A comparative study of various models to classify normal and pneumonia infected chest xray images

Prithwish Basu Roy, Anil Poonai, Ifedayo Olusanya

NYU Tandon School of Engineering
{pb2718, ap5254, iio2004}@nyu.edu

Abstract

Pneumonia often becomes a life threat to children under five years. In developing countries, the death rate of children dying from pneumonia is higher than the combined death rate of HIV/AIDS and measles. Thus, early detection of pneumonia is very critical. Chest X-ray images can provide a deep insight into pneumonia-affected lungs. We compare various deep learning models' performance on the popular chest X-ray dataset and suggest the best model to use to detect pneumonia. We show that YOLOv8 a sophisticated CNN gives the best performance with 94% accuracy. Architecturally lighter CNNs like ResNet can also be a good choice for resource constrained setup.

Introduction

Pneumonia is a serious respiratory infection that predominantly affects the lungs, making it a significant health threat, especially for young children. It is mainly caused by bacteria, viruses, or fungi, with the most common bacterial cause being Streptococcus pneumoniae. In young children, their immune systems are not fully developed, and hence, they are more vulnerable to severe infections. The disease shows symptoms like high fever, cough, and difficulty breathing, which can worsen quickly in children because of their smaller airways and less developed lungs. The impact of pneumonia on young children worldwide is alarming. It is still one of the top causes of death among children under the age of five, mainly in low and middle-income countries where access to healthcare and vaccinations may be limited. The fight against pneumonia in young children is both a medical challenge and a public health priority. The Chest X-Ray dataset (Kermany et al. 2018) is a popular and commonly used dataset to analyze the x-ray images of lungs affected by pneumonia. In (Kundu et al. 2021), the authors utilized advanced transfer learning to address the limited availability of data and created an ensemble consisting of three convolutional neural network models: GoogLeNet, ResNet-18, and DenseNet-121. For the ensemble, they implemented a weighted average method where the weights for the base models were assigned using an innovative technique. This approach combines the scores of four key evaluation metrics—precision, recall, f1-score, and area under the curve—into a weight vector. Traditionally in the literature, these weights were set experimentally, a method susceptible to inaccuracies. Their proposed methodology was tested on two publicly accessible pneumonia X-ray datasets provided by (Kermany et al. 2018) and the Radiological Society of North America (RSNA), using a five-fold cross-validation strategy. On the Kermany and RSNA datasets, their method achieved impressive accuracy rates of 98.81% and 86.85%, along with sensitivity rates of 98.80% and 87.02%, respectively. In their work (Saha et al. 2021), the authors introduced GraphCovidNet, a model based on the Graph Isomorphic Network (GIN) for detecting COVID-19 from CT scans and CXRs. This model exclusively processes data in graph form. Images are initially pre-processed into undirected graphs, focusing on edges rather than the entire image. GraphCovidNet was tested on four datasets: SARS-COV-2 Ct-Scan, COVID-CT, a combination of covid-chestxray-dataset and Chest X-Ray Images (Pneumonia) (Kermany et al. 2018), and CMSC-678-ML-Project. It achieved an outstanding 99% accuracy across these datasets and was 100% effective in the binary classification task of identifying COVID-19 scans. Although this work claimed of high accuracy, this paper currently stands retracted.

In this paper, we take the chest x-ray dataset (Kermany et al. 2018) and try to find the best model suitable for the image classification problem of normal and pneumonia infected lungs. Following are our contributions:

- We compare various models of CNNs like ResNet-18, AlexNet, DenseNet-18, VGG-16 and YOLOv8.
- We also use DCGAN, a Generative Adversierial Neural Network model for the analysis and show that it is not suitable for the classification.
- We observe that YOLOv8 performs best on the given data set with the highest accuracy of 93.4% and an average accuracy of 91.2% across 20 epochs.
- We also observe that models like ResNet-18 and VGG-16 which are less complex and architecture heavy than YOLOv8 also performs well with highest test accuracy going upto 91.83% for ResNet, and 89.1% for VGG-16.
- We make our code public on github (DevonARP 2023).

Background

The dataset (Kermany et al. 2018) is structured into three main folders: train, test, and validation. Each folder contains subfolders for two image categories: Pneumonia and Normal. The dataset consists of 5,863 X-Ray images in total, which were taken from pediatric patients aged between one and five years old at Guangzhou Women and Children's Medical Center. The chest X-ray images were obtained as part of the patients' routine clinical care, and only high-quality scans were used for the analysis. The diagnoses of the images were graded by two expert physicians and later evaluated by a third expert to ensure accuracy.

Identifying lungs infected with pneumonia from X-ray images is a medical image classification problem that can be solved using Convolutional Neural Networks(CNNs) (O'Shea and Nash 2015). These models are particularly effective for this task because they can automatically detect and learn features from images. CNNs can identify patterns, shapes, and textures that differentiate normal lungs from those affected by pneumonia, even when these differences might not be immediately apparent to the human eye. The layered architecture of CNNs enables them to learn hierarchical feature representations, with deeper layers learning more complex features specific to normal and pneumonia-affected lungs.

Medical imaging datasets may be limited due to privacy concerns and the difficulty of gathering labeled medical data. However, CNNs, particularly pre-trained models such as ResNet (He et al. 2015), VGG (Simonyan and Zisserman 2014), or Inception (Szegedy et al. 2014), can be fine-tuned on specific datasets. These models have already learned a rich set of features from large, diverse image datasets, which can be a powerful starting point for a specific task.

Advanced CNN models, such as U-Nets (Ronneberger, Fischer, and Brox 2015) or models with attention mechanisms(give examples), can not only classify images but also help to localize the affected areas. This can be particularly useful for medical professionals to understand the extent and location of pneumonia in the lungs.

CNNs work well with data augmentation techniques. In medical imaging, where the dataset might be imbalanced (e.g., more normal cases than pneumonia cases), augmentation techniques like rotation, scaling, or horizontal flipping can help enhance the dataset and prevent overfitting.

GAN's are typically used to gerenate artificial content but it does have two models involved, a generator and discriminator, where the discriminator has to differentiate between artificial and real content. We can use the discriminator of a complex GAN (Goodfellow et al. 2014) such as DCGAN (Radford, Metz, and Chintala 2015) to diagnose x-ray images for pneumonia.

Our in-depth analysis further delves into an advanced algorithm such as YOLO (Juan Terven 2023). Our methodology differs from the conventional application of YOLOv8 in real-time object detection, as we investigate its viability for image classification. While YOLOv8 is designed for dynamic object localization, our study focuses on its adaptability in recognizing patterns within static medical images. This unconventional exploration seeks to unveil the broader versatility and applicability of YOLOv8 in diverse computer vision tasks, providing insights into its latent capabilities and potential extensions beyond its initial design.

Models Used and Reasons to use the models

In this section we provide a brief description of the architectures of the models that we have used, followed by an analysis of the result obtained when we train these models on the chest x-ray dataset. We start with a Basic CNN model and then use complex models like ResNet18, VGG16, DenseNet-121, YOLOv8 etc.

A basic CNN implementation

The first convolutional layer takes a single-channel (grayscale) input (1 input channel) and applies 16 filters/kernels of size 3x3 with padding of 1. Padding is used to preserve the spatial dimensions of the input. The second convolutional layer takes the 16-channel output from the previous and applies 32 filters/kernels of size 3x3 with padding of 1. In the activation Layer ReLU (Rectified Linear Unit) activation function is used here. It introduces non-linearity into the model, allowing it to learn more complex patterns in the data. After this a max pooling layer with a 2x2 window and stride of 2. This reduces the spatial dimensions (height and width) of the output from the convolutional layers by half, effectively downsampling the feature maps and reducing the computation required for subsequent layers. this is followed by two fully connected networks. The first fully connected layer, which flattens the output of the last convolutional layer and transforms it into a vector of size 32 * 64 * 64. It then maps this vector to a 128-dimensional space. The second fully connected layer, which maps the 128-dimensional vector from the previous fully-connected network to a 2-dimensional output space. This is for a binary classification task. For the loss calculation we are using the Cross Entropy Loss. The Adam optimizer which is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments, is used with a learning rate of 0.01 and 0.001.

Analysis: On trying out this basic model with two different learning rate of 0.01 and 0.001 we achieved test accuracies of 73% and 76% respectively. We also had to use data augmentation for limited amount of training data. Figure. 1 shows how the average test accuracy improves with data augmentation(blue line in (n)) than the one without augnmentation(in (a) and green line in (b)).

AlexNet

AlexNet (Krizhevsky, Sutskever, and Hinton 2012) is a revolutionary convolutional neural network (CNN) that significantly advanced the field of deep learning, particularly in image recognition tasks. Its architecture, which was introduced in 2012, consists of five convolutional layers followed by three fully connected layers. Key features of AlexNet include the use of ReLU (Rectified Linear Unit) for the non-linear part, overlapping pooling to reduce the size of the network, and dropout to combat overfitting in the

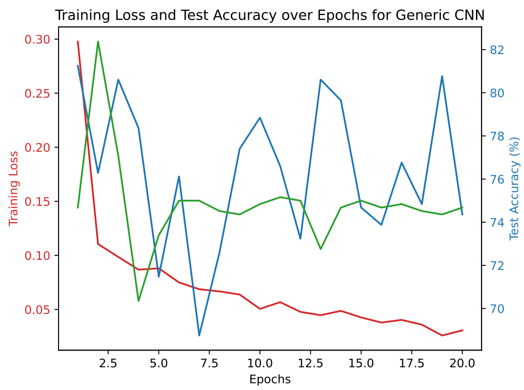


Figure 1: A basic CNN has an accuracy of around 75% on the chest X-ray dataset

fully connected layers. Additionally, AlexNet was one of the first CNNs to use GPU computing, which contributed to its performance and popularity. AlexNet's success in the ImageNet Large Scale Visual Recognition Challenge marked a breakthrough in the application of deep learning in computer vision.

Analysis In Figure 2, the training loss is relatively high and does not show significant improvement over the epochs, fluctuating around the 0.57 - 0.64 range. This suggests that the model is struggling to learn effectively from the training data or that it has reached a plateau in its learning capacity. Both validation and test accuracies are consistently low and stagnant across all epochs, with validation accuracy at a constant 50.0% and test accuracy at 62.5%. Such consistent scores, especially in validation accuracy, are indicative of certain underlying issues. This could imply that the model is performing no better than random guessing, especially if the validation set is balanced. This is a strong indicator of model ineffectiveness or overfitting to a particular class. Although higher than the validation accuracy the test accuracy score is also static across all epochs, which further suggests that the model is not learning or generalizing well. This observations might be the result of the following reasons: a) AlexNet might be too simplistic or not sufficiently capable of capturing the complexities within the chest X-ray dataset. (b) The consistent accuracy scores hint at potential overfitting (model not generalizing well) or underfitting (model not learning enough from the training data). (c) The dataset is imbalanced and affects the model's ability to learn effectively, reflected in poor validation and test performances.

ResNet

ResNet, which stands for Residual Network, is a type of Convolutional Neural Network (CNN) architecture that was developed to solve the challenge of training deep neural networks. The main innovation in ResNet is the introduction of *skip connections* or *residual connections,* which allow the input to a layer to be added to its output. This helps to prevent the vanishing gradient problem that can occur

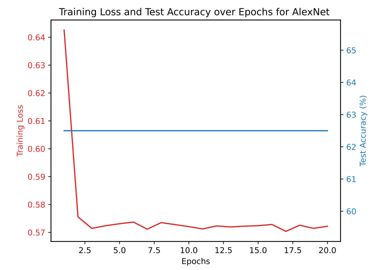


Figure 2: AlexNet Test Accuracy vs Training Loss

in deep networks. ResNet architectures come in different depths, such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152, with the number indicating the number of layers. They are made up of convolutional layers, residual blocks (each with a skip connection), batch normalization, and ReLU activations, followed by a global average pooling layer and a fully connected layer. ResNet's architecture enables it to learn effectively from a vast number of parameters and layers without experiencing significant performance degradation, making it highly efficient for various image classification and recognition tasks.

Analysis: Analyzing the training loss, validation, and test accuracies(Figure. 3) for ResNet-18 over 20 epochs on the chest X-ray dataset, we can draw several conclusions: The training loss steadily decreases from 0.1485 in epoch 1 to 0.0328 in epoch 20, indicating that the model is effectively learning and improving its performance on the training data over time.The validation accuracy fluctuates significantly across epochs, ranging from as low as 50.0% to as high as 100.0%. This wide variation suggests that the model's performance on the validation set is not consistent. Periods of 100% validation accuracy (epochs 2, 15, 16, 17) raise concerns about the representativeness of the validation set or potential overfitting, where the model might be too specifically attuned to the validation data. Test accuracy also varies across epochs, with a peak of 90.47% in epoch 2 and other highs and lows. This inconsistency might indicate that the model's generalization to new data is not stable across epochs. The highest test accuracy achieved is notably good, but the model does not consistently maintain this high performance level. The disparity between validation and test accuracies, especially during epochs with perfect validation accuracy, could indicate overfitting. The model might be performing exceptionally well on the validation set but not as well on the unseen test set. ResNet-18, known for its residual connections that help avoid vanishing gradient problems, shows its effectiveness in learning from the training data, as evidenced by the decreasing training loss. However, the fluctuation in validation and test accuracies suggests that while the model is learning, its ability to generalize might be varying.

VGG-16

VGG-16 and VGG-19 are convolutional neural network models known for their simplicity and depth. They were

Figure 3: ResNet Test Accuracy vs Training Loss

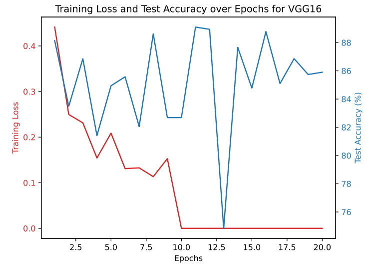


Figure 4: VGG-16 Test Accuracy vs Training Loss

key in advancing the understanding of deep CNN architectures. Both consist of multiple convolutional layers, followed by pooling layers, and finish with fully connected layers. The numbers 16 and 19 represent the total count of layers with weights, including both convolutional and fully connected layers. VGG models use small convolution filters (3x3) with a stride of 1 and always use the same padding, alongside max-pooling layers. This consistent architecture allows them to go deeper without making the network overly complex. They are widely used as baseline models for a variety of image processing tasks.

Analysis: In Figure. 4, the training loss shows a general decreasing trend from 0.4414 in the first epoch to 0.0876 in the final epoch. This trend indicates effective learning and adaptation of the model to the training data over time. The validation accuracy is consistently high, reaching 100% in several epochs. While this might initially seem positive, consistently perfect or near-perfect validation accuracies could indicate that the validation set might not be challenging enough or might not be representative of the broader dataset complexity. The consistency of 100% validation accuracy raises a concern about overfitting, where the model is too specifically tuned to the validation set. Test accuracy shows variability across epochs, with a high of 89.10% and lows around 74.84%. Unlike validation accuracy, test accuracy does not consistently hit the 100% mark, suggesting that the test set may present more variation or complexity than the validation set. The fluctuation in test accuracy also indicates variability in the model's ability to generalize. The highest test accuracies are achieved in epochs 11 and 16, which could be considered optimal points in training. The substantial gap between validation and test accuracies, especially in epochs where validation accuracy is 100%, suggests a potential overfitting to the validation set. VGG16, known for its depth and capability to learn complex features, seems to be effective in learning from the training data, as indicated by the decreasing training loss. However, the high validation accuracies and lower test accuracies highlight the need for careful consideration of the dataset's representativeness and model tuning to avoid overfitting.

DenseNet

DenseNet-121 (Huang et al. 2016), a member of the DenseNet (Densely Connected Convolutional Networks) family, is a notable CNN architecture distinguished for its dense connectivity pattern. Unlike traditional CNNs, DenseNet connects each layer to every other layer in a feed-forward fashion. This structure ensures that each layer receives feature maps from all preceding layers, fostering improved feature reuse and reducing the vanishing gradient problem. The architecture consists of 121 layers, including four dense blocks and transition layers. Dense blocks comprise Batch Normalization, ReLU activation, and 3x3 Convolution operations, where each layer's output is concatenated with its input, incrementally increasing the feature map depth. Transition layers, featuring BN, 1x1 Convolution, and 2x2 Average Pooling, follow each dense block (except the last), helping to compress and reduce the feature map dimensions. DenseNet-121 employs bottleneck layers before each 3x3 convolution to enhance computational efficiency by reducing the number of input feature maps. The network's growth rate, typically 32, signifies the number of features added by each layer to the network. The architecture concludes with global average pooling and a classification layer, mapping the features to the desired output classes. The design of DenseNet-121 results in remarkable parameter efficiency and robust feature propagation, making it highly effective for image classification tasks, especially where model efficiency is paramount.

Analysis: The training loss consistently decreases from 0.1485 in epoch 1 to 0.0328 in epoch 20. This trend indicates that the model is learning effectively from the training data across epochs. Validation accuracy shows significant fluctuation throughout the epochs, ranging from as low as 50.0% to as high as 100.0%. Such fluctuations could suggest varying degrees of model generalization on the validation set from epoch to epoch. The 100.0% validation accuracy at certain epochs (2, 15, 16, 17) is notably high, which might indicate that the validation set is not challenging enough or not representative of the broader dataset. Test accuracy also fluctuates but generally follows an increasing trend, starting from 70.31% in the first epoch and reaching up to 90.47% in the second epoch, then varying and ending at 84.22% in the final epoch. The highest test accuracy (90.47% in epoch 2) suggests that the model can potentially achieve high performance, but the inconsistency across epochs indicates variability in how well the model generalizes to new data. Periods where the validation accuracy is significantly higher than the test accuracy (e.g., epoch 2) raise concerns about overfitting, where the model may be too tuned to the validation set. The discrepancies between validation and test accuracies and their fluctuations suggest that the model's ability to generalize may not be stable across epochs. DenseNet-121's architecture, known for its efficiency in parameter usage and feature reuse, seems to be learning effectively, as evidenced by the decreasing training loss. However, the model's performance on external data (validation and test sets) is less consistent.

DCGAN

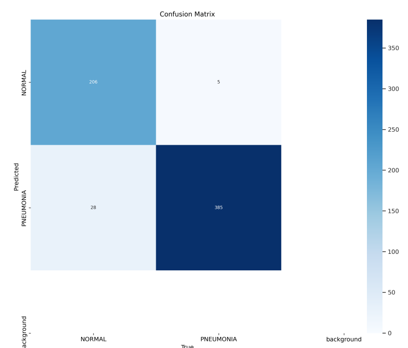
DCGAN is a generative adversarial network architecture (GAN). It uses two models a generator and discriminator, the generator is used for generating content, while the discriminator is used to determine how well it can pass for actual content. Some key factors involved with this model include replacing all of the max pooling layers with a convulational stride, usampling is dine by transposed convolutions, no fully connected layers, requires batch normalization for each layer except the output layer for the generator and input layer for the discriminator, ReLU function is used in the every layer of the generator except for the final layer where tanh is used, and LeakyReLU is used in the discriminator. Due to these differences especially the use of convulational layers instead of fully connected layers, DCGAN can capture spatial dependencies more accurately than a generic GAN. There are actually specific parameters for the weights that need to be given for both parts of the model, they were not used in these scenarios possibly leading to the below expected accuracy percentage. This is combined with the use of strided convolutions, with less of a use on pooled functions, leading to more of a binary classification with less granular focus on data as opposed to a regular CNN. The poor performance of the model was surprising given the complicated architecture and use case of the model, it seems that it isn't meant for as granular of a task as this is, but it does shine in identifying real data from artificial data.

Analysis DCGAN (Average: 37.66%, Highest: 37.98%), primarily used for generative tasks, performs poorly in this classification context. Its architecture is not suited for classification tasks, explaining the low accuracy.

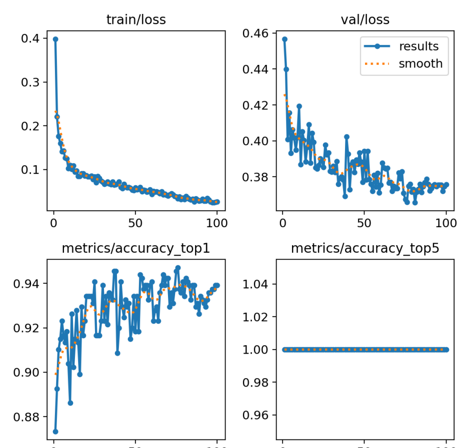
YOLOv8

YOLOv8 (Juan Terven 2023), a significant advancement in the YOLO series, showcases a sophisticated convolutional neural network architecture that integrates a modified CSPDarknet53 (Wang et al. 2020) backbone and a multi-layered head. The CSPDarknet53 backbone, consisting of 53 convolutional layers with cross-stage partial connections, enhances the information flow between layers. In the head, composed of both convolutional and fully connected layers, YOLOv8 adeptly predicts bounding boxes, objectness scores, and class probabilities. A distinctive feature of YOLOv8 is the inclusion of a self-attention mechanism in the head, allowing for dynamic focus on relevant image areas. This innovation, coupled with the model's proficiency in multi-scaled object detection through a feature pyramid network, enables precise detection of objects of varying sizes within an image through multiple layers. Together, these architectural enhancements contribute to YOLOv8's robust performance in real-time object detection scenarios, firmly establishing it as a significant breakthrough in computer vision applications.

Analysis: The analysis of YOLOv8 model(Figure. 5) reveals a commendable progression in performance across training epochs. The accuracy of the model has notably increased from 87% to 94%, indicative of improved classification precision. Moreover, the training loss has shown a substantial decrease from 0.3985 in epoch 1 to 0.0263 in epoch 100, reflecting effective learning and optimization of model parameters. Concurrently, the validation loss exhibited a consistent decline from 0.457 to 0.376 over the same epochs, suggesting enhanced generalization capabilities on unseen data. These collective outcomes underscore the success of the YOLOv8 model, demonstrating its capacity to learn, generalize, and achieve high accuracy in object detection tasks.



(a) Confusion Matrix for YOLOv8



(b) Training loss vs epochs(top-left), Validation loss vs epochs(top-right), test accuracy top1(bottom left) and top5(bottom right)

Figure 5: YOLOv8 achieves an accuracy of 94% on the ChestXray dataset

	Basic CNN	AlexNet	ResNet-18	DenseNet121	VGG16	DCGAN	YOLOv8
Highest Test Accuracy(%)	81.25	62.5	91.83	90.468	89.1	37.98	**93.429**
Average Test Accuracy(%)	76.35	62.5	81.86	81.17	85.20	37.66	**91.20**

Table 1: Comparing all the models

Results

In the table 1 we can see the Basic CNN (Average: 76.35%, Highest: 81.25%) performs reasonably well, suggesting that the basic architecture is sufficient for capturing significant features in the dataset. However, its performance is outclassed by more advanced architectures. AlexNet (Average: 62.5%, Highest: 62.5%) shows the lowest performance among the CNN architectures. This might be due to its relatively older and simpler architecture, which might not capture complex features as effectively as more modern architectures. ResNet-18(Average: 81.86%, Highest: 91.83%) demonstrates strong performance, likely benefiting from its residual connections that help in training deeper networks more effectively. The high peak accuracy indicates its potential in learning complex features. DenseNet-121 (Average: 81.17%, Highest: 90.468%) also performs well, with its densely connected layers likely contributing to effective feature reuse and reduced vanishing gradient issues. Its performance is comparable to ResNet-18. VGG16 (Average: 85.20%, Highest: 89.1%) shows the highest average accuracy. This model's architecture, known for its depth and use of small convolution filters, seems to be very effective for this dataset. However, its peak performance is slightly lower than ResNet-18 and DenseNet-121. YOLOv8 (Average: 91.20%, Highest: 93.429%), known for object detection, surprisingly shows the highest performance. However, YOLO's primary use case is object detection rather than classification. Its high performance here might be due to its robust feature extraction capabilities, but it's overkill for a straightforward classification task and may not be the most efficient choice.

Best Overall Performance: YOLOv8, although it's an unconventional choice for a pure classification task and might be more complex than necessary. Best Suited for Classification: ResNet-18 and DenseNet-121 are strong contenders. They both show high peak accuracies and are designed specifically for classification tasks. Efficiency and Complexity: If computational efficiency is a concern, ResNet-18 might be preferable as it generally has fewer parameters than DenseNet-121. Given its highest average accuracy, VGG16 is worth considering, especially if the model's higher computational and memory requirements are not an issue.

Conclusion

In conclusion, the choice of the best model depends on your specific requirements and constraints, such as computational resources, inference time, and model complexity. For pure classification tasks, ResNet-18, DenseNet-121, and VGG16 are strong choices, with ResNet-18 and DenseNet-121 slightly edging out due to their architectural advantages.

References

DevonARP. 2023. X-Ray Prediction. https://github.com/DevonARP/X-Ray-Prediction.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Networks. Cite arxiv:1406.2661.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. Cite arxiv:1512.03385Comment: Tech report.

Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2016. Densely Connected Convolutional Networks. Cite arxiv:1608.06993Comment: CVPR 2017.

Juan Terven, D. C.-E. 2023. A Comprehensive Review of YOLO: From YOLOv1 and Beyond.

Kermany, D. S.; Goldbaum, M.; Cai, W.; Valentim, C. C.; Liang, H.; Baxter, S. L.; McKeown, A.; Yang, G.; Wu, X.; Yan, F.; Dong, J.; Prasadha, M. K.; Pei, J.; Ting, M. Y.; Zhu, J.; Li, C.; Hewett, S.; Dong, J.; Ziyar, I.; Shi, A.; Zhang, R.; Zheng, L.; Hou, R.; Shi, W.; Fu, X.; Duan, Y.; Huu, V. A.; Wen, C.; Zhang, E. D.; Zhang, C. L.; Li, O.; Wang, X.; Singer, M. A.; Sun, X.; Xu, J.; Tafreshi, A.; Lewis, M. A.; Xia, H.; and Zhang, K. 2018. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5): 1122–1131.e9.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Kundu, R.; Das, R.; Geem, Z. W.; Han, G.-T.; and Sarkar, R. 2021. Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PLOS ONE*, 16(9): 1–29.

O'Shea, K.; and Nash, R. 2015. An Introduction to Convolutional Neural Networks. *ArXiv e-prints*.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. Cite arxiv:1511.06434Comment: Under review as a conference paper at ICLR 2016.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. Cite arxiv:1505.04597Comment: conditionally accepted at MICCAI 2015.

Saha, P.; Mukherjee, D.; Singh, P. K.; Ahmadian, A.; Ferrara, M.; and Sarkar, R. 2021. GraphCovidNet: A graph neural network based model for detecting COVID-19 from CT scans and X-rays of chest.

Simonyan, K.; and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. Cite arxiv:1409.1556.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2014. Going Deeper with Convolutions. Cite arxiv:1409.4842.

Wang, C.-Y.; Liao, H.-Y. M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; and Yeh, I.-H. 2020. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In *CVPR Workshops*, 1571–1580. IEEE. ISBN 978-1-7281-9360-1.