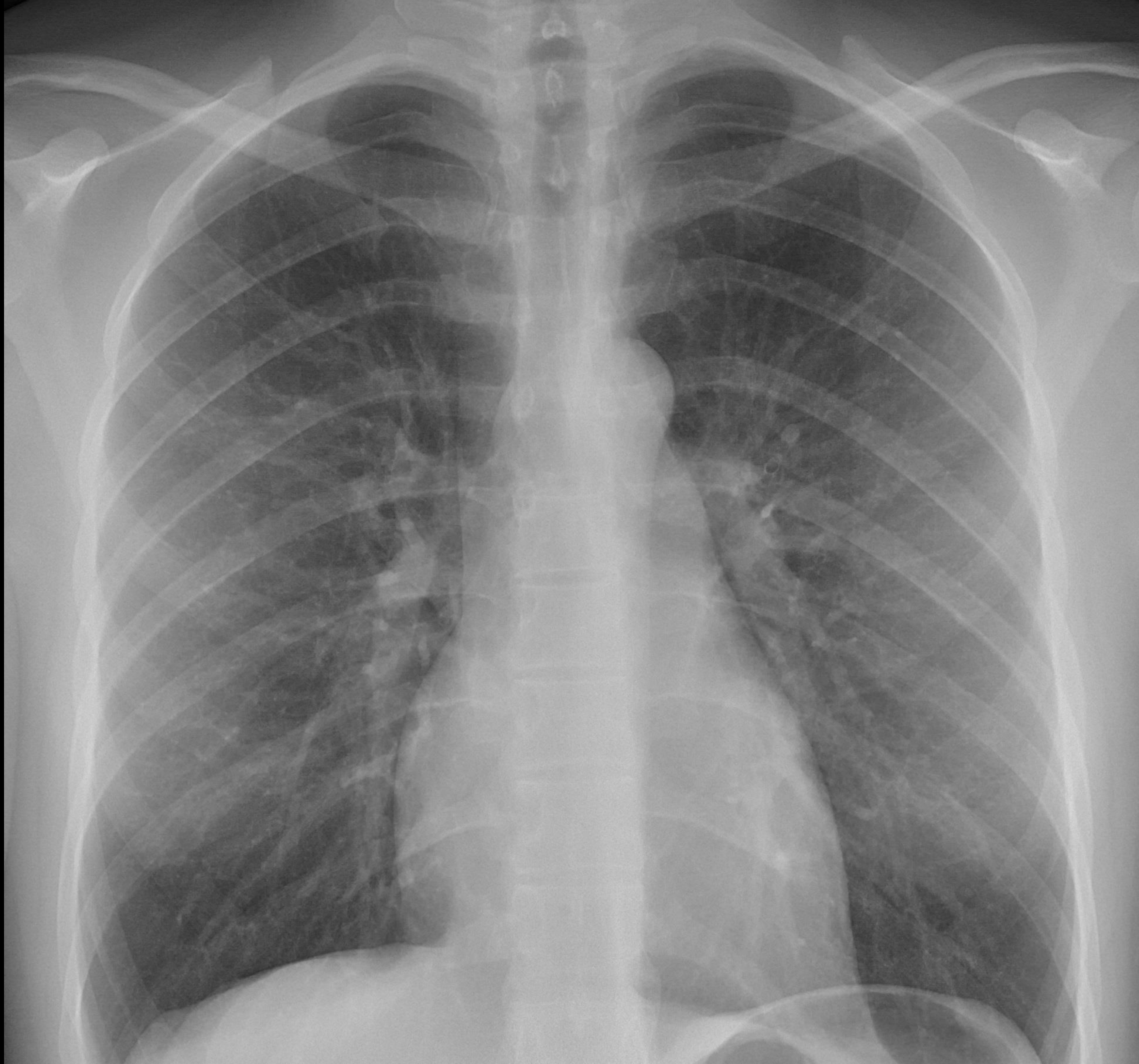
DEEP LEARNING TECHNIQUES FOR PNEUMONIA DETECTION

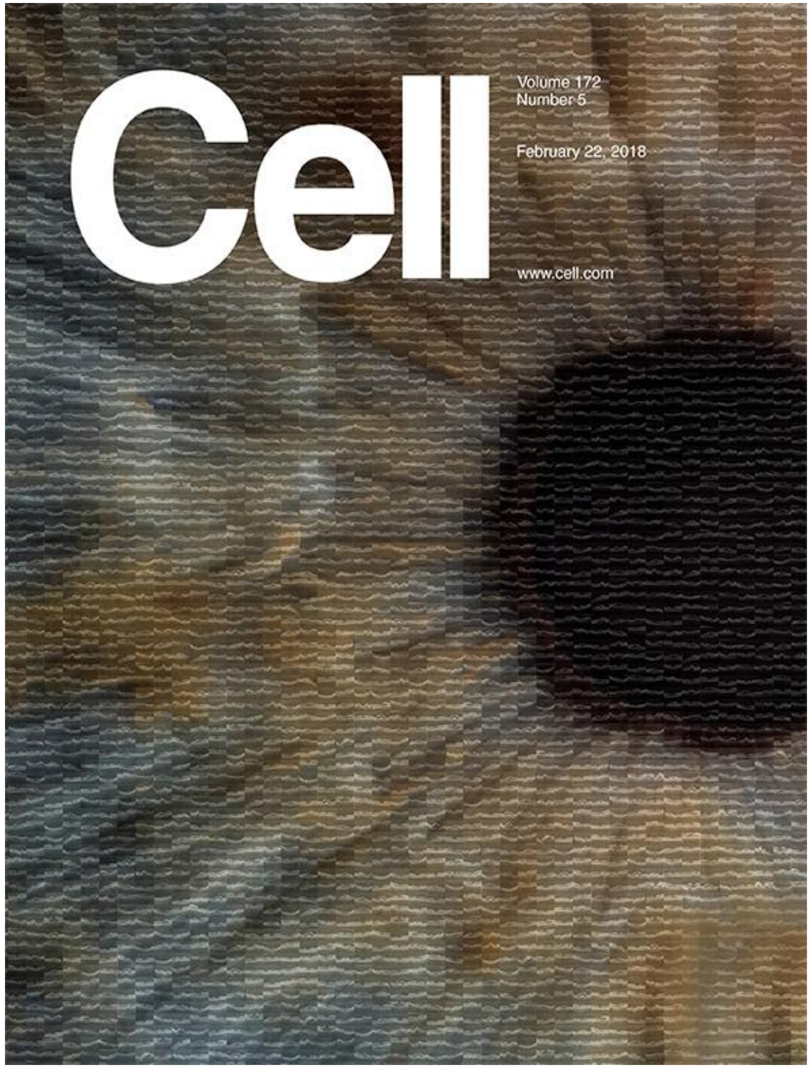
Deep Learning (A.A. 2024-2025)

Alberto Lazzeri



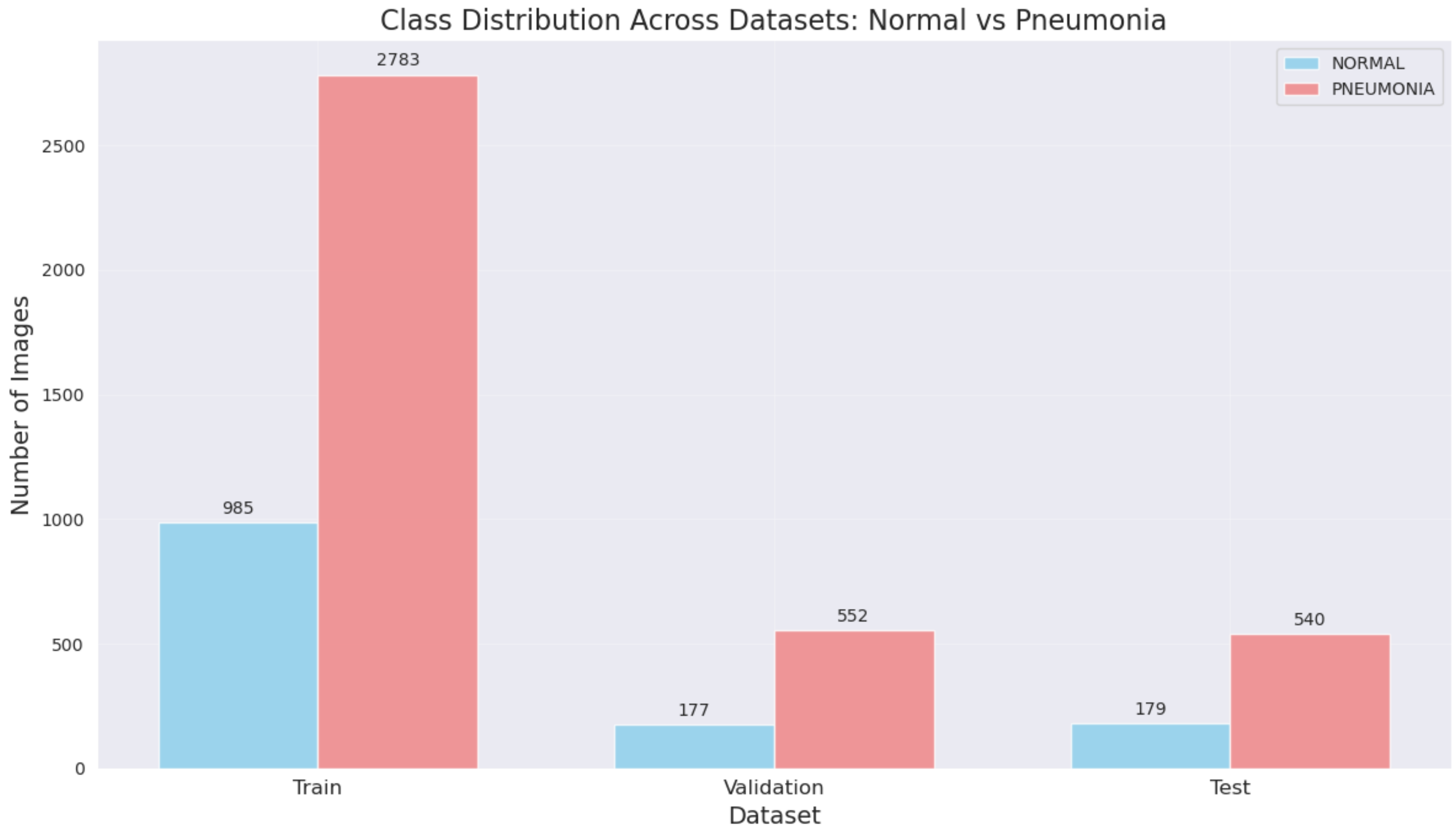
INTRODUCTION

- Apply deep learning techniques to medical diagnostics
- Challenges:
  - Interpretability
  - Dataset size and generalization
  - Reliabilty and safety
  - Clinical validation
- Task: detect pneumonia in pediatric patients
- Use of pre-trained models improves performances[1]



[1] "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning", Cell 2018

DATA VISUALIZATION

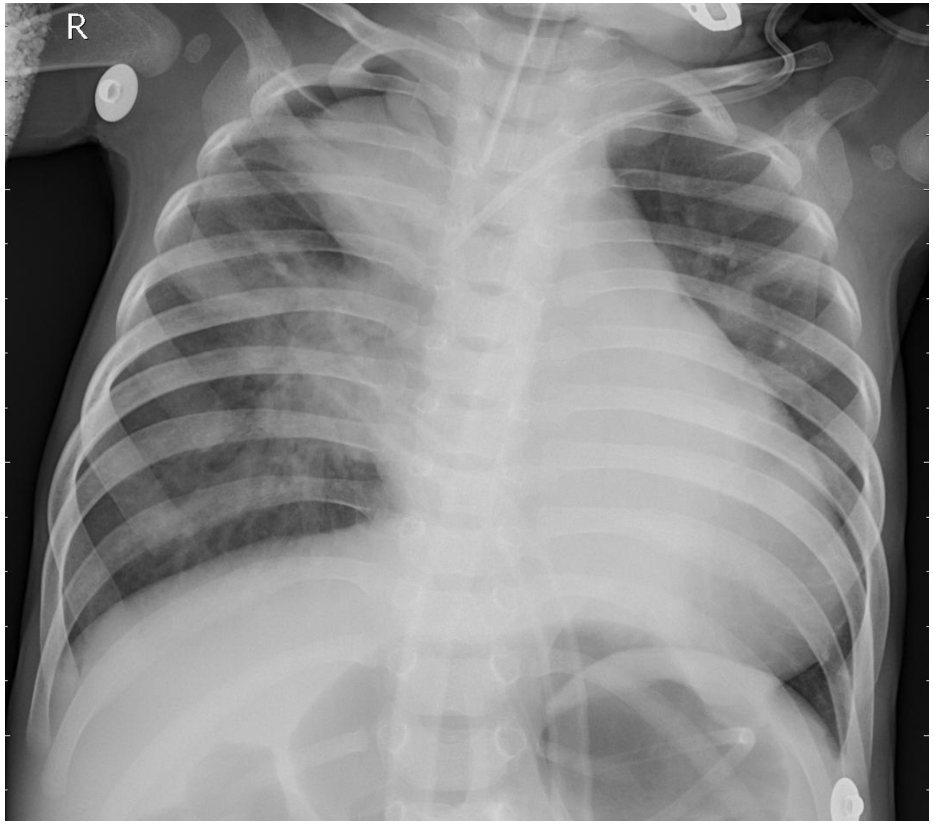


- X-rays of children chests[2]
- Dataset redistribution (total samples, 5216):
    - Train (~70%)
    - Validation (~15%)
    - Test (~15%)
- Remove duplicates between Validation and Test samples (keep in Train) to avoid overfitting
- Class imbalance: Pneumonia class (~75%) dominates over Normal class (~25%), use class weights
- Pixel values range and image dimensions

[2] Kaggle

DATA GENERATION AND AUGMENTATION

- Choose a standard image dimension (320 x 320)
- Batch size: 32
- Normalize pixels (division by 255) and duplicate channels (from [320, 320, 1] to [320, 320, 3])
- Augmentation, useful with small datasets:
  - Rotation (±10 degrees)
  - Translation (up to 5%)
  - Zoom (up to 5%)
  - Brightness adjustement (±5%)
  - Fill mode (nearest)
  - Random horizontal flip
- Class weights: {'NORMAL': 1.913, 'PNEUMONIA': 0.677}

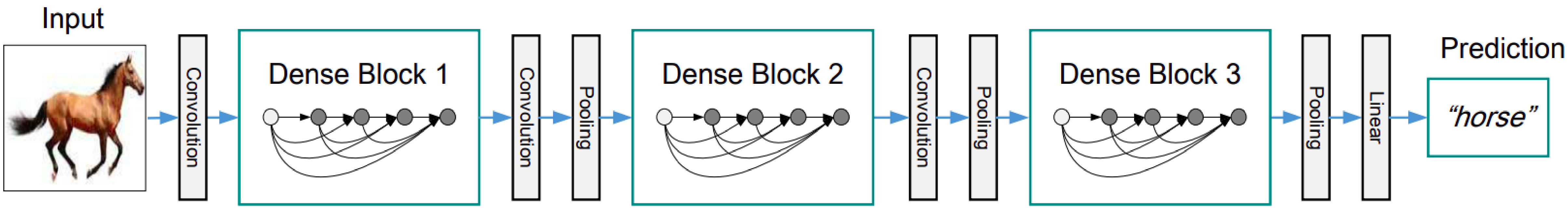


TRANSFER LEARNING

- Widely used for image classification: allows to have good performance on small datasets
- Complex models (Convolution-based) trained on ImageNet datasets (large datasets)
- Good accuracy and generalization ability
- Developed by Keras, Google, Facebook, etc., stored in Keras 3 API Application
- Model's choice factors (student with PC):
  - Reduced number of parameters and low RAM/GPU usage
  - Good trade-off between accuracy and efficiency
  - Strong performance on high-resolution grayscale medical images

DENSENET121[3]

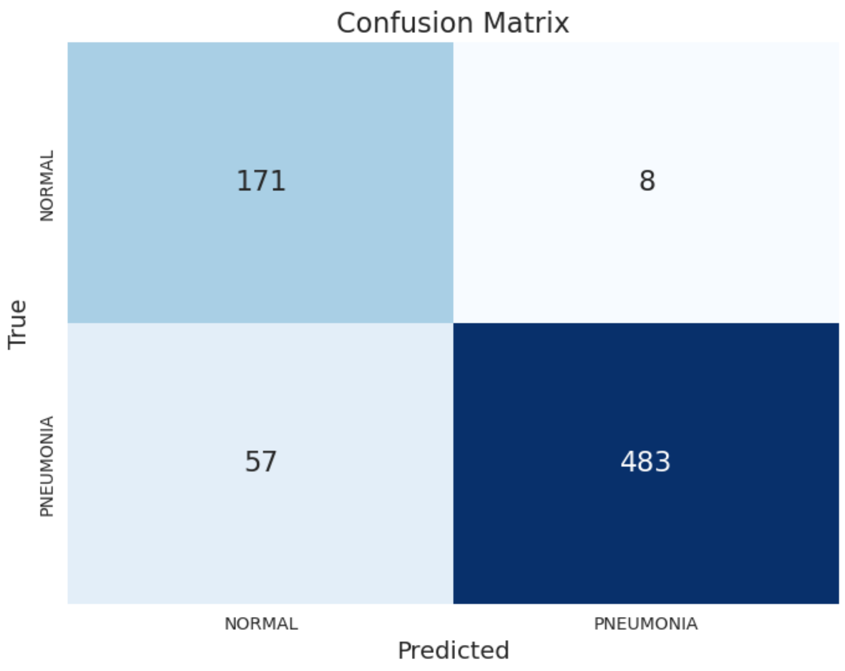
- Parameters: ~8M, Top-1 accuracy: 75.0%
- Feature maps processed as a sequence: $x_l = H_l([x_0, x_1, \ldots, x_{i-1}])$
- H( ) = BN + ReLU + 1×1 Conv (bottleneck) + BN + ReLU + 3×3 Conv

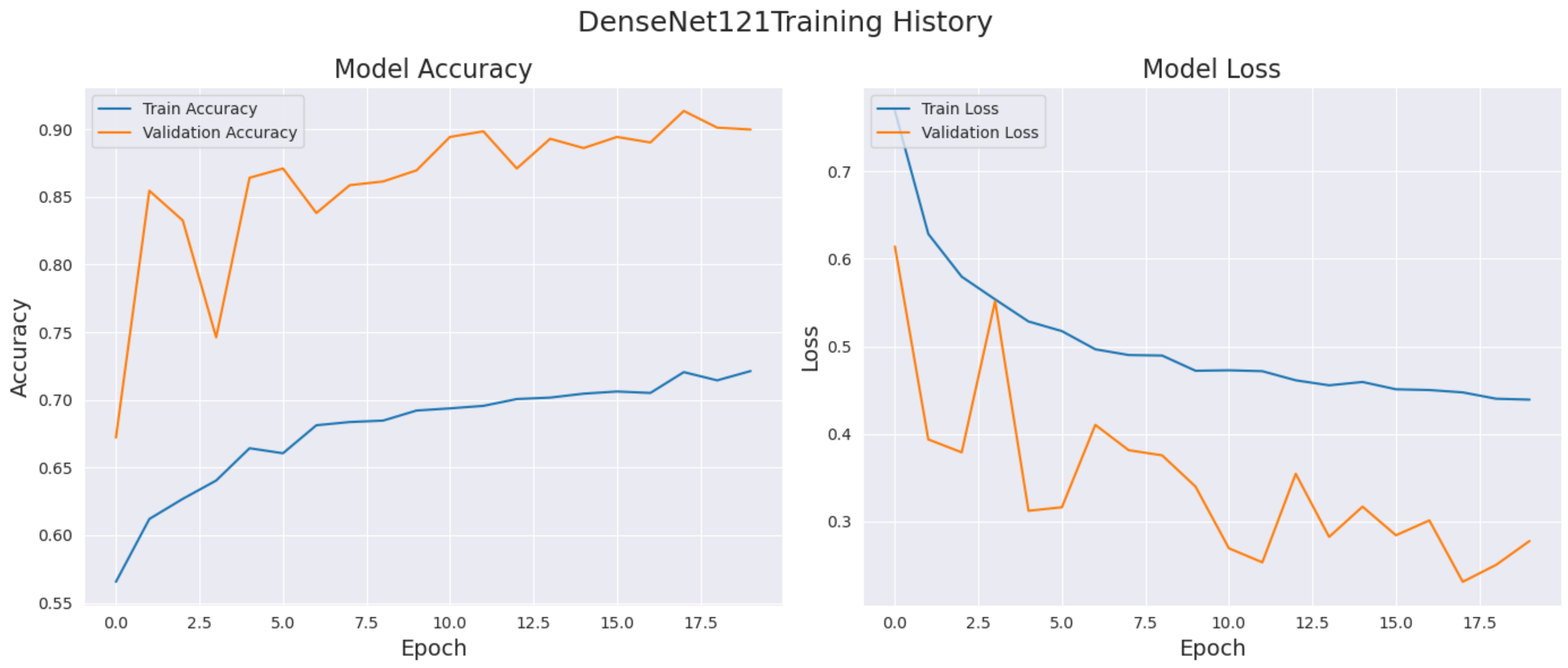


- Callbacks: {EarlyStopping (10 epochs), ModelCheckpoint, ReduceLROnPlateau (2 epochs)}
- Hyperparameter set: {lr: 0.0001, dropout layer rate: 0.50, final dense layer units: 128}

[3] DenseNet, Huang et al., 2017

RESULTS





- Accuracy: 90.96%
- Precision: 98.37%
- Recall: 89.44%
- F1-score: 0.94
- MCC: 0.79
- ROC AUC: 98.30%

XCEPTION[4]

- Parameters: ~23M, Top-1 accuracy: 79.0%
- Evolution of InceptionV3, through Depthwise Separable Convolution

Figure 1. A canonical Inception module (Inception V3).

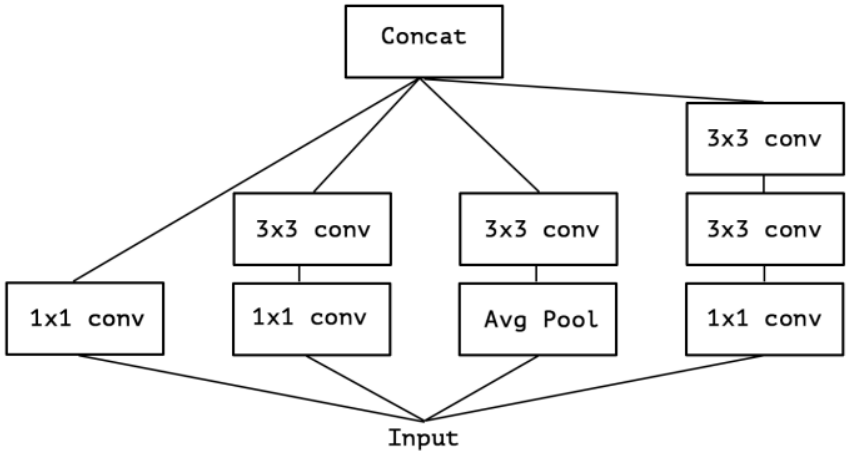
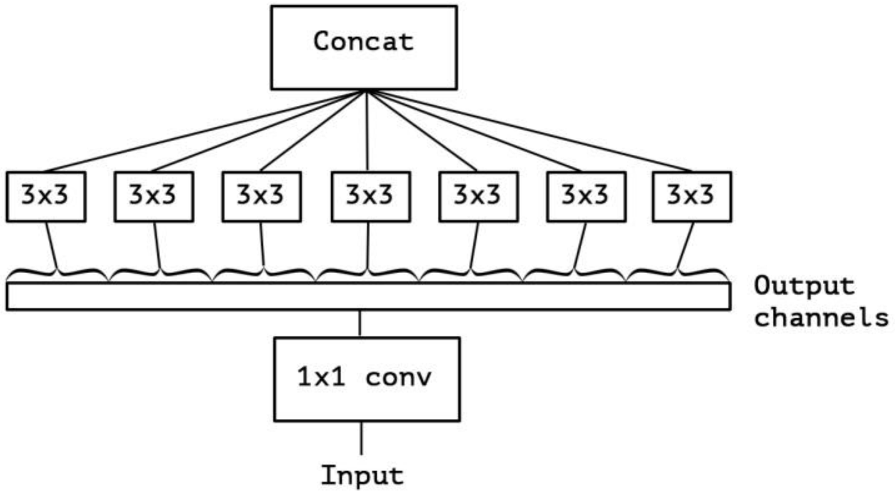


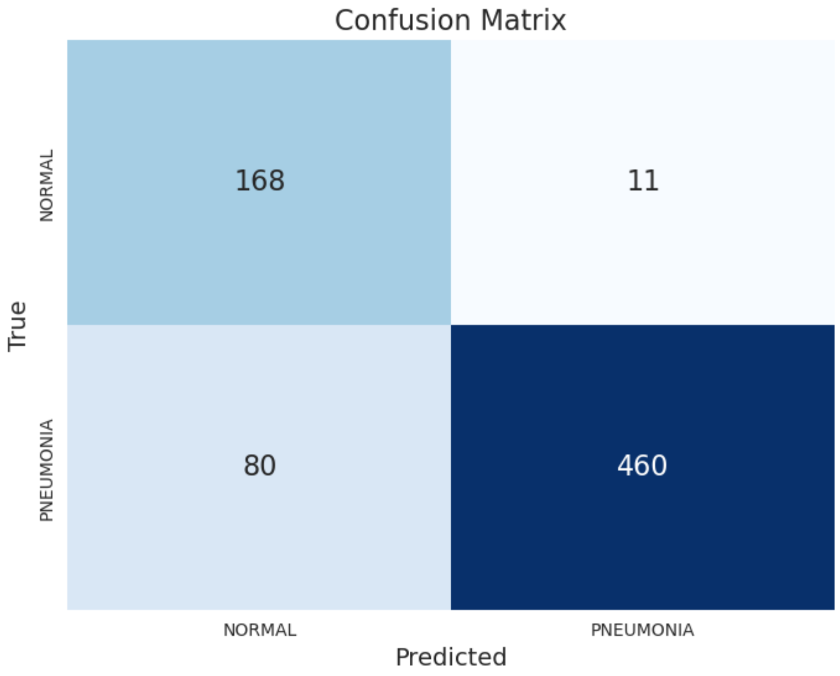
Figure 4. An “extreme” version of our Inception module, with one spatial convolution per output channel of the 1x1 convolution.



- Callbacks: {EarlyStopping (10 epochs), ModelCheckpoint, ReduceLROnPlateu (2 epochs)}
- Hyperparameter set: {lr: 0.0001, dropout layer rate: 0.50, final dense layer units: 128}

[4] *Extreme Inception*, Google Inc., 2017

RESULTS



- Accuracy: 87.34%
- Precision: 97.66%
- Recall: 85.19%
- F1-score: 0.91
- MCC: 0.72
- ROC AUC: 96.68%

HYPERPARAMETERS OPTIMIZATION

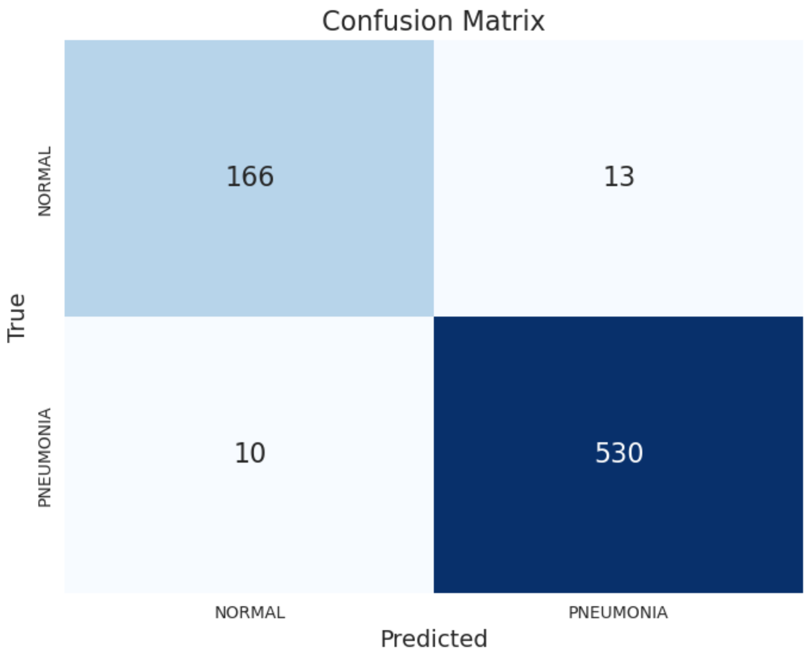
- Define a search space

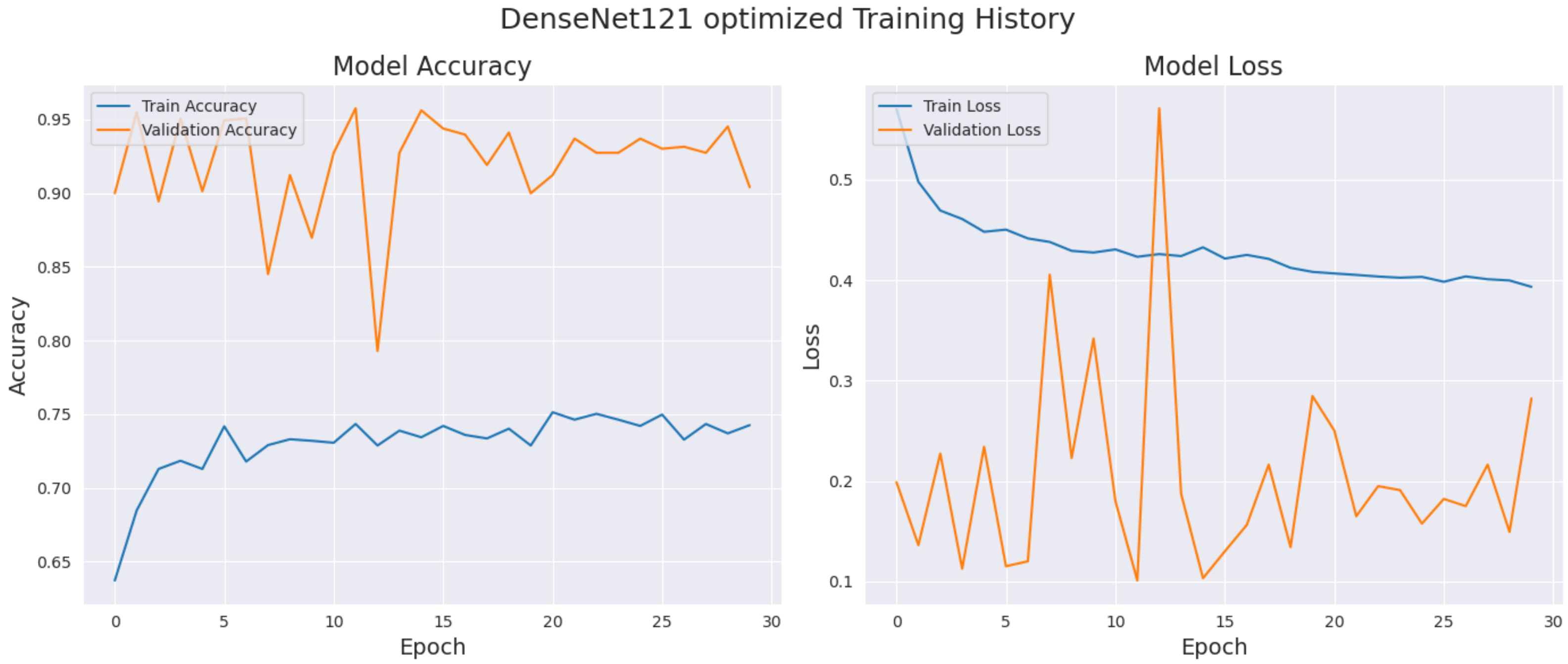
```
space = {
    'lr': hp.choice('lr', [1e-3, 1e-4]),
    'dense_units': hp.choice('dense_units', [64, 128, 256]),
    'dropout': hp.uniform('dropout', 0.25, 0.5),
}
```

- Train the model using different hyperparameters combination (5 epochs)
- Use TPE search to select better combinations
- Select the combination that yields the best performance (i.e. has the best val_acc)

DenseNet121	**Xception**
lr: 0.001	lr: 0.001
Dense units: 128	Dense units: 256
Dropout rate: 0.362	Dropout rate: 0.343

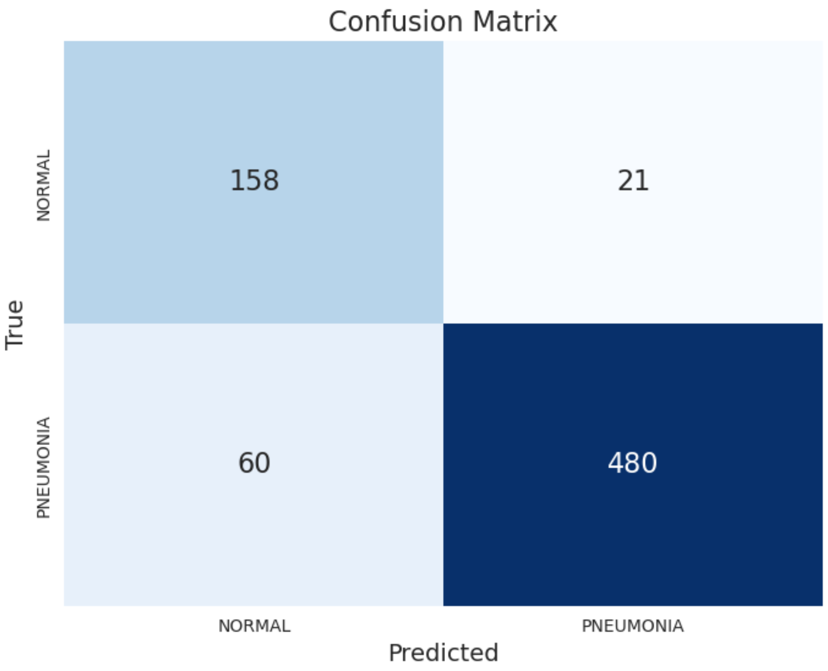
DENSENET121 OPTIMIZED

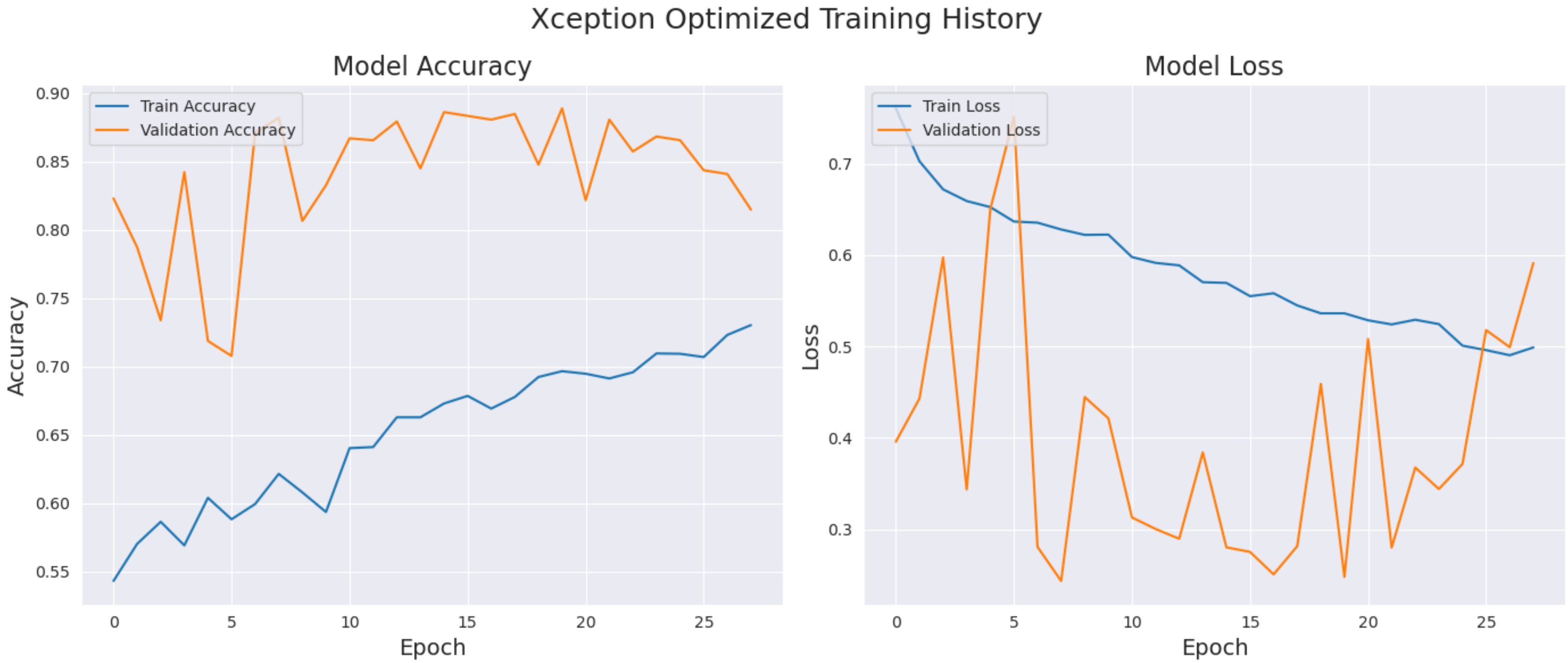




- Accuracy: 96.80%
- Precision: 97.61%
- Recall: 98.15%
- F1-score: 0.98
- MCC: 0.91
- ROC AUC: 99.19%

XCEPTION OPTIMIZED



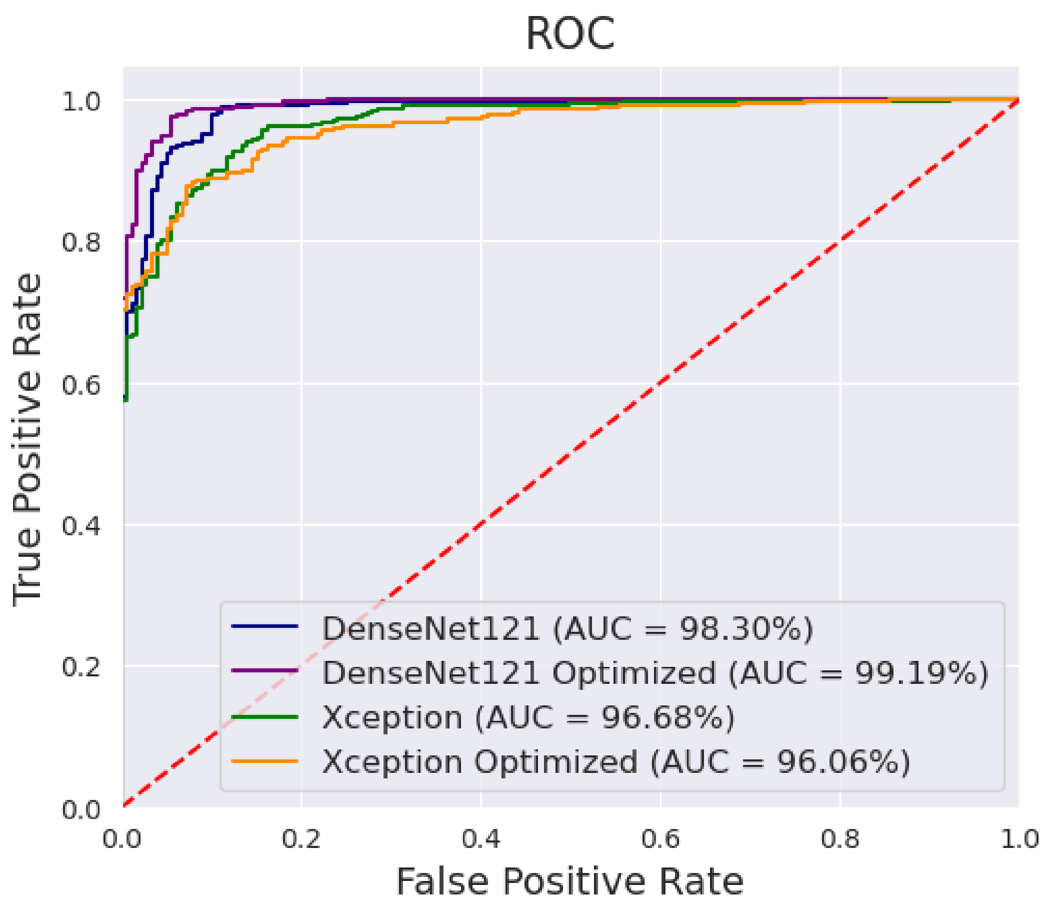


- Accuracy: 88.73%%
- Precision: 95.81%
- Recall: 88.89%
- F1-score: 0.92
- MCC: 0.73
- ROC AUC: 96.06%

CONCLUSION

- Zhang et al. with InceptionV3:
    - Accuracy: 92.8%
    - Recall: 93.2%
    - ROC AUC: 96.8%

- Best model is DenseNet121 optimized:
    - Accuracy: 96.80%
    - Recall: 98.15%
    - ROC AUC: 99.19%



FUTURE DEVELOPMENTS

- Significantly increase the number of training epochs (requires more powerful resources)
- Expand the hyperparameter optimization search space (requires more powerful resources)
- Use versions of DenseNet with more features (e.g., DenseNet169, DenseNet201 - requires more powerful resources)
- Apply the same model to detect whether pneumonia is caused by a virus or by bacteria

THANK YOU FOR LISTENING