



**Submitted By**

**Hardik Mohanty-Roll No.2206024**

**Ayush Swain-Roll No.2229190**

**Akhilesh Das-Roll No.2229007**

**Ayush Mohanty-Roll No.-2129063**

**Rohit Kumar Yadurudra -Roll No.2206045**

**Submitted To**

**Faculty Name:- Dr. Abhaya Kumar Sahoo Sir**

**(Faculty of School of Computer Engineering, KIIT University)**

# TABLE OF CONTENTS

S. No.	Content	Page No.
1	OUTLINE OF THE PROJECT	2-4
2	CONFUSION MATRIX	5-6
3	TYPE OF ARCHITECTURE AND EXPLANATION	8-19
3-1	SIMPLE CNN AND RESNET	8-11
3-2	DENSE NET	12-15
3-3	VGG ARCHITECTURE	16-18
3-4	INCEPTION ARCHITECTURE	19
4	OUTPUT(ENSEMBLE ROC, PRECISION AND RECALL, ACCURACY ETC.)	20-30
5	CONCLUSION	31-33
6	REFERENCE	34

# OUTLINE OF THIS PROJECT

This project aims to build a deep learning-based system capable of differentiating between normal chest X-rays and those showing signs of pneumonia. To accomplish this, five CNN architectures—**SimpleCNN**, **ResNet-like**, **DenseNet-like**, **VGG-like**, and **Inception-like**—are designed and trained separately on the dataset. The results from each model are then compared to determine which architecture delivers the most accurate and reliable performance for pneumonia detection and analysis.

# WHAT IS PNEUMONIA AND HOW THIS PROJECT NECESSARY FOR IT?

- **Pneumonia** is a type of illness that is usually caused by microorganisms such as **bacteria, fungi, or viruses**, which lead to inflammation of the **air sacs** in one or both sides of the lungs and can result in severe illness or even death.
- This project is useful for the early detection of the spread of pneumonia disease through the use of **Deep Learning Architecture**, so that treatment can be provided more efficiently beforehand.

# HOW THIS PROJECT WORKS?

**Our project employs five Convolutional Neural Network (CNN) architectures—SimpleCNN, ResNet-like, DenseNet-like, VGG-like, and Inception-like—each offering unique structural advantages for image classification tasks. The dataset consists of chest X-ray images from both healthy individuals and patients affected by pneumonia. These images are used to train the models, enabling them to learn distinguishing features between normal and infected lungs. To evaluate performance, a Confusion Matrix is utilized, which helps categorize the results into True Positives, False Negatives, False Positives, and True Negatives, providing a comprehensive assessment of each model's accuracy and reliability.**



# WHAT IS A CONFUSION MATRIX?

A **confusion matrix** is a table which is used in Machine Learning and Deep Learning which is used to evaluate the performance of a classification model such as Biomedical use, Natural Language Processing etc. where output shows how many predictions are true and false based on 4 outputs:- **True Positive**, **True Negative**, **False Negative** and **False Positive** by using the existing dataset. **Final accuracy** is calculated by total no. of true outcome divided by total outcome.

		PREDICTED	
		Positive	Negative
ACTUAL	Positive	TRUE POSITIVE	FALSE NEGATIVE
	Negative	FALSE POSITIVE	TRUE NEGATIVE

dataaspirant.com

# CALCULATION FOR EACH ATTRIBUTE

- **Accuracy**=How correct is the model

$$A = (TP + TN) / (TP + TN + FP + FN)$$

- **Precision**=How many predicted cases are actually pneumonia

$$P = TP / (TP + FP)$$

- **Recall/Sensitivity**=How accurately model catches pneumonia

$$R = TP / (TP + FN)$$

- **F1-Score**=It is balance between Precision and Recall

$$F1 = 2 * (P * R) / (P + R)$$

# TYPES OF ARCHITECTURES TRAINED ON

- SimpleCNN Model
- ResNetLike Model
- DenseNetLike Model
- VGGLike Model
- InceptionLike Model



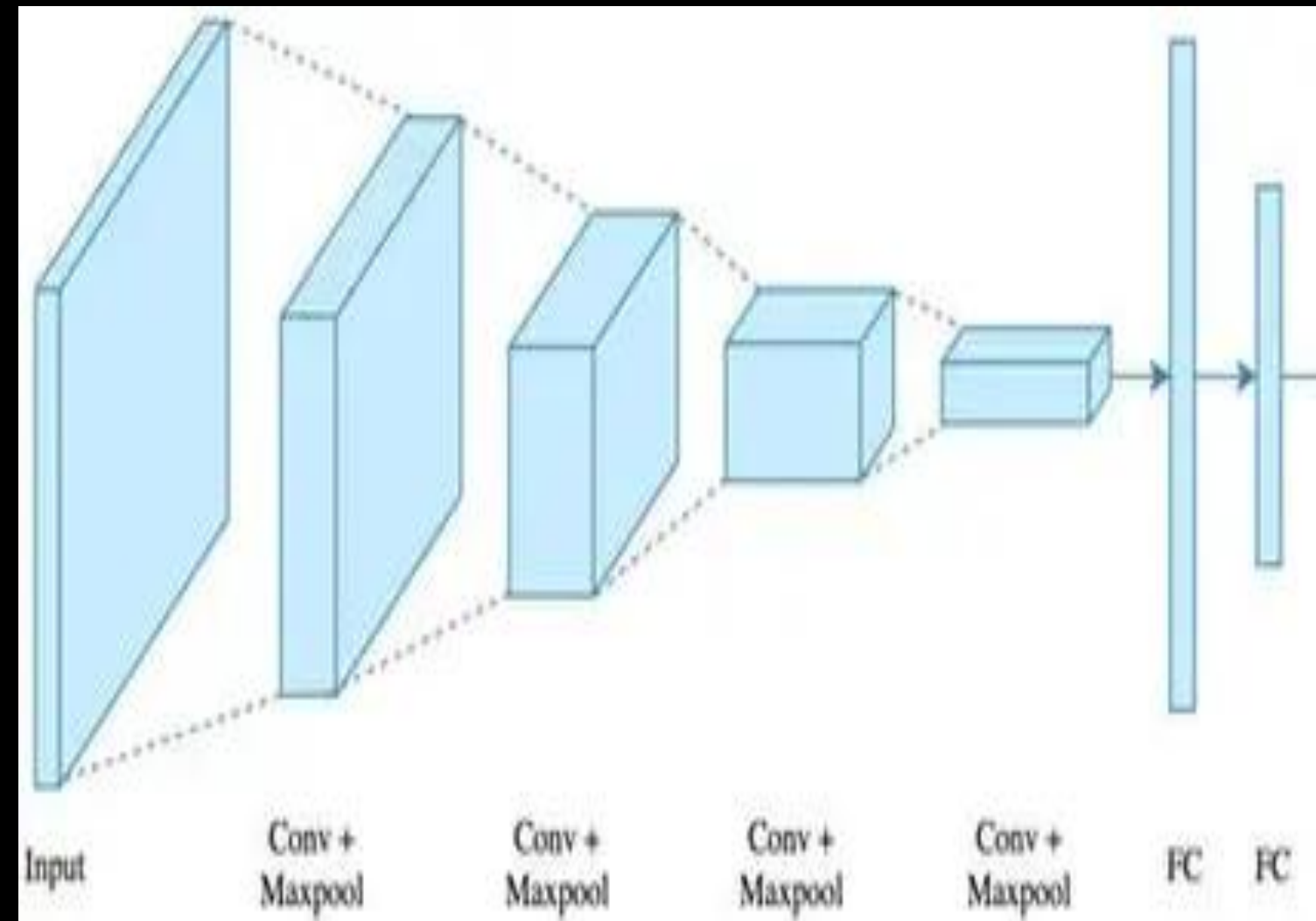


# SIMPLECNN ARCHITECTURE

The **SimpleCNN** model uses multiple convolutional layers with **ReLU** activation and sequential **Conv-ReLU-Pooling** layers followed by dense layers.

It uses smaller kernels (3\*3) and "same" padding to preserve spatial size, extracting patterns like :- edges and *textures* from x- rays The model ends with a sigmoid activation for binary output, balancing speed and accuracy.

Sigmoid activation producing probabilities between 0 and 1.



$$\text{Output} = \text{sigma}(Wx + b)$$

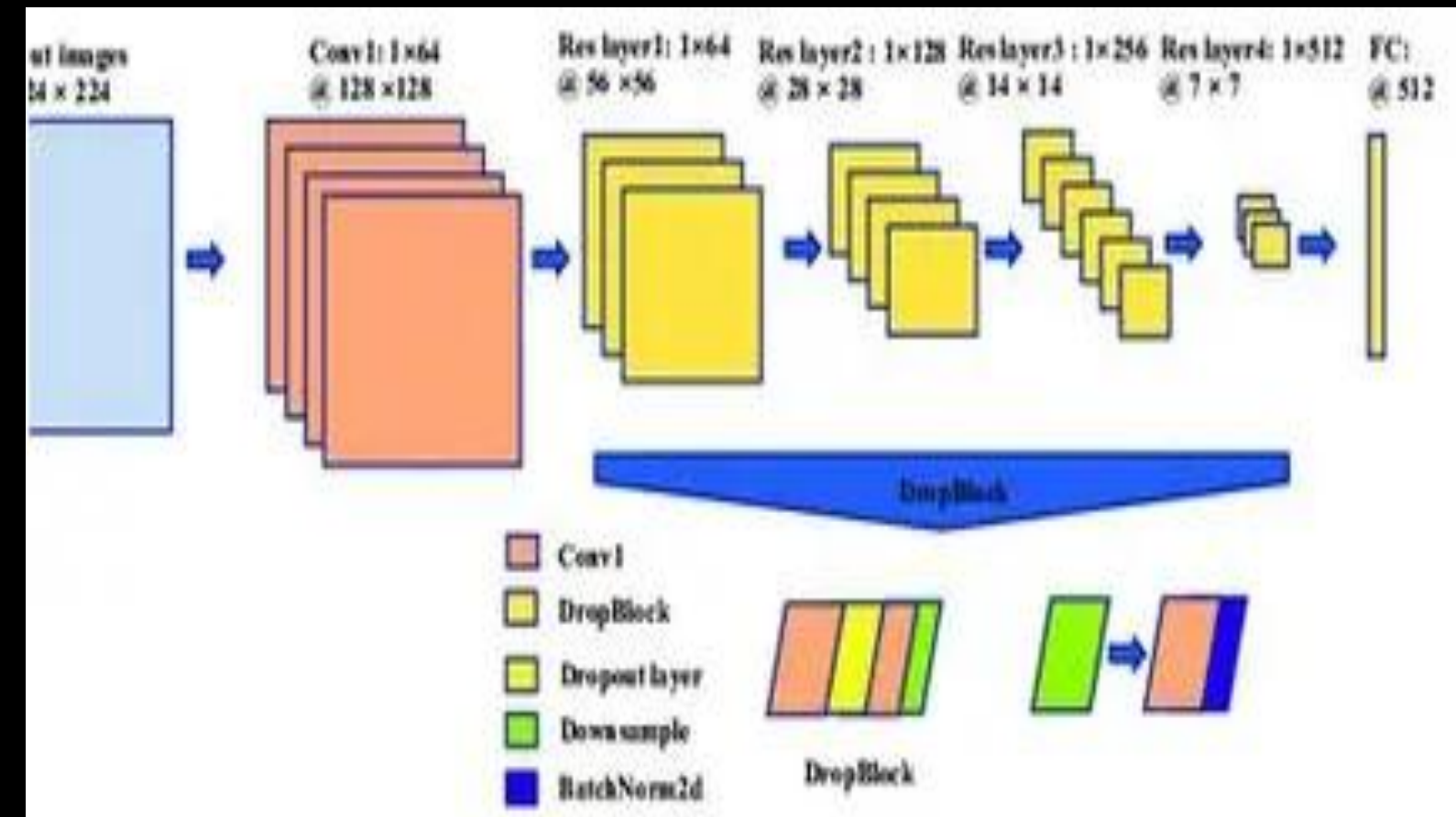
# RESNET (RESIDUAL NETWORKS) ARCHITECTURE

- **Connectivity** : Incorporates skip connections or residual blocks to allow gradients to bypass one or more layers, solving the vanishing gradient problem in very deep networks.
- **Formulation**: Residual blocks are represented as:

$$y = F(x) + x$$

where  $F(x)$  is the residual mapping learned by stacked layers, and the addition directly helps in maintaining feature integrity over depth.

- **Feature Reuse**: Based on the hypothesis that residual learning makes it easier for the network to learn identity mappings, enabling deeper architectures without degradation.



# SIMPLE CNN & RESNET

**Simple CNN** and **RESNET** are two **CNN** architectures used for pneumonia detection from chest X-ray images. Simple CNN is a lightweight model composed of sequential convolution and pooling layers for fast and efficient feature extraction.

**ResNet (Residual Network)** introduces skip connections that prevent vanishing gradients, enabling the training of deeper and more accurate networks. Both were trained on same dataset to compare performance in identifying normal and pneumonia affected lungs.

$$y = f(x) + x \text{ (Residual learning in ResNet)}$$



# VISUALIZATION AND INTERPRETATION

**Grad-CAM** visualizations reveal that **ResNet** focuses sharply on infected lung regions, while **SimpleCNN** spreads attention more broadly. **T-SNE** plots of the learned features show preparation between normal and pneumonia samples in **ResNet**, confirming it's stronger representational learning. Together, these analyses validate that **ResNet** provides more reliable and interpretable results for pneumonia detection.

## Concept Formula :

**Grad-CAM:  $L_{\text{GradCAM}} = \text{ReLU}(\sum_k a_k A^k)$**

*Where  $A^k$  = activation maps,  $a_k$  = gradient weights*



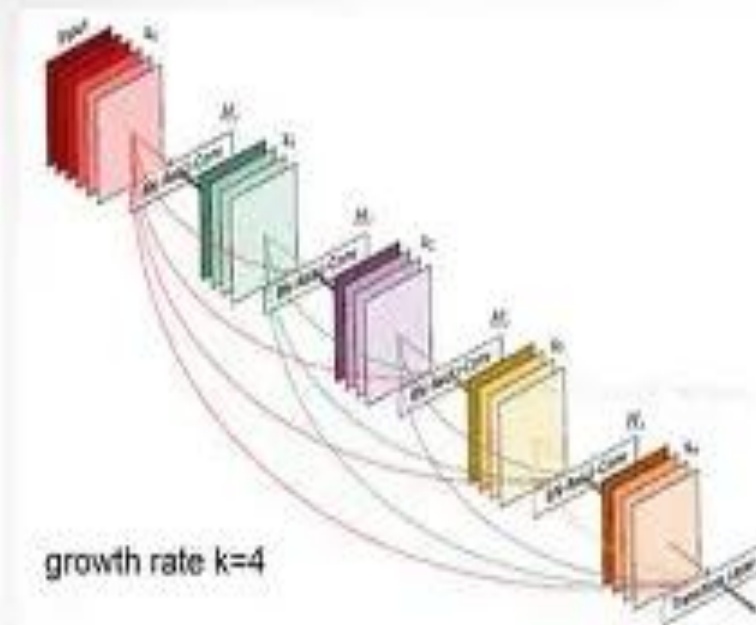
# DENSENET (DENSELY CONNECTED CONVOLUTIONAL NETWORKS)

- **Connectivity:** Connects each layer to every other layer in a feed-forward fashion, promoting feature reuse and reducing the number of parameters.
- **Feature Reuse:** The dense connectivity allows gradients to flow more efficiently during training, which is particularly useful for small datasets or transfer learning in medical imaging.
- **Formulation:** DenseNet layers are connected via concatenation rather than addition, which preserves features learned in earlier layers and improves gradient flow:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}])$$

where **H** is a composite of batch normalization, **ReLU**, convolution and **[]** denotes concatenation.

DenseNet (CVPR 2017 Best Paper)



$$x_l = H_l(x_{l-1})$$

$$H = \text{BN} \mid \text{ReLU} \mid \text{Pooling} \mid \text{Conv}$$

$$x_l = H_l(x_{l-1}) + x_{l-1}$$

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}])$$

$$\text{input\_feature\_maps}_l$$

$$= k \times (l - 1) + k_0$$

# COMPARITIVE INSIGHTS

- **Training Efficiency:** **ResNet** tends to have a faster convergence and often requires less training time to reach high accuracy, especially in larger datasets, due to simpler residual connections.
- **Feature Extraction:** **DenseNet's** dense connectivity leads to more meaningful features related to subtle textures and patterns which enhances interpretability and potentially better clinical relevance in imaging analysis.
- **Explainability:** **DenseNet** models often produce activation heatmaps concentrated on relevant pathological regions (e.g. *lung opacities*), providing more clinically aligned interpretability compared to **ResNet**, which sometimes shows activation around non-pathological areas.

# SUMMARY

**DenseNet's** dense connectivity fosters detailed feature reuse, often resulting in higher sensitivity and interpretability for pneumonia detection, especially when combined with transfer learning and limited datasets.

**ResNet** with its residual learning, remains a powerful, computationally efficient architecture suitable for larger datasets.

Both models are effective, but **DenseNet** generally shows a performance edge in subtle medical image analysis, aligning with clinical interpretability needs.

# TABLE

Feature	DenseNet-121	ResNet-50
Key Features	Dense connectivity, feature reuse	Residual blocks, skip connections
Formula	$x_{\{l\}} = H_{\{l\}}([x_{\{0\}}, \dots, x_{\{l-1\}}])(\text{concatenation})$	$y = F(x) + x(\text{addition residual connection})$
Gradient flow	Improved via dense connectivity	Facilitated by residual skip connections
Model interpretability	Better localization of relevant pathological regions	Sometimes activation at less relevant areas
Training time and convergence	Slightly slower but more detailed feature extraction	Faster convergence in larger datasets



# VGG ARCHITECTURE

**VGG**, which stands for **Visual Geometry Group**, is a deep **convolutional neural network (CNN)** architecture known for its *simple, uniform design* and *deep layers*.

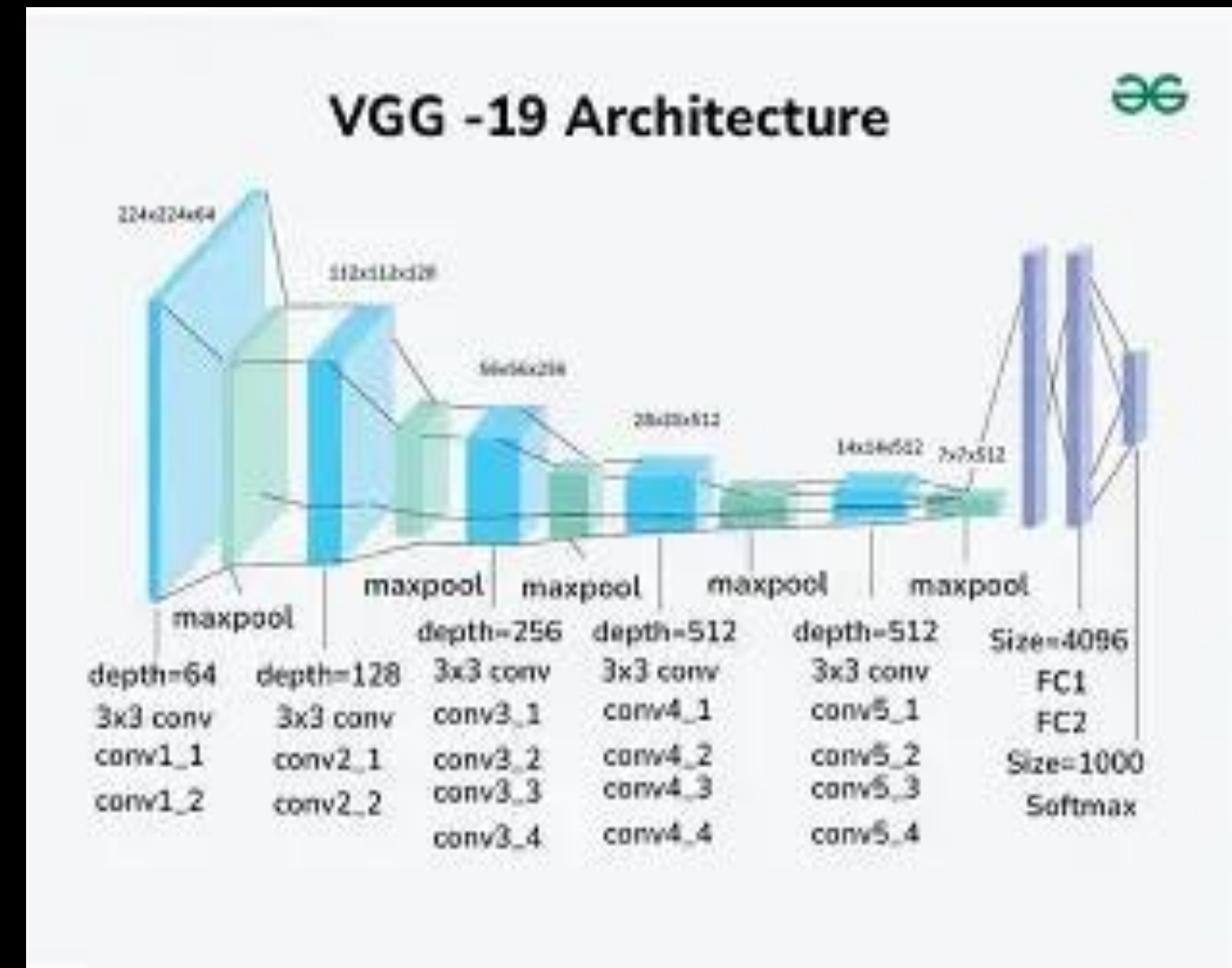
## Key Features of VGG Architecture

### Uniform Architecture:

- Uses only **3×3 convolutional filters** (the smallest size that can capture spatial patterns in all directions) and **2×2 max pooling** throughout the network.
- This consistent design makes it easier to generalize and extend.

### Depth:

VGG networks come in different versions — mainly **VGG-16** and **VGG-19**, referring to the total number of layers (16 and 19, respectively, including convolutional and fully connected layers).



- **Stacked Convolutions:**

- Instead of using large filters (like  $5 \times 5$  or  $7 \times 7$ ), VGG uses multiple small  $3 \times 3$  filters stacked together.
- This increases the **non-linearity** and **effective receptive field** while reducing the number of parameters.

- **Pooling Layers:**

- **Max pooling** with  $2 \times 2$  filters and a stride of 2 is applied after every few convolutional layers to reduce spatial dimensions gradually.

- **Fully Connected Layers:**

- After the convolutional and pooling layers, the network has **three fully connected (FC) layers** — the first two with 4096 neurons and the last with 1000 neurons (for ImageNet's 1000 classes).
- The final layer uses a **Softmax** activation function for classification.

## Advantages

- Simple and uniform design makes it easy to implement.
- Deep structure allows learning of complex image features.
- Performs well in transfer learning (used widely as a backbone in many vision models).

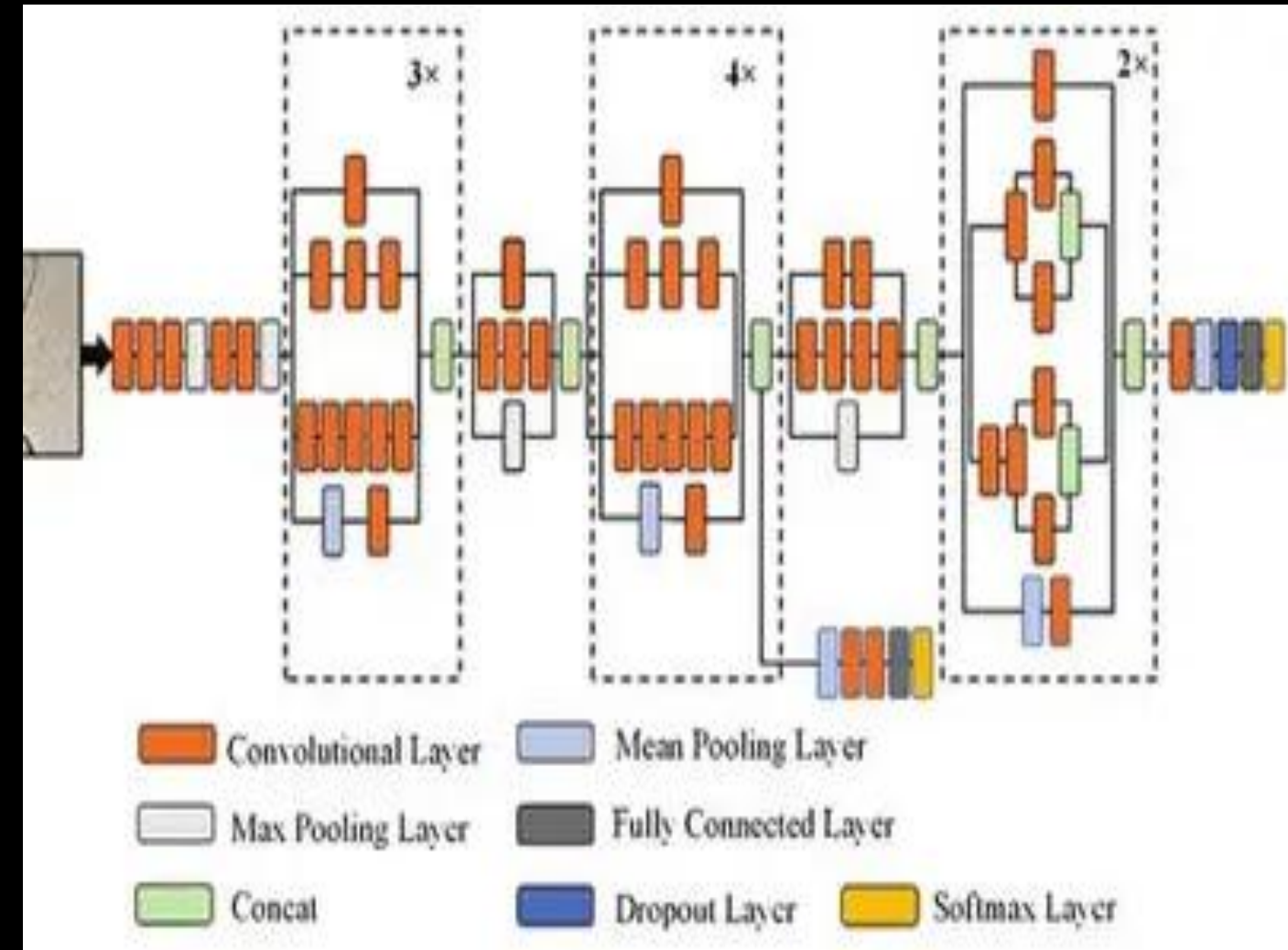
## Limitations

- **High computational cost** — requires a large amount of memory and training time.
- **Large number of parameters** ( $\approx 138$  million in VGG-16).
- Not as efficient as newer architectures (e.g., *ResNet*, *EfficientNet*).



# INCEPTION ARCHITECTURE

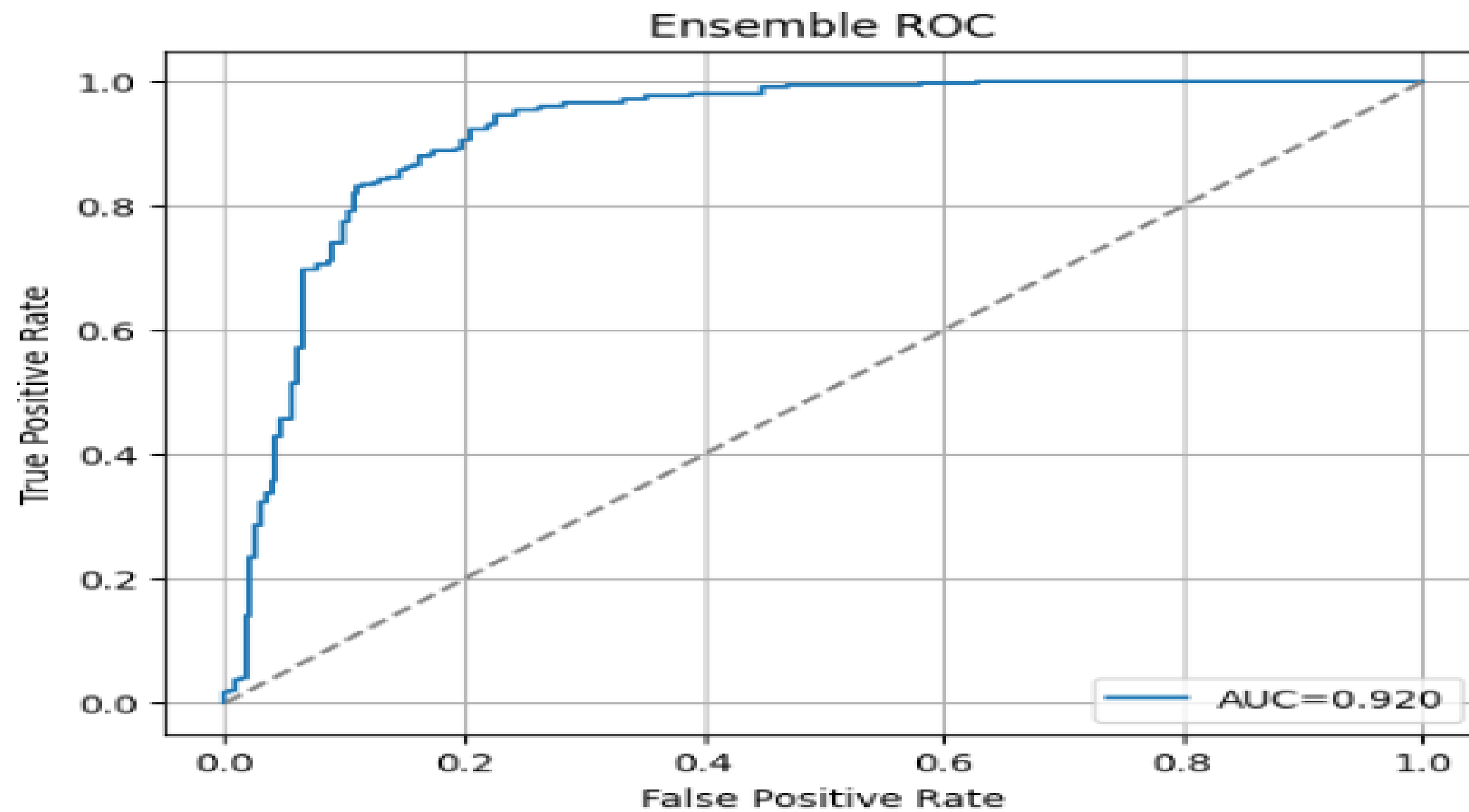
- ❑ The **Inception architecture**, also known as **GoogLeNet**, was introduced by in the 2014 paper “*Going Deeper with Convolutions*” by Google. It marked a major step forward in designing efficient deep convolutional networks, achieving **high accuracy** with **fewer parameters** compared to models like VGG.
- ❑ The main concept behind the Inception architecture is to **capture features at multiple scales simultaneously** by applying **different types of convolution and pooling operations in parallel** and then concatenating their outputs. This allows the network to decide which filter size is most useful for each spatial pattern without manually tuning it.





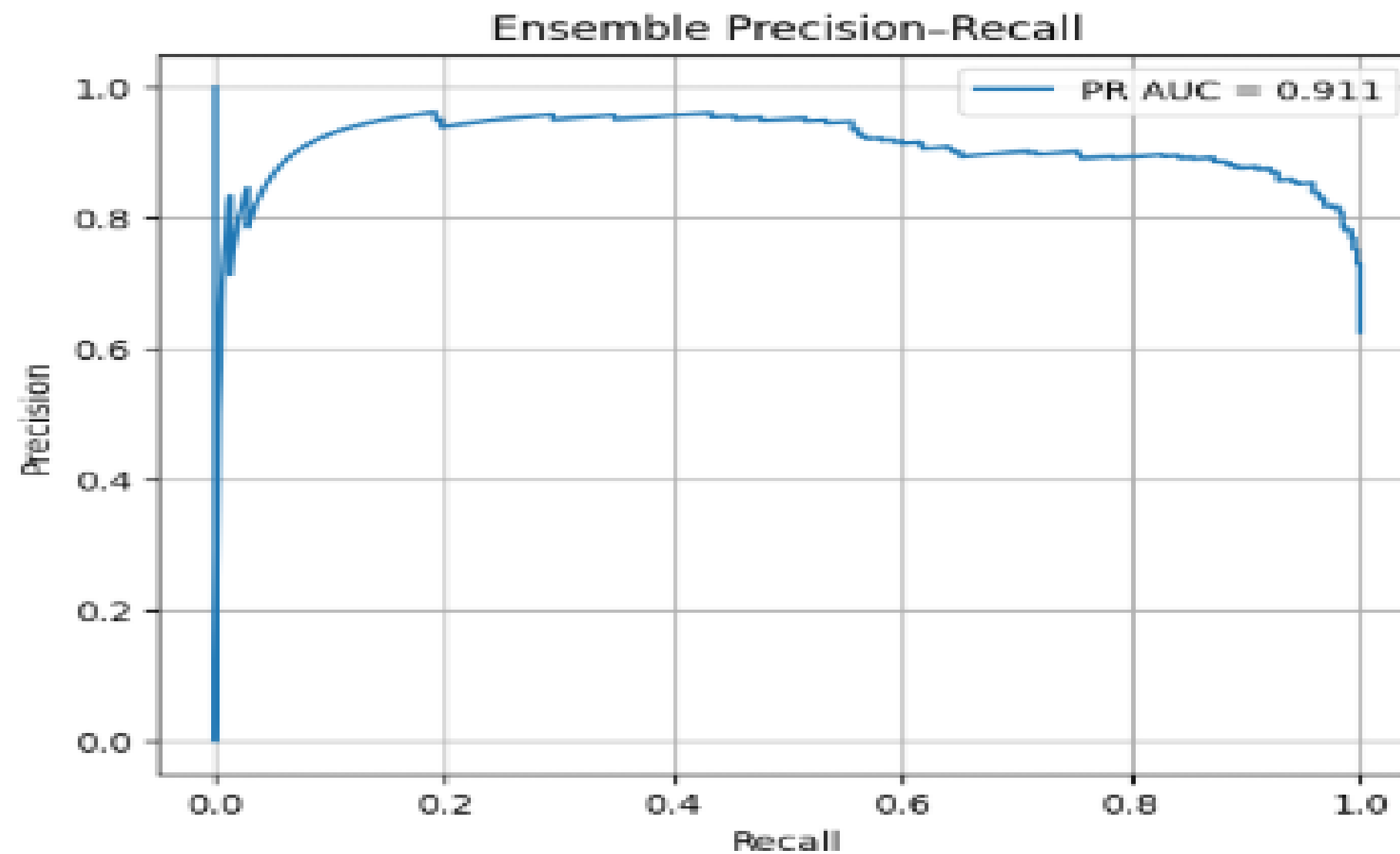
# ENSEMBLE ROC

Ensemble ROC AUC: 0.9199



# ENSEMBLE PRECISION AND RECALL

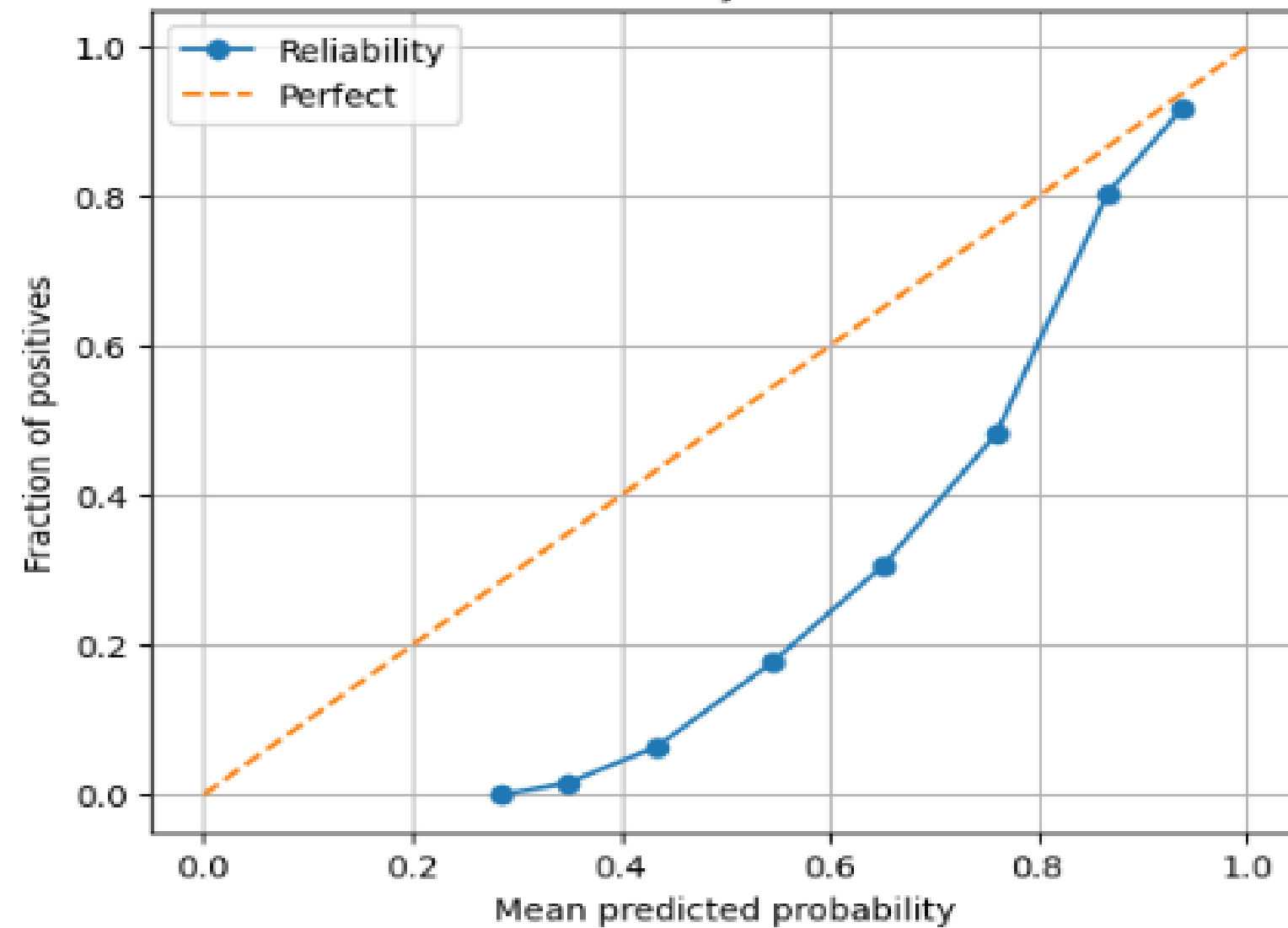
=== OUTPUT 3: PR AUC (Ensemble) = 0.9111



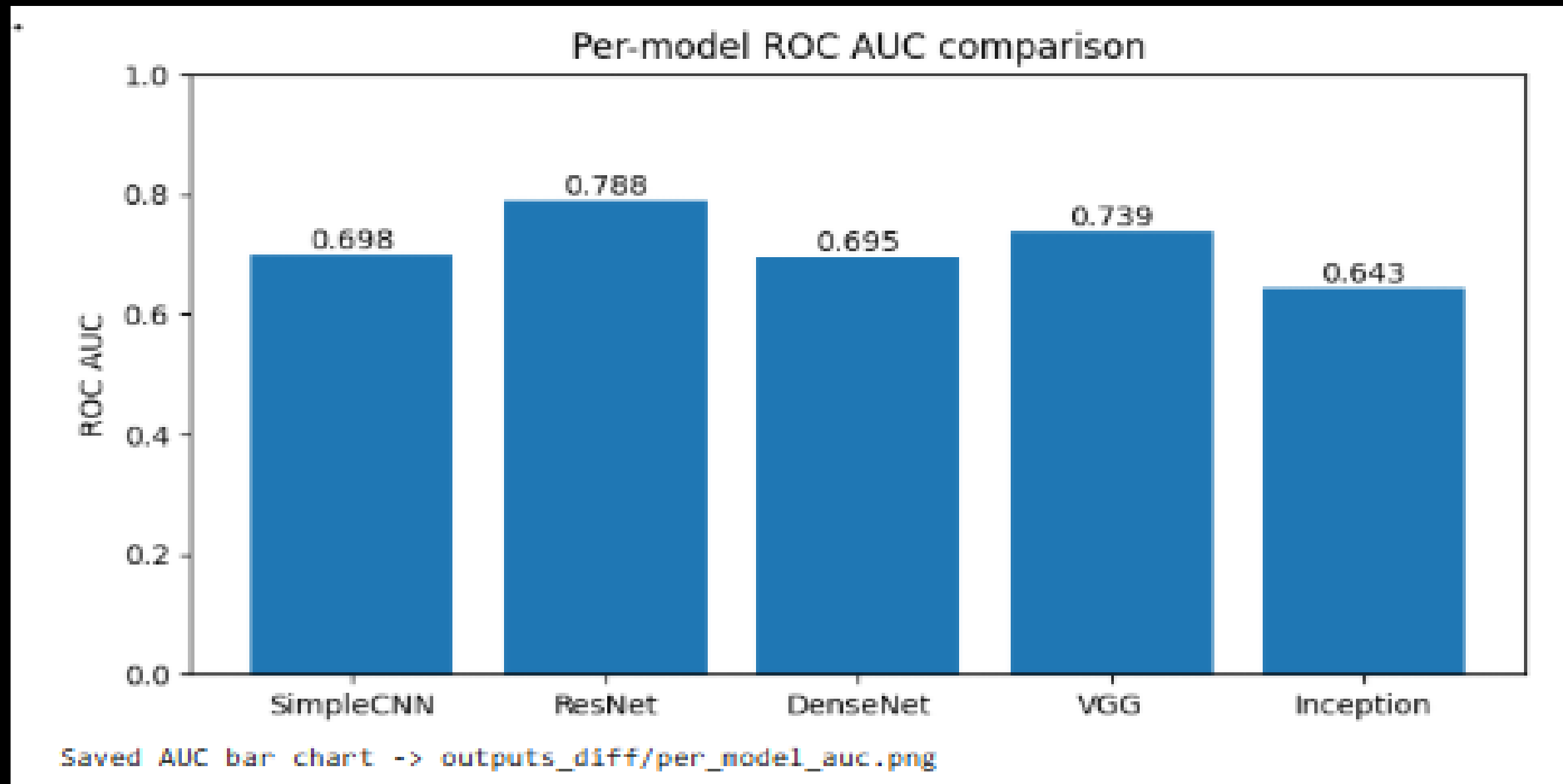
# RELIABILITY CURVE

=== OUTPUT 4: Brier score (Ensemble) = 0.1489 (lower is better)

Calibration (Reliability) Curve — Ensemble

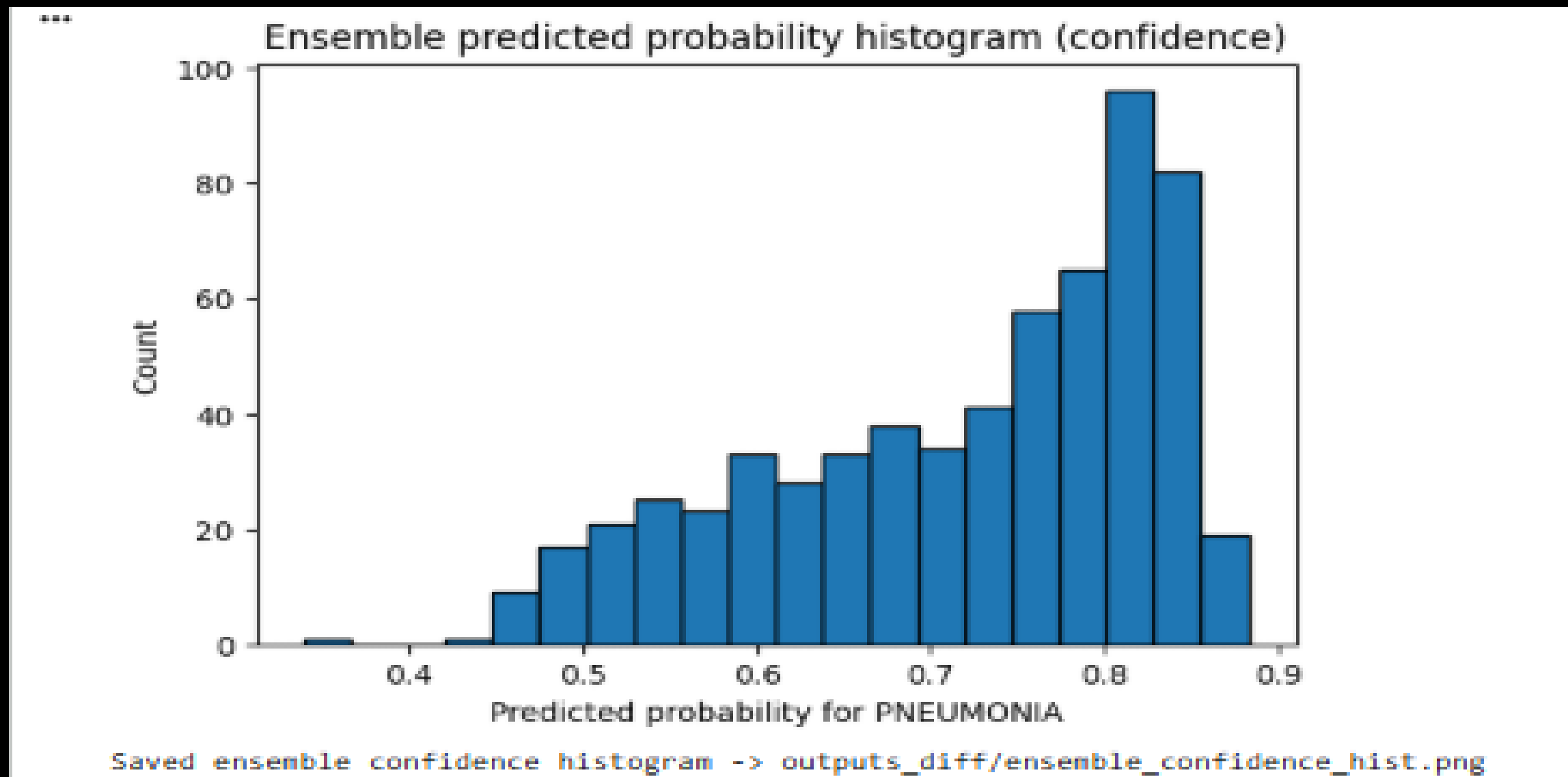


# PER-MODEL ROC AUC COMPARISON

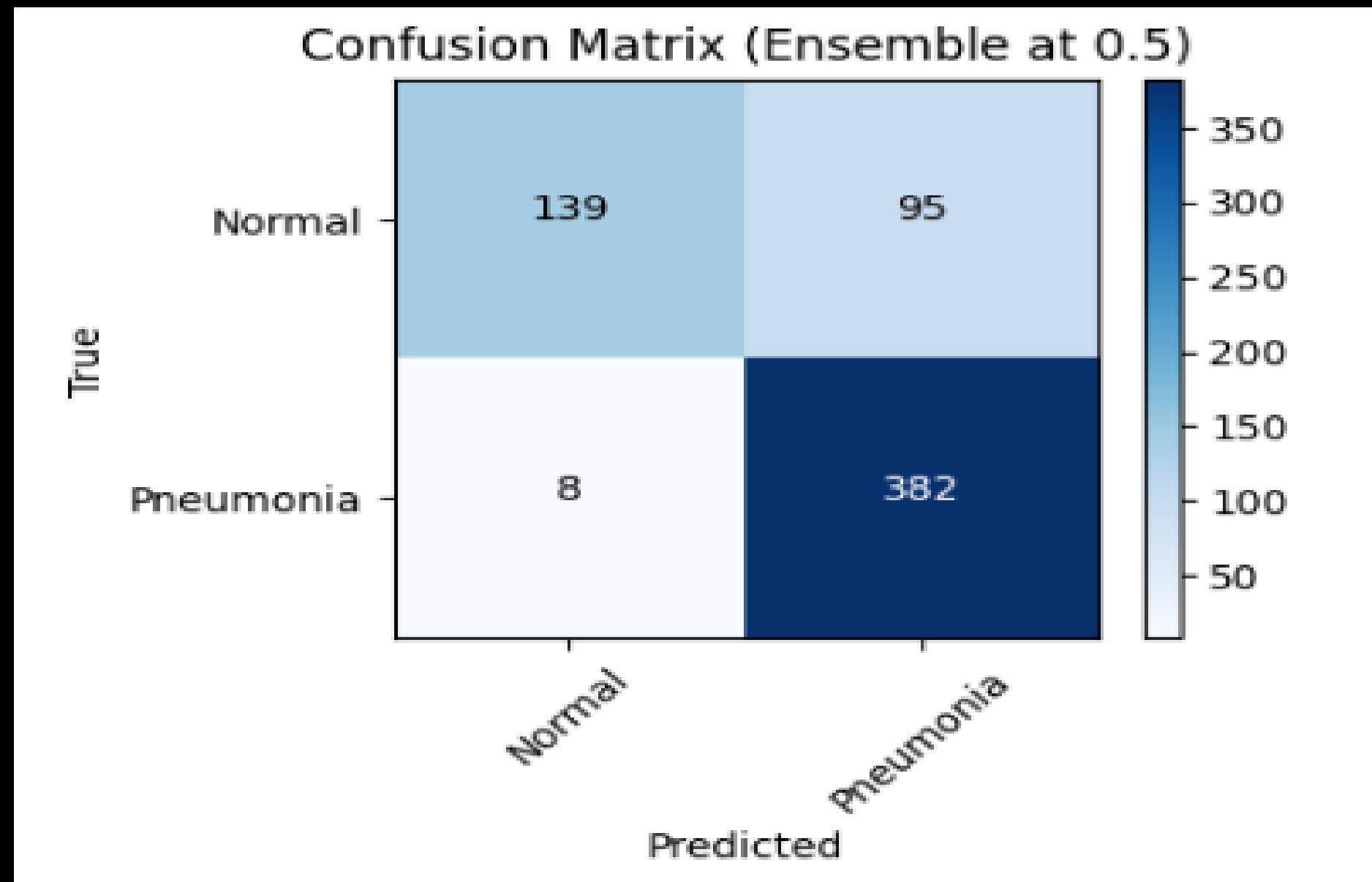




# HISTOGRAM



# CONFUSION MATRIX



# CLASSIFICATION REPORT

\*\*\*

Classification report (ensemble at 0.5):

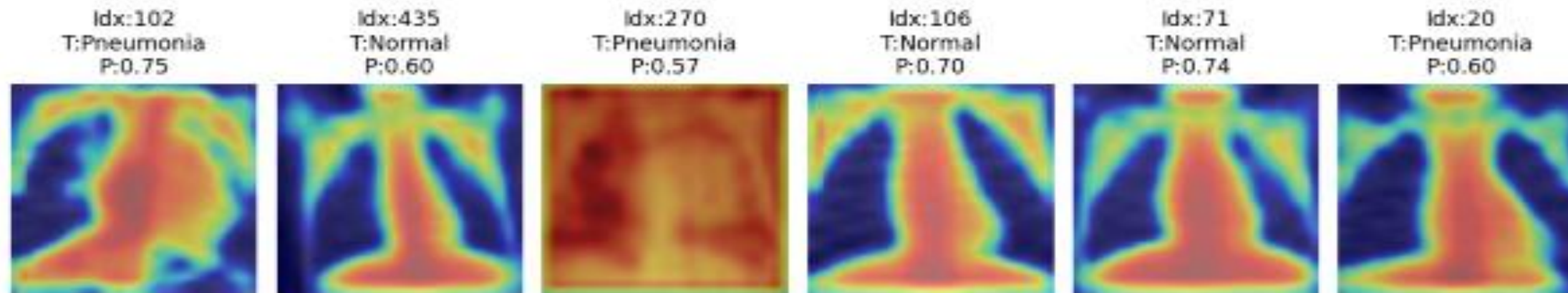
	precision	recall	f1-score	support
Normal	0.9456	0.5940	0.7297	234
Pneumonia	0.8008	0.9795	0.8812	390
accuracy			0.8349	624
macro avg	0.8732	0.7868	0.8054	624
weighted avg	0.8551	0.8349	0.8244	624



# GRAD-CAM OVERLAYS-SIMPLECNN AND VGG ARCHITECTURE

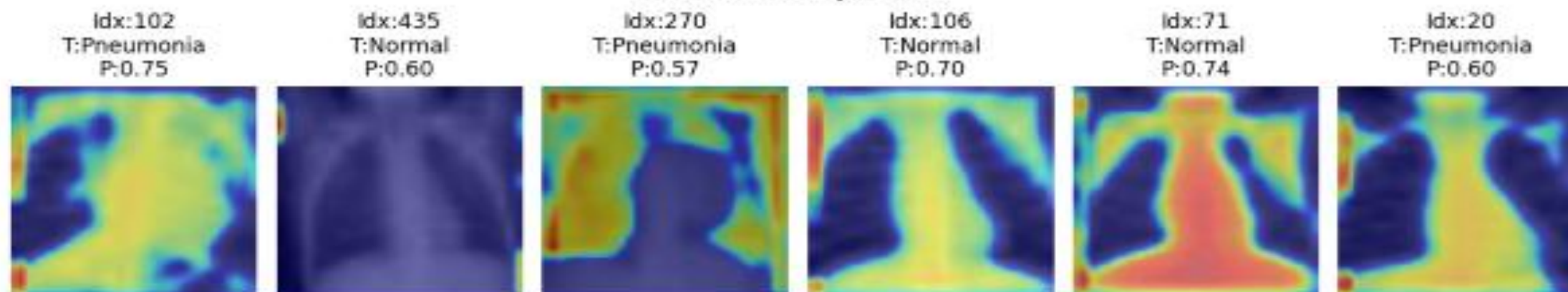
```
Grad-CAM examples indices: [102, 435, 270, 106, 71, 20]
/usr/local/lib/python3.12/dist-packages/keras/src/models/functional.py:241: UserWarning: The structure of "inputs" doesn't match the expected structure.
Expected: [['keras_tensor_268']]
Received: inputs=Tensor(shape=(1, 64, 64, 1))
warnings.warn(msg)
```

Grad-CAM overlays — SimpleCNN



```
/usr/local/lib/python3.12/dist-packages/keras/src/models/functional.py:241: UserWarning: The structure of "inputs" doesn't match the expected structure.
Expected: [['keras_tensor_295']]
Received: inputs=Tensor(shape=(1, 64, 64, 1))
warnings.warn(msg)
```

Grad-CAM overlays — VGG



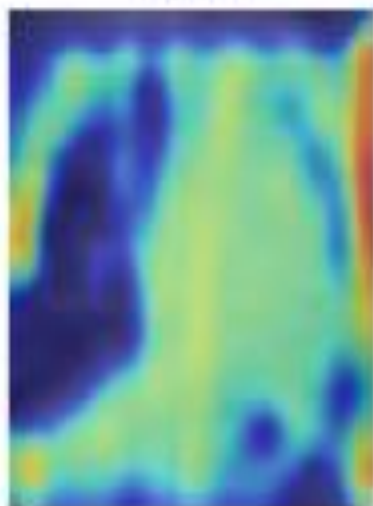
```
/usr/local/lib/python3.12/dist-packages/keras/src/models/functional.py:241: UserWarning: The structure of "inputs" doesn't match the expected structure.
Expected: [['keras_tensor_275']]
Received: inputs=Tensor(shape=(1, 64, 64, 1))
warnings.warn(msg)
```

# GRAD-CAM OVERLAYS-RESNET

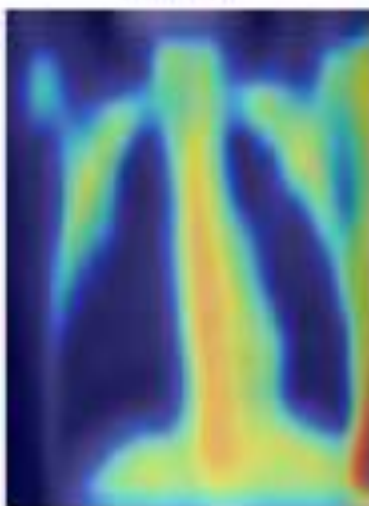
warnings.warn(msg)

Grad-CAM overlays — ResNet

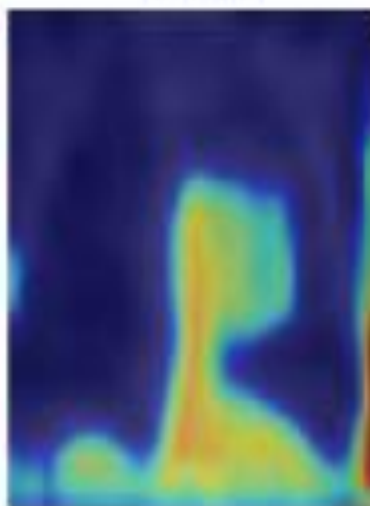
Idx:102  
T:Pneumonia  
P:0.75



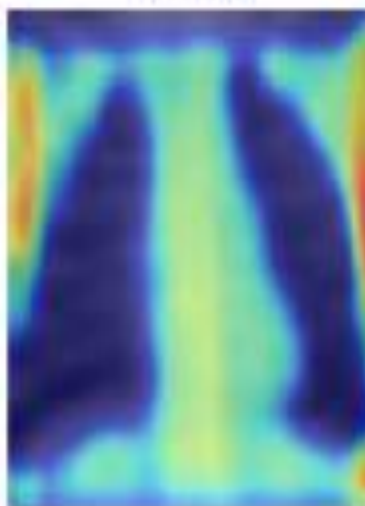
Idx:435  
T:Normal  
P:0.60



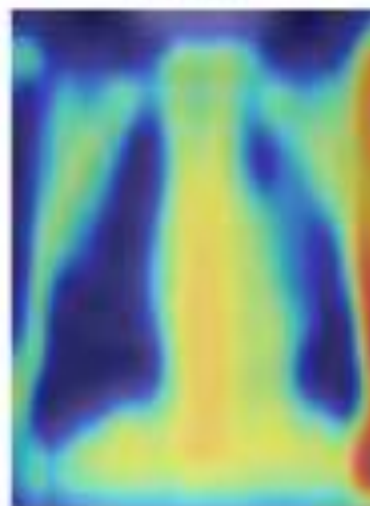
Idx:270  
T:Pneumonia  
P:0.57



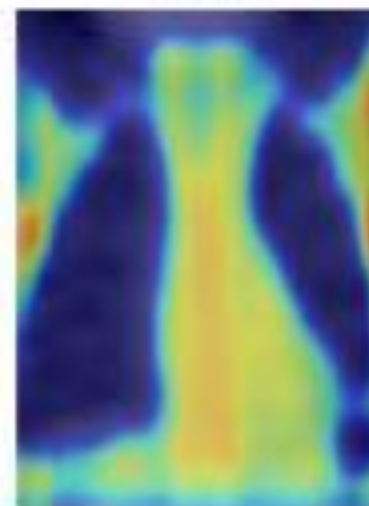
Idx:106  
T:Normal  
P:0.70



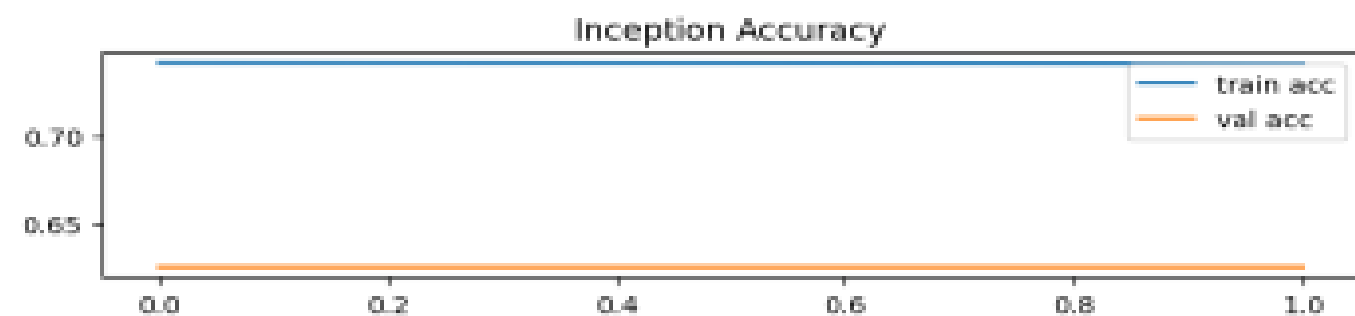
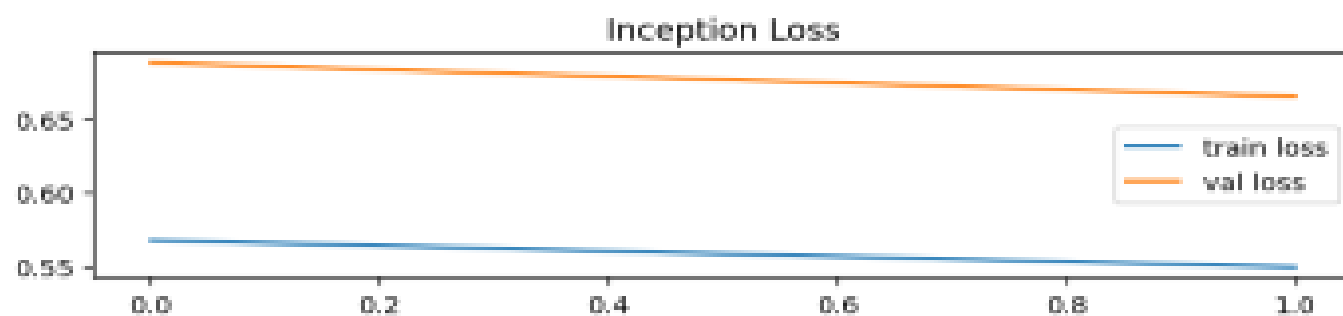
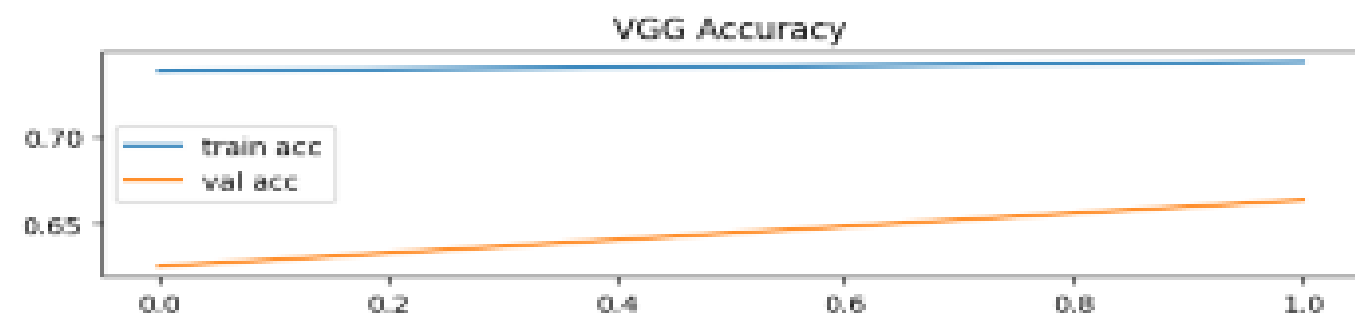
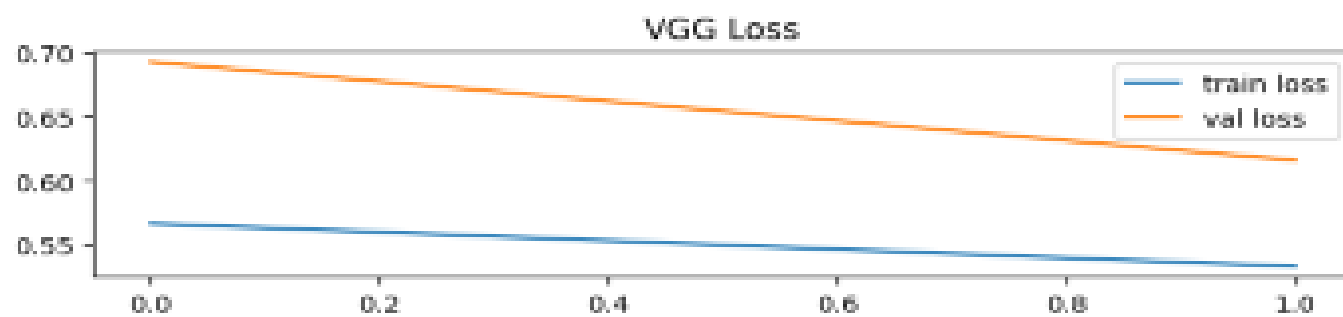
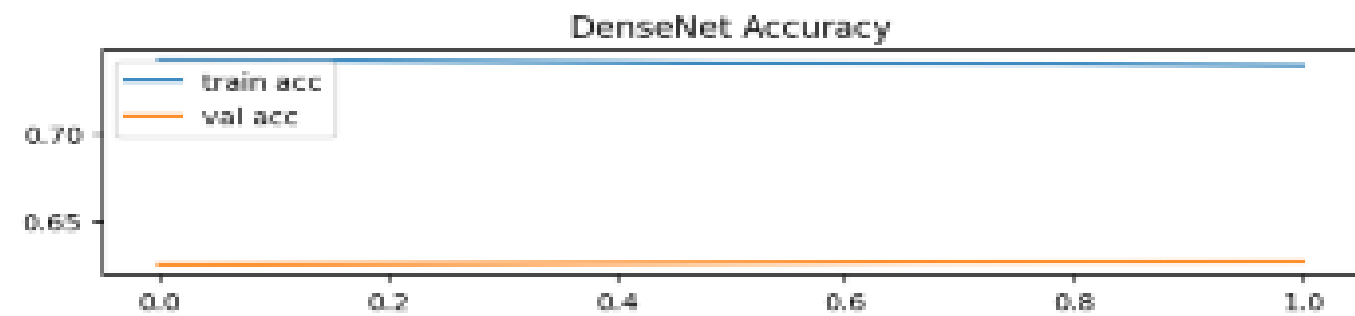
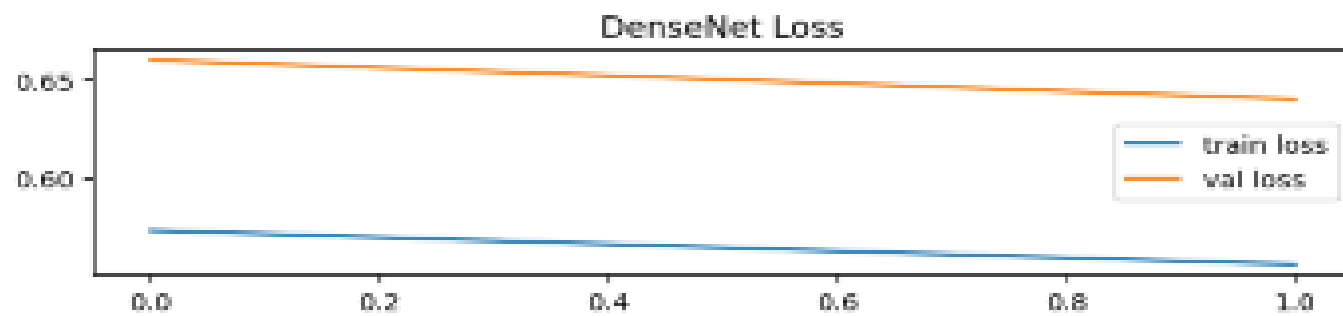
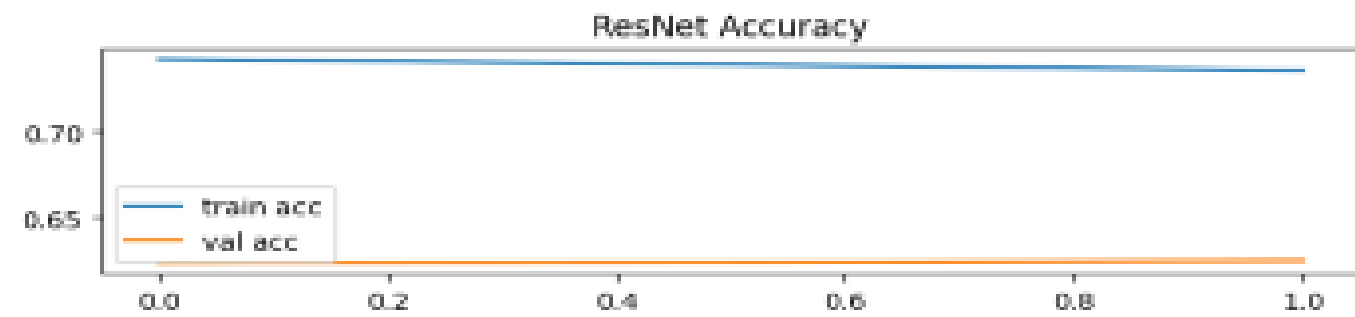
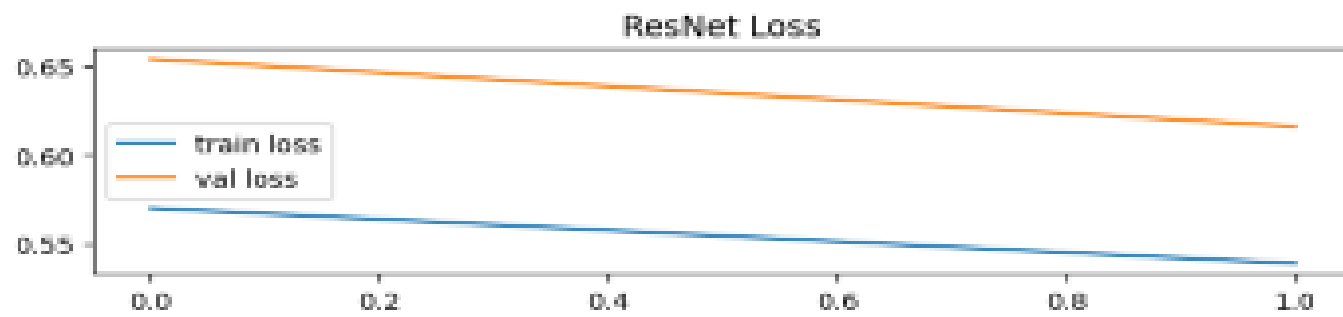
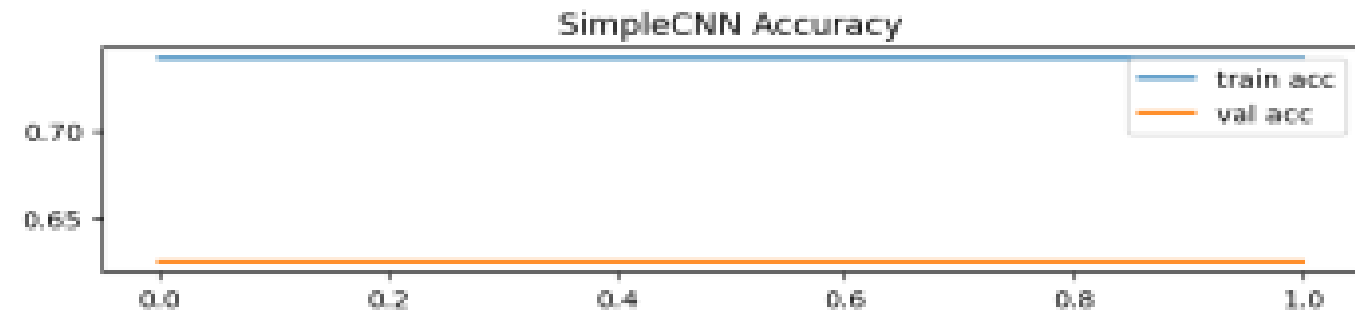
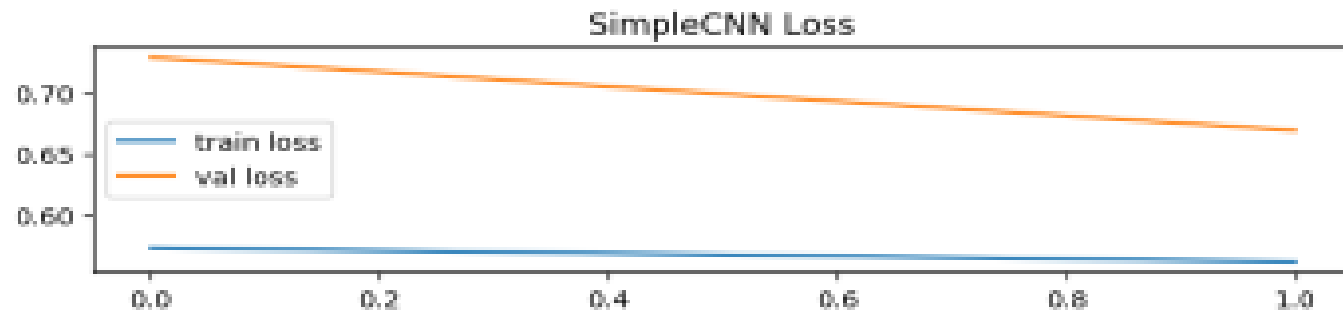
Idx:71  
T:Normal  
P:0.74



Idx:20  
T:Pneumonia  
P:0.60

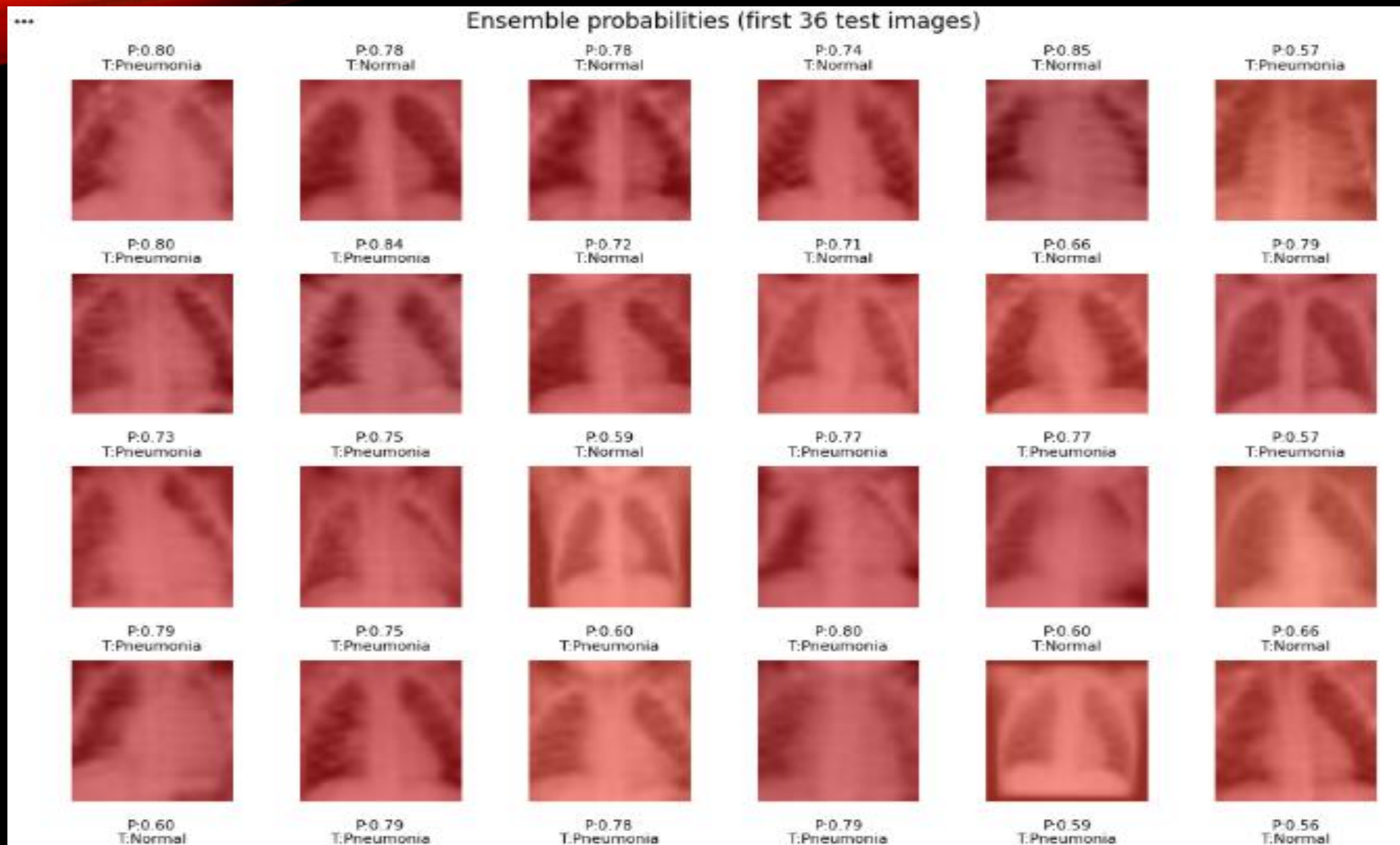


# ACCURACY AND LOSS





# ENSEMBLE PROBABILITIES



# CONCLUSION: OVERVIEW AND INSIGHT

Our project successfully demonstrated the use of **Deep Learning (CNNs)** for detecting **pneumonia from chest X-rays**.

Five architectures — **SimpleCNN**, **ResNet-like**, **DenseNet-like**, **VGG-like**, and **Inception-like** — were trained and compared to evaluate performance and accuracy.

Each model learned to identify key visual differences between normal and infected lungs.

- **SimpleCNN**: Fast but less accurate.
- **ResNet-like**: Better gradient flow and higher stability.
- **DenseNet-like**: Strong feature reuse and best sensitivity.  
The study proved that deeper CNNs with advanced connectivity provide more reliable results in medical image classification.
- **VGGNet**: Deep yet straightforward architecture using stacked 3×3 convolutions; achieves good accuracy but requires high computational resources.
- **InceptionNet**: Employs multi-scale feature extraction through parallel convolutional filters, improving efficiency and performance.

# CONCLUSION: COMPARATIVE OUTCOME

Model performance was evaluated using **Accuracy, Precision, Recall, and F1-Score** with a **Confusion Matrix** for detailed analysis.

- **SimpleCNN**: 82.37%
- **ResNet-like**: 62.66%
- **DenseNet-like**: 62.66%
- **VGG-like**: 85.58%
- **Inception-like**: 62.50%

**Grad-CAM visualization** confirmed that among all architectures, the **VGG-like model achieved the best overall performance**, demonstrating the highest accuracy and strong consistency across metrics, making it the most reliable and effective model for the given task.



# CONCLUSION: FUTURE SCOPE AND REAL WORLD IMPACT

This project highlights how **AI-driven Deep Learning models** can assist doctors by enabling **early and accurate pneumonia detection**. They can reduce diagnostic workload, save time, and improve patient outcomes.

## Future Scope:

- Expand dataset for better generalization.
- Use **transfer learning** for higher efficiency.
- Add **explainable AI** for medical transparency.
- Integrate with **cloud or mobile platforms** for real-time use.
- In summary, our work shows that **Deep Learning can revolutionize medical imaging**, supporting faster, smarter, and more reliable healthcare solutions.

# REFERENCE

1. Yang, Y., Zhang, L., Du, M., et al. A Comparative Analysis of Neural Network Architectures for Lung Image Classification in COVID-19 Detection. **Computers in Biology and Medicine**, 2021.
2. Zhou, T., Ye, X., Lu, H., Zheng, X., Qiu, S., Liu, Y. Dense Convolutional Network (DenseNet) and Its Application in Medical Image Analysis. **Frontiers in Computational Neuroscience**, 2022
3. Zhang, X., Ma, Z., Yu, F. (2021). ResNet or DenseNet? Introducing Dense Shortcuts to ResNet. **IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)**, 2021.
4. Putra, I. P. G. Y., Dewi, N. W. J. K., Lesmana, P. S. U., Suryawan, I. G. T. Comparison of ResNet-50 and DenseNet-121 Architectures in Classifying Diabetic Retinopathy. **International Journal of Data Science (IJODAS)**, 2024.