layer to reconfigure encoded features, data is gradually upsampled using ConvTranspose2d layers to produce the output. A SELU activation function after each ConvTranspose2d layer adds non-linearity and aids learning. The last layer uses the Sigmoid activation function to rescale the output to 0–1, matching the input image's pixel intensity range.

The encoder's forward pass extracts features from input data using the EfficientNet-B4 model. The encoder compresses and adds elements to match the decoder's input dimensions. The decoder reconstructs the input image from encoded features using transpose convolutional layers. Output should match input data.

This architecture uses the encoder to create a hidden representation of the input data and the decoder to reconstruct it. The method uses transpose convolutions to increase resolution and decode encoded features, producing an image that matches the input.

Adam optimizer, cross-entropy loss function, 32-batch size, 0.001 learning rate, and 10 epochs were used for training.

2.4 Variational Autoencoder (VAE)

VAEs excel at generative tasks and data reconstruction. However, unlike 2.3, they focus on exploring and generating new samples from learned latent space. VAE uses stochasticity and is not deterministic.

The training and testing methods, process, parameters, and hyperparameters are similar to 2.3, with the exception that the loss function combines sigmoid activation and binary cross-entropy losses into one class. Due to resource constraints, the architecture is simpler. About 121 million parameters make up the model.

The pre-trained convolutional neural network EfficientNet-B4 encodes input images to extract features. Fully connected layers and classification components are removed. Compressing the result creates a feature vector.

The model has two linear layers, fc_{mean} and fc_{logvar}. These layers calculate the latent space mean and log-variance from the encoder's flattened output. Layers define latent space probability distribution parameters.

The encoder's mean and log variance were used to reparameterize the latent space and generate samples. Sampling from a normal distribution with a mean and standard deviation adds randomness to the network during training.

Latent space is mapped back to its encoded size using linear layers in the decoder. The dimensions are gradually increased until the output image size is reached. After each linear layer, ReLU activations introduce non-linear behavior. The output layer is a linear layer that uses a Sigmoid activation function to rescale output values to 0–1 for image reconstruction.

The forward function passes input through the encoder to calculate latent space average and logarithmic variance. Samples from latent space are generated using reparameterization. After sampling, the decoder uses the latent space vector to recreate the image. Output dimensions are adjusted to match image dimensions.

VAE training uses a loss function consisting of image reconstruction loss measures and the difference between the reconstructed and original input images. KL divergence Divergence ensures the acquired latent space follows a predetermined distribution, and the minimum loss optimization of model parameters is often done with an Adam optimizer.

This VAE architecture acquires a latent representation of input images and reconstructs them while promoting a well-organized latent space for novel and lifelike samples. Adding a stochastic component during training gives the model generative abilities and a wide range of outputs.

2.5 Convolutional Neural Network (CNN)

CNNs, specialized in processing and interpreting visual data, are inspired by the human visual system and excel in capturing local patterns in larger images, leading to superior performance [3] [4]. We developed a custom CNN architecture based on ConvNet principles, designed for efficient image-based feature extraction with reduced complexity [5]. Our architecture comprises multiple convolutional layers. Each convolutional layer will be systematically followed by a max-pooling layer, designed to reduce the spatial dimensions of the input volume while retaining the most critical feature information. The network will culminate in a fully connected layer tasked with the classification process. Notably, the CNN will process images of 256x256 pixels. This decision is based on the ability of CNNs to manage more effectively and extract relevant features from larger image dimensions, which is anticipated to enhance the model's performance in identifying intricate patterns within medical imagery.

Our deep learning model's configuration involves carefully selected hyperparameters for optimal performance. The training utilized the Adam optimizer, cross-entropy loss, a batch size of 32, a learning rate of 0.01, and 30 epochs with a patience of 5 for early stopping. This approach ensures efficient training and prevents overfitting by halting training based on validation performance.

2.6 VGG16

The VGG16 model, part of the VGGNet series, is a renowned deep CNN architecture known for its simplicity and effectiveness in image classification [5]. It consists of sixteen layers, including thirteen convolutional and three fully connected layers, characterized by its use of small 3x3 convolutional filters, which are uniformly stacked, contributing significantly to the network's depth and capacity to capture complex features in visual data.

Their pre-trained models on ImageNet are sought after for their extensive training on a vast array of images, enabling intricate feature extraction of low-level visual details critical in transfer learning applications.

Hence, we employed VGG16 for its pre-trained feature extraction capabilities, requiring minimal additional training. The training process involved the Adam optimizer with a cross-entropy loss function, a batch size of 32, and a learning rate of 0.001, with a momentum of 0.9 and weight decay of 0.01. The learning rate scheduler applied exponential decay with a step size of 10 epochs. The model trained for 30 epochs, with an early stopping patience of 5 to prevent overfitting and optimize training efficiency.

2.7 Resnet

ResNet, or Residual Network, is a deep neural network architecture specifically designed to overcome challenges associated with training deep networks [6]. The key innovation lies in using residual learning blocks, incorporating skip connections, to address issues like vanishing gradients and facilitate practical training of deep networks [6]. ResNet architectures, such as ResNet-18, ResNet-34, and ResNet-50, vary in depth, with the number indicating the layers in the network.

In this project, we implemented ResNet-50, which follows a hierarchical structure with convolutional blocks, residual blocks, and global pooling layers. In each residual block, a skip connection is created by adding the input to the output of the second convolutional layer. The network utilizes identity and projection connections to maintain dimensional consistency. After passing through residual blocks, global pooling layers reduce spatial dimensions, and a fully connected layer is connected for classification, having two nodes for binary classification ("Normal" and "Pneumonia").

The model is adapted for the specific task by modifying the fully connected layer, and the training phase involves adjusting the model on the training set for 30 epochs.

2.8 DenseNet

DenseNet, an abbreviation for Densely Connected Convolutional Networks, is a deep neural network architecture designed to tackle challenges associated with network depth [8]. Its distinctive feature lies in densely connected layers, promoting effective feature reuse and enhancing information flow throughout the network [8].

In DenseNet, each layer receives input not only from the previous layer but also from all preceding layers, forming dense connections by concatenating outputs [8]. This design facilitates efficient feature utilization, addressing the vanishing gradient problem. Each layer in DenseNet has direct access to the collective knowledge of all preceding layers, resulting in highly interconnected feature maps that capture intricate patterns and representations across various scales. The dense connections also create shorter paths for gradients during backpropagation, mitigating the vanishing gradient problem and enabling more effective training of deep networks.

In our project, we leverage the DenseNet model for binary image classification, precisely predicting between "Normal" and "Pneumonia" classes. The pre-trained DenseNet is fine-tuned for this task by adjusting the classifier layer. The training process involves stochastic gradient descent (SGD) optimization to minimize cross-entropy loss.

The training phase spans 10 epochs, with visualization of training and validation history. After training, the best-stored model is loaded and evaluated on the test set, displaying metrics like loss, accuracy, recall, precision, and the confusion matrix. Finally, predicted and actual labels are visualized in the test set.

2.9 Evaluation Metrics

In our project, we evaluated all models using the same set of metrics to ensure accuracy and reliability in pneumonia detection. It included: (a) Loss: to measure the prediction error; (b) Accuracy: to assess the percentage of correct predictions, giving us the model's effectiveness; (c) Precision: which evaluates the accuracy of the positive predictions; (d) Recall: that assesses the model's ability to detect all relevant cases; (e) F1 Score: balances precision and recall, offering a single metric for model performance where both metrics are crucial; (f) Confusion Matrix: to allows us to visualizes the model's performance.

3 Results and Discussion

In this section, we will succinctly analyze the performance outcomes of each model, highlighting key findings and interpreting their implications for our project's objectives. The results obtained for our models' training, validation, and testing are summarized in tables 1 and 2.

TABLE 1: Training and Validation parameters and results comparison per model

Models	Nr of	Params		Training		Validation	
	Epochs	Trainable	Non-trainable	Accuracy (%)	Loss	Accuracy (%)	Loss
AutoEncoder	10	1003605779	0	85.9	0.3327	87.1	0.3247
VAE	10	121242736	0	76.9	0.4835	79.8	0.4806
CNN	30	19578946	0	98.1	0.0520	97.9	0.0670
VGG16	30	134268738	0	95.2	0.1420	96.9	0.0890
ResNet	30	23512130	0	99.9	0.009	98.7	0.038
DenseNet	10	4022920	0	100	0.000	99.2	0.024

TABLE 2: Testing results comparison per model

Measures	AutoEncoder	VAE	CNN	VGG16	ResNet	DenseNet
Loss	0.4562	0.7136	1.418	0.836	0.6910	1.118
Accuracy (%)	81.7	65.2	72.1	72.9	81.4	85.1
Precision (%)	80.53	66.48	79.6	78.8	85.4	88.0
Recall (%)	93.33	89.49	72.1	72.9	81.4	85.1
F1 (%)	86.46	76.28	66.7	68.2	79.6	84.0

In the evaluation of our models and by comparing results, DenseNet and ResNet stood out, with DenseNet achieving a notable testing accuracy of 85.1%, with excellent precision and recall. Regardless, it displayed high accuracy in training, which might indicate overfitting. Resnet also performed well, obtaining a strong test accuracy of 81.4% and a good balance in precision and recall, demonstrating its dependability. The AutoEncoder, while showing a moderate drop in testing accuracy to 81.7%, excelled in recall (93.33%), making it very efficient in identifying relevant cases. The same can be said for VAE despite their lower overall accuracy and precision. However, despite high training accuracy, the CNN and VGG16 models experienced a significant performance drop in testing. This suggests that the models are experiencing overfitting in the data, meaning that they have become too tailored to the training data and can struggle to generalize to new, unseen data.

This leads us to the conclusion that, for model selection, models such as DenseNet and Resnet may be more adequate due to their balance performance across different metrics, and they should also minimize false positives. On the other hand, due to their high recall rates, AutoEncoder and VAE might be more suitable if the objective is to find as many cases as possible.

4 Conclusion and Future Work

Diagnosing pneumonia, or a respiratory infection characterized by inflammation of the lung tissue, is a critical task for healthcare professionals. Traditional diagnosis relies heavily on interpreting chest X-rays, which can be complex and subtle, requiring expertise and experience. Different doctors may interpret X-rays differently due to variations in experience, expertise, and subjective judgment. They often deal with a large volume of medical imaging data, making it challenging to analyze and interpret efficiently. Also, manual analysis and interpretation of medical images can be time-consuming, especially when doctors are dealing with many cases simultaneously.

With this project, the goal was to automatically analyze and interpret medical images, providing a faster and more consistent approach to detecting abnormalities in chest X-rays and we tried different approaches.

The AutoEncoder, despite a slight drop in testing accuracy to 81.7%, excelled in recall (93.33%), making it highly efficient in identifying relevant cases. The VAE, while having lower accuracy and precision, also performed well in recall. In contrast, despite high training accuracies, the CNN and VGG16 models suffered from overfitting, as evidenced by their reduced testing performance. DenseNet stood out with a high testing accuracy of 85.1%, showcasing excellent precision and recall. Still, it has high accuracy in training, which might indicate overfitting. ResNet emerged as the most effective, as it showed strong performance with a testing accuracy of 81.4% and balanced precision and recall, indicating its reliability. Therefore we can conclude that this analysis suggests that DenseNet and ResNet are more suitable for balanced performance across various metrics, especially in minimizing false positives. At the same time, AutoEncoder and VAE are preferable for projects prioritizing high recall rates.

In conclusion, our exploration of various deep-learning models highlights the trade-offs between precision, recall, and accuracy in medical image analysis. The results indicate that while models like DenseNet and ResNet offer balanced and robust performance, nevertheless the choice of model should ultimately be guided by the specific requirements of the medical imaging task, such as the preference for either minimizing false positives (precision) or maximizing the detection of relevant cases (recall). With this project, we hope to contribute valuable insights into applying deep learning in medical imaging, paving the way for more advanced, reliable, and efficient diagnostic tools.

It's important to note that these models should be used as decision support tools, and qualified healthcare professionals should always confirm the final diagnosis. Additionally, ethical considerations, data privacy, and model interpretability are essential aspects that must be carefully addressed in deploying deep learning models in healthcare settings.

Therefore for future work, we propose enhancing our AI models for diagnosing pneumonia from chest X-rays. This can include the enhancement of hyperparameters, using advanced deep learning architectures, or hybrid models that can combine strengths with the ones used in this project. Expanding, diversifying the dataset, and focusing on data augmentation should be another goal. Plans to make these models interpretable for healthcare professionals, conduct real-world clinical trials integrating them into healthcare IT systems, and try to adapt them for use in low-resource settings should also be considered. Another suggestion is to extend the technologies for detecting multiple lung diseases in compliance with ethical and privacy concerns. Compliance, including collaboration with medical experts, should be a goal to get more reliable results and improvements. Regardless, and finally, continuous monitoring of the model's long-term performance is essential for ensuring reliability and accuracy.

References

[1] P. Mooney, "Chest X-ray Images (Pneumonia)," Kaggle, 2018. [Online]. Available: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia. [Accessed: 15- 11- 2023].

[2] Nayeeb Rashid, Md Adnan Faisal Hossain, Mohammad Ali, Mumtahina Islam Sukanya, Tanvir Mahmud, Shaikh Anowarul Fattah, *AutoCovNet: Unsupervised feature learning using autoencoder and feature merging for detection of COVID-19 from chest X-ray images*, *Biocybernetics and Biomedical Engineering*, Volume 41, Issue 4, 2021, Pages 1685–1701, ISSN 0208-5216, https://doi.org/10.1016/j.bbe.2021.09.004. (https://www.sciencedirect.com/science/article/pii/S020852162100108X)

[3] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[4] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

[5] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Thesis authored by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, titled 'Deep Residual Learning for Image Recognition.

[7] PyTorch Documentation and Resources for Model Loading and Loss Function Definition.

[8] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).