PNEUMONIA DETECTION

A Hybrid Ensemble Learning Approach

Machine Learning Project Report

Amine OUBBÉA
Thomas MERCIER
Florian CHOUSTERMAN

Specialization: MedTech & Santé A4

December 2025

Contents

1 Business Scope

1.1 Clinical Context and Problem Definition

Pneumonia remains a critical public health challenge, particularly within pediatric populations where high morbidity and mortality rates persist. According to the World Health Organization (WHO), pneumonia is the single largest infectious cause of death in children worldwide, accounting for 14% of all deaths of children under five years old [1]. As a time-sensitive pathology, untreated bacterial or viral pneumonia can rapidly deteriorate into Acute Respiratory Distress Syndrome (ARDS). In pediatric radiology, clinical presentation is often atypical, making image interpretation significantly more subtle than in adults.

Compounding this challenge is the systemic issue of cognitive overload. The demand for medical imaging has surged by approximately 10% annually, significantly outpacing the growth of the radiologist workforce . This imbalance has led to a "vigilance decrement": after prolonged periods of interpretation, human attention fatigues. Research indicates that the retrospective diagnostic error rate in radiology is estimated at approximately 3% to 5% in daily practice [2], with fatigue being a major contributing factor. Missed diagnoses are therefore an expected byproduct of modern clinical workflows.

1.2 Solution Positioning: The "Safety Net" Approach

To address these challenges, this project develops a Computer-Aided Diagnosis (CADx) system designed strictly as a Clinical Decision Support System (CDSS). The algorithm acts as a triage tool within a human-in-the-loop workflow. By analysing X-rays before human reading, the system functions as a digital safety net that:

- flags high-risk cases so they appear earlier in the radiologist's worklist,
- raises alerts on clearly abnormal exams that might otherwise be overlooked,
- leaves low-risk cases in the standard workflow without blocking human decisions.

The system is therefore intended to support radiologists, not to replace them.

1.3 Strategic Alignment and Data Strategy

Aligned with the MedTech & Santé specialization, our primary objective is patient safety. This directly shapes our data and modelling strategy:

- Cost of False Negative (C_{FN}): a missed diagnosis results in a patient being discharged without treatment, with a risk of rapid deterioration. This cost is critical.
- Cost of False Positive (C_{FP}): a false alarm mainly triggers a secondary review or a confirmatory test. This cost is administrative and comparatively minor.

Given that $C_{FN} \gg C_{FP}$, we adopt a cost-sensitive perspective. Our primary Key Performance Indicator (KPI) is Recall (Sensitivity) on the Pneumonia class. We deliberately accept a moderate increase in false positives if it reduces the probability of discharging a child with undiagnosed pneumonia.

2 Problem Formalisation and Methods

2.1 Clinical and Operational Formulation

From a hospital point of view, our question is simple: when a pediatric chest X-ray arrives in the emergency department, should it be flagged as suspicious for pneumonia and reviewed with priority by a radiologist?

Today, this triage is done manually and under time pressure. Our system is designed as an additional safety layer: for each incoming X-ray, it outputs a risk score between 0 and 1 for pneumonia. This score is not meant to replace the radiologist's decision but to:

- push high-risk exams to the top of the worklist,
- raise an alert when an exam looks clearly abnormal,
- leave low-risk exams in the standard workflow.

Clinically, missing a pneumonia (false negative) is much more serious than asking for one extra check (false positive). This asymmetry, introduced in Section 1, drives all our modelling choices: we systematically favour high Sensitivity (Recall), even if it costs a few additional false alarms.

2.2 Machine Learning Viewpoint

From a machine learning viewpoint, we treat the problem as a supervised **binary classification** task on images: each X-ray is labelled as NORMAL or PNEUMONIA. All our models output a probability score for the PNEUMONIA class, which we convert into a final decision using a threshold (the default value is 0.5 in our experiments for fair comparison between models).

Because the dataset is moderately imbalanced (more pneumonia than normal cases) and because of the clinical context, global Accuracy is not our primary goal. Instead, we:

- use **Recall on the PNEUMONIA class** as the main metric for model selection (safety),
- still report Accuracy, Precision, F1-score and confusion matrices to understand the trade-offs in terms of false positives and false negatives.

2.3 Data Constraints and Assumptions

We work on a public pediatric chest X-ray dataset organised by class (NORMAL vs PNEUMONIA). Images are labelled at the exam level only; no patient identifier is available. We therefore assume that images are independent, even though several exams may belong to the same patient. This is a common simplification for public medical datasets, but it slightly overestimates the diversity of the data.

On the training split, pneumonia represents about 63% of the cases. This prevalence is higher than in a real emergency department, where most children are not sick. As a result, our scores mainly measure how well the models separate the two classes inside this dataset; they cannot be directly interpreted as real-world predictive values without additional calibration and external validation.

The technical details of the train/validation/test split (70/15/15) are described in Section 3.2. The same split is reused for all models to avoid any information leakage.

2.4 Overall Modelling Strategy

We decided not to rely on a single model family. Classical machine learning models are relatively interpretable and fast at inference once they work on compact numerical features, while deep convolutional networks are better at extracting complex visual patterns directly from pixels.

Our strategy is therefore threefold:

1. build a **classical branch** on top of deep visual features (Model A),
2. build a **deep learning branch** that learns directly from images (Model B),
3. combine both branches into a **hybrid ensemble** that acts as a "second opinion" system.

This design mirrors the clinical workflow: two different "readers" look at the same exam in different ways, and their conclusions are aggregated.

2.5 Model A – Logistic Regression on Deep Features

The objective of the classical branch is to have a simple and fast model based on high-level image features, which could be more easily discussed with clinicians than a full black-box network.

Deep feature extraction. Instead of hand-crafting image descriptors, we reuse a ResNet50 network pre-trained on ImageNet as a fixed feature extractor. For each X-ray, we take the output of its last convolutional layers (after global average pooling). This gives a 2048-dimensional vector that summarises the global lung appearance (texture, opacities, shapes).

Dimensionality reduction. Working directly in 2048 dimensions would make classical models heavier, slower to train, and more prone to overfitting on our 619 images. We therefore apply PCA and keep only the components that explain 95% of the variance. In practice, this reduces the feature space to 301 dimensions, i.e. an 85.3% reduction. Figure 1 shows that beyond this point, additional components bring very limited extra information while increasing model complexity.

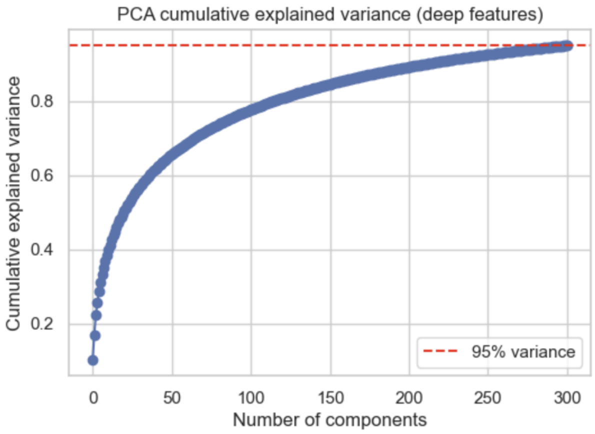

Figure 1: Cumulative explained variance of PCA applied on ResNet50 deep features. The dashed line marks the 95% threshold, reached with 301 components.

Choice of the final classifier. On these PCA features, we tested four scikit-learn models using GridSearchCV: Logistic Regression, Support Vector Machine, Random Forest and Gradient Boosting, with hyperparameters tuned on the combined train+validation set and F1-score as the objective.

On the held-out test set:

- SVM reaches the highest F1-score (≈ 0.95),
- Logistic Regression is very close (F1 ≈ 0.94) with high Recall (≈ 0.93),
- Random Forest achieves perfect Recall but at the cost of many false positives,
- Gradient Boosting lies between these behaviours.

We finally chose **Logistic Regression** as Model A for three main reasons:

- it preserves a high Recall while avoiding the explosion of false positives observed with Random Forest,
- it offers the fastest inference time (around 0.04 ms per sample), which matters for integration in a real-time workflow,
- it remains simple and more interpretable than SVM in this setting.

To account for the class imbalance, `class_weight='balanced'` is used so that errors on the minority class have a higher impact during training.

2.6 Model B – Convolutional Neural Networks on Images

In parallel, we train convolutional networks directly on the images to benefit from end-to-end learning of visual patterns.

Baseline CNN. We first designed a small CNN with only three convolutional blocks, followed by a dense layer and a sigmoid output neuron. The idea was not to beat state-of-the-art performance, but to check that a relatively simple architecture can learn meaningful patterns on this dataset and to observe overfitting behaviour.

This baseline network has about 12.9 million trainable parameters, most of them concentrated in the first dense layer after flattening. With only 619 images, this capacity is quite high, and the learning curves indeed show a small but persistent gap between training and validation performance. The model still reaches good test accuracy, but these curves motivated the use of transfer learning rather than deeper custom architectures.

Transfer learning with ResNet50. To improve robustness, we use ResNet50 pre-trained on ImageNet as a backbone. Chest X-rays in the dataset are grayscale, whereas ResNet50 expects three-channel RGB inputs. We therefore replicate the single channel three times to create pseudo-RGB images and apply the standard ResNet50 preprocessing.

On top of the convolutional base, we add a lightweight classification head (Global Average Pooling, Dropout 0.5, Dense sigmoid). Most layers of ResNet50 are frozen and only the last residual blocks, together with the head, are fine-tuned on our data. This keeps roughly 4.5 million parameters trainable out of about 23.6 million in total, which is a better match for our dataset size.

We also apply data augmentation (random rotations, shifts, zoom, horizontal flips) and callbacks (EarlyStopping and ReduceLROnPlateau) to further reduce overfitting. In practice, this transfer learning model achieves the best standalone performance among our deep architectures and provides the probability scores used as Model B in the ensemble.

2.7 Hybrid Ensemble

Relying on a single model can still lead to critical errors on atypical cases. To reduce this risk, we combine the two best models:

- Model A: Logistic Regression on PCA-compressed deep features,
- Model B: ResNet50-based transfer learning model.

For each exam, both models output a probability of pneumonia. The hybrid ensemble simply takes the average of these two probabilities and applies the same decision threshold as before. We chose this **soft voting** strategy with equal weights because:

- it is easy to implement and explain,
- both models have similar overall F1-scores but different error patterns, so averaging helps smooth out individual mistakes,
- it avoids introducing extra hyperparameters (weights, new thresholds) that could overfit our relatively small test set.

In our experiments, this ensemble slightly increases Recall on the PNEUMONIA class while keeping high overall Accuracy. The exact gain is quantified in the Results section.

2.8 Limitations

Even with this hybrid design, several limitations remain:

- The ResNet-based component is still a black box. We do not yet include explainability tools (such as Grad-CAM) to highlight which lung regions drive the predictions.
- A few pneumonia cases are still misclassified, especially borderline or low-quality exams. With more diverse data, these cases could be better represented during training.
- The decision threshold is fixed at 0.5 for all models to make comparisons fair. In a real deployment, this threshold should be calibrated together with clinicians, depending on how much Recall they are willing to trade for fewer false positives.

3 Methodology

3.1 Data Description and Exploration

We use the public *Chest X-Ray Images (Pneumonia)* dataset, which contains pediatric chest radiographs labelled as NORMAL or PNEUMONIA. In its raw form, the dataset is organised in two folders (`NORMAL/` and `PNEUMONIA/`), without predefined train/validation/test splits.

As a first step, we counted the number of images in each class and plotted the global distribution (Figure 2). The dataset contains 619 pediatric chest X-ray images in total: 230 NORMAL and 389 PNEUMONIA. This corresponds to 37.2% Normal and 62.8% Pneumonia cases, which confirms a clear class imbalance that must be handled during modelling.

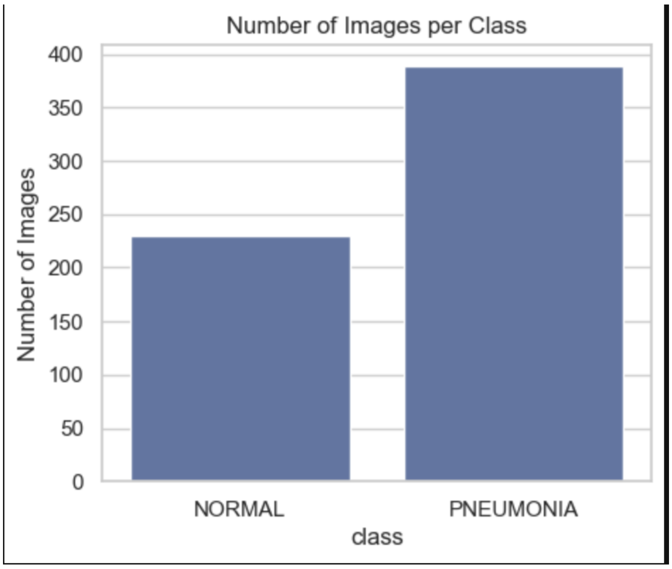

Figure 2: Number of images per class in the full dataset.

To check that the labels and images were consistent with radiological expectations, we:

- visually inspected a sample of NORMAL and PNEUMONIA images,
- analysed the distribution of the average pixel intensity per image.

On average, PNEUMONIA images show higher mean pixel intensity (more opaque lungs) than NORMAL images, which matches the expected pattern of lung opacities in pneumonia. This gave us confidence that the dataset could be used as a reasonable training base despite its simplicity.

3.1.1 Missing Values and Corrupted Files

Since the data consist of image files, the main risk is corrupted or unreadable images rather than missing values in a table. We therefore attempted to open every file using the PIL library. No image failed to load, so we did not need to discard any samples for integrity reasons.

3.1.2 Imbalanced Data

On the training split (defined below), we counted 161 NORMAL and 272 PNEUMONIA images, i.e. roughly 37% vs 63% and an imbalance factor of about 1:1.69 in favour of PNEUMONIA. A naive classifier biased towards the majority class could therefore obtain a high global Accuracy while still missing many NORMAL cases.

Because our clinical objective is to reduce missed pneumonia cases, we chose to handle this imbalance in two ways:

- we enforced `class_weight='balanced'` in Logistic Regression, SVM and Random Forest, so that errors on the minority class have a stronger impact during training;
- we used Recall on the PNEUMONIA class as a primary selection criterion when comparing models, rather than Accuracy alone.

We deliberately did not use oversampling or synthetic data generation, to keep the classical pipeline simple and avoid introducing additional sources of bias on such a small dataset.

3.1.3 Visual Inspection and Outliers

During exploration and later in the error analysis, we identified several types of "difficult" images:

- low-contrast exams,
- images with medical devices or artefacts,
- borderline cases with very subtle opacities.

We decided to keep these cases in both training and test sets. Excluding them would have made the problem artificially easier, whereas in a real emergency department such noisy or atypical exams are common. Instead, we discuss these cases explicitly when analysing false negatives and false positives in the Results section.

3.2 Data Splitting for Train / Validation / Test

The original dataset does not provide a train/validation/test split. To obtain reproducible and fair comparisons between models, we created our own split with three objectives:

- enough training data to fit deep models,
- a separate validation set for hyperparameter tuning and early stopping,
- an untouched test set for the final evaluation.

All file paths and labels (NORMAL or PNEUMONIA) were first collected into a `pandas` DataFrame. We then applied a stratified split using `train_test_split` with a fixed random seed (`random_state=42`):

- 70% of the data for training,
- 15% for validation,
- 15% for testing.

The final split is:

- Train set: 161 NORMAL and 272 PNEUMONIA images,

- Validation set: 34 NORMAL and 59 PNEUMONIA images,
- Test set: 35 NORMAL and 58 PNEUMONIA images.

This distribution is illustrated in Figure 3.

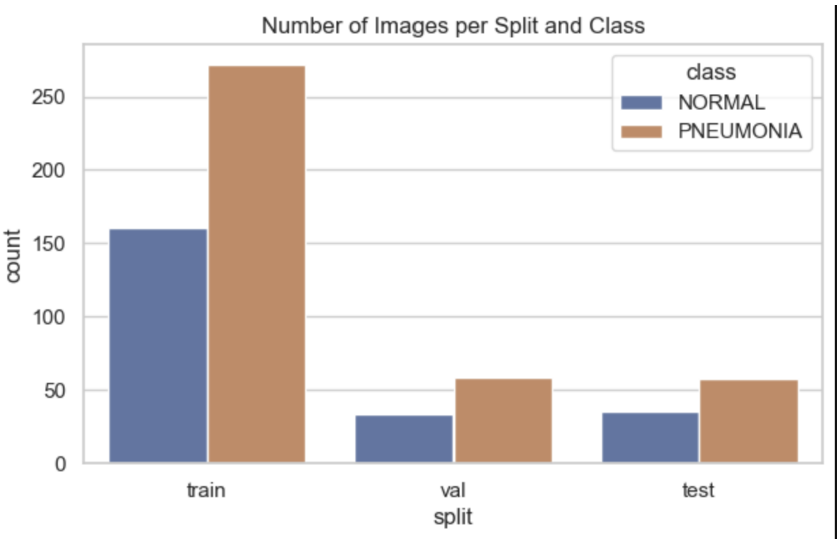

Figure 3: Number of images per split (train/validation/test) and per class.

Stratification ensures that the proportion of NORMAL vs PNEUMONIA is preserved in each subset. We then mirrored this split in the directory structure used by the Keras generators:

- `dataset/train/NORMAL`, `dataset/train/PNEUMONIA`
- `dataset/val/NORMAL`, `dataset/val/PNEUMONIA`
- `dataset/test/NORMAL`, `dataset/test/PNEUMONIA`

The same split is reused consistently for all models (classical branch, baseline CNN and ResNet50 transfer learning). This was a deliberate choice to avoid any information leakage and to make performance comparisons directly meaningful.

3.3 Algorithm Implementation and Hyperparameters

In this section we describe how the different models were implemented in practice, and which design choices were made in terms of preprocessing and hyperparameters.

Preprocessing and Data Loaders

All models operate on images resized to 224 × 224 pixels. This resolution is a standard input size for ResNet50 and provides a good compromise between preserving anatomical details and limiting computation time.

Although the original chest X-rays are grayscale, images are loaded as three-channel RGB tensors. This "channel replication" allows us to reuse ImageNet pre-trained weights in ResNet50 without modifying the architecture.

We use two slightly different preprocessing pipelines:

- For the baseline CNN, we use an `ImageDataGenerator` with `rescale=1./255`, which simply normalises pixel values to $[0, 1]$. This is enough for a model trained from scratch.

- For ResNet50 (feature extraction and transfer learning), we use a dedicated `ImageDataGenerator` with the Keras `preprocess_input` function, to match the preprocessing used during ImageNet training.

All generators use a batch size of 32. Training generators shuffle the data at each epoch to reduce the risk of learning artefacts from the order of images, while validation and test generators do not shuffle, to keep evaluation deterministic.

Classical Machine Learning Branch

The classical branch (Model A) is built on top of deep features extracted by a pre-trained ResNet50. Our goal is to obtain a simple, fast model that still benefits from powerful visual representations.

The implementation steps are:

1. Use ResNet50 with `include_top=False` and `pooling='avg'` to extract one 2048-dimensional feature vector per image for the train, validation and test sets.

2. Merge train and validation features into a single cross-validation set $(X_{\text{cv}}, y_{\text{cv}})$ to make better use of the limited data when searching hyperparameters.

3. Standardise features with `StandardScaler`, which is important for models such as Logistic Regression and SVM.

4. Apply PCA and keep enough components to retain 95% of the variance. This reduces the feature space from 2048 dimensions to 301 components (an 85.3% reduction) and makes the classical models less prone to overfitting.

5. Train several scikit-learn classifiers (Logistic Regression, Random Forest, Gradient Boosting, SVM) using `GridSearchCV` with 5-fold cross-validation, optimising the F1-score to balance Precision and Recall.

Because the training set is imbalanced, we set `class_weight='balanced'` in Logistic Regression, SVM and Random Forest. This forces the models to pay more attention to the minority class during training.

On the validation folds and on the held-out test set, Logistic Regression offers the best compromise between Recall, number of false positives, inference time and model simplicity. It is therefore selected as Model A, with `max_iter=500` and `class_weight='balanced'`.

Baseline CNN

The baseline CNN is implemented with Keras as a small, custom architecture:

- three convolutional blocks (Conv2D + ReLU + MaxPooling),

- a Flatten layer,

- a Dense layer with 128 units and ReLU,

- an output Dense layer with 1 sigmoid unit for binary classification.

We chose this architecture because it is easy to implement and train on a laptop GPU, and it provides a reference point to judge the benefit of transfer learning.

Training details are:

- Optimiser: Adam with a learning rate of 10^{-4} (stable training on this dataset).
- Loss: binary cross-entropy.
- Metric monitored during training: Accuracy (other metrics are computed on the test set afterwards).
- Batch size: 32.
- Maximum epochs: 20, with an `EarlyStopping` callback on validation loss (`patience=5`, `restore_best_weights=True`) to stop training as soon as overfitting appears.

This network has about 12.9 million trainable parameters, the vast majority being in the first dense layer after flattening. With only 619 images, this capacity is high, and the learning curves indeed show a gap between training and validation performance. For this reason, we mainly use this model as a baseline rather than as our final choice.

Advanced CNN with ResNet50 Transfer Learning

The advanced CNN (Model B) uses ResNet50 pre-trained on ImageNet as a backbone. This choice is motivated by:

- the limited size of our dataset, which makes training a deep network from scratch risky in terms of overfitting,
- the strong performance of residual architectures on medical imaging in the literature.

Technically, we configure ResNet50 with `include_top=False` and add a small classification head:

- GlobalAveragePooling2D,
- Dropout layer with rate 0.5 for regularisation,
- Dense layer with 1 sigmoid unit.

Most layers of ResNet50 are frozen, and only the last residual blocks plus the classification head are fine-tuned on our chest X-ray dataset. In total, the model contains about 23.6 million parameters, but only roughly 4.5 million are trainable. This drastically limits the number of parameters that must adapt to our data and helps control overfitting.

We use the following training setup:

- Optimiser: Adam with learning rate 10^{-4}.
- Loss: binary cross-entropy.
- Batch size: 32.

- Data augmentation on the training set (random rotations, shifts, zoom and horizontal flips) to increase robustness to acquisition variability.
- `EarlyStopping` on validation loss with `patience=5` and `restore_best_weights=True`.
- `ReduceLROnPlateau` to reduce the learning rate when the validation loss stops improving.

This transfer learning model achieves the best standalone performance among our deep architectures and provides the probability scores $P_{\text{ResNet}}(y = 1 \mid X)$ used in the hybrid ensemble.

Hybrid Ensemble Implementation

Finally, we combine Model A (Logistic Regression) and Model B (ResNet50 transfer learning) into a hybrid ensemble.

For each image in the test set, we compute:

- the probability of pneumonia given by Model A, obtained from `predict_proba` on the Logistic Regression classifier,
- the probability of pneumonia given by Model B, obtained from the sigmoid output of the ResNet50-based network.

The ensemble score is defined as the simple average of these two probabilities, and we apply the same decision threshold (0.5) as for individual models. We chose equal weights and a simple averaging rule to:

- keep the method easy to interpret and implement,
- avoid introducing additional hyperparameters (such as learned weights) that could overfit a small test set,
- benefit from the complementary error patterns of the classical and deep branches.

All metrics reported for the ensemble in the Results section are computed from these averaged predictions.

4 Results

4.1 Evaluation Protocol and Metrics

All models are evaluated on the held-out test set (35 NORMAL, 58 PNEUMONIA), which is only used once at the very end. To stay aligned with the clinical objective (avoid missed pneumonias), we focus on:

- **Recall on the PNEUMONIA class** (Sensitivity): proportion of sick patients correctly detected. This is our main “safety” metric, as False Negatives correspond to missed pneumonias.
- **Accuracy** : overall proportion of correctly classified exams (NORMAL and PNEUMONIA combined).

- **Precision on PNEUMONIA** : proportion of predicted pneumonias that are actually sick (controls the number of unnecessary alerts).
- **F1-score** : harmonic mean between Precision and Recall, used mainly to compare classical models during model selection.
- **Confusion matrix** : qualitative view of False Positives and False Negatives for the main models.
- **Inference time per exam** : useful to judge if the model can realistically be used in a triage setting.

4.2 Classical Models on Deep Features

Table 1 summarises the performance of the four classical models trained on PCA-compressed deep features.

Table 1: Test performance of classical ML models (deep features + PCA).

Model	Accuracy	Precision	Recall	F1-score	Time (ms/sample)
Logistic Regression	92.5%	94.7%	93.1%	93.9%	**0.04**
SVM	**93.5%**	94.8%	94.8%	**94.8%**	0.19
Random Forest	79.6%	75.3%	**100%**	85.9%	0.37
Gradient Boosting	91.4%	90.3%	96.6%	93.3%	0.06

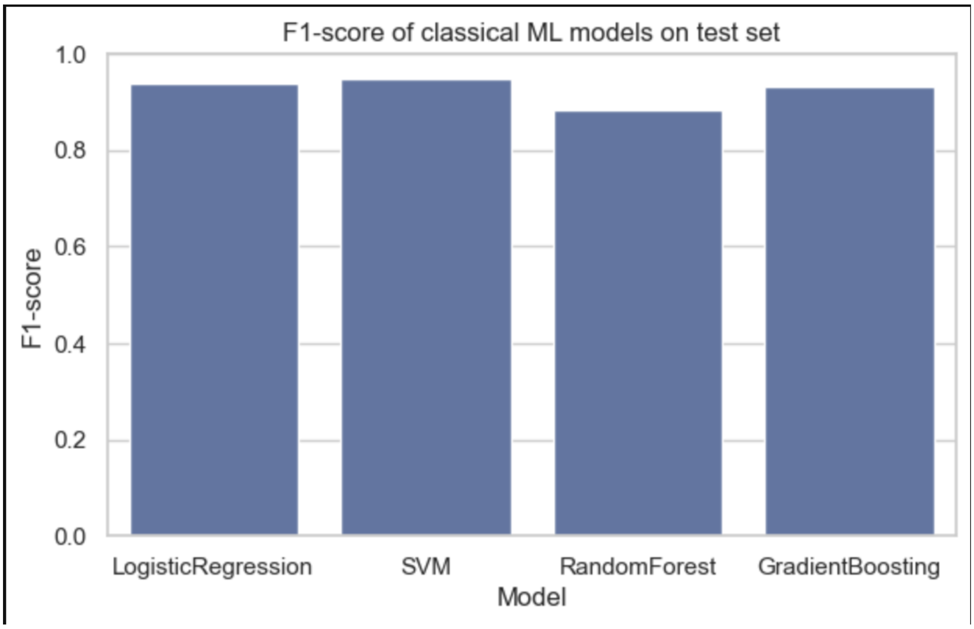

Figure 4: F1-score of classical ML models on the test set.

The SVM obtains the best F1-score and Recall, but Logistic Regression is extremely close while being much faster and easier to interpret. Random Forest reaches perfect Recall but at the cost of many False Positives (19 healthy patients out of 35 are incorrectly flagged as PNEUMONIA), which would generate too many unnecessary alerts in practice.

Given this trade-off between safety, practicality and interpretability, we select **Logistic Regression** as our reference classical model (Model A). It offers high Recall on PNEUMONIA with almost no computational cost, which is attractive for a triage system that may need to process large daily volumes of exams.

4.3 Baseline CNN

The baseline CNN trained directly on images reaches the following test metrics:

- Accuracy: 88.2%
- Precision (PNEUMONIA): 92.7%
- Recall (PNEUMONIA): 87.9%
- F1-score: 90.3%

It correctly identifies most pneumonias but remains slightly below the best classical models in terms of Recall. The confusion matrix shows 7 False Negatives (PNEUMONIA predicted as NORMAL) and 4 False Positives.

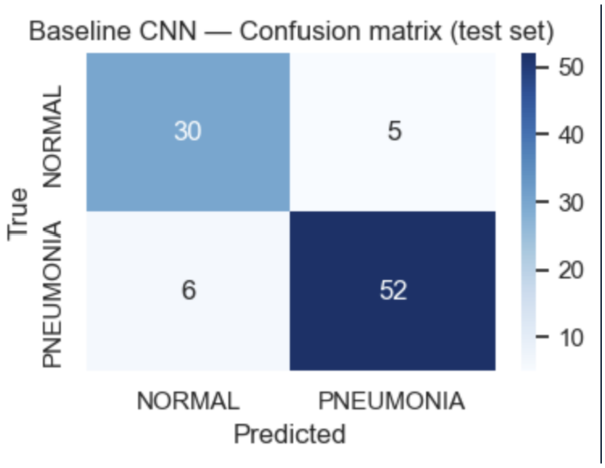

Figure 5: Confusion matrix CNN Baseline

The learning curves in Figure 6 show a small but visible gap between training and validation performance: training Accuracy keeps increasing while validation Accuracy stabilises, and validation Loss stops decreasing after a few epochs. This confirms that, given the limited dataset size, the baseline CNN is slightly over-parameterised (about 12.9 million trainable parameters) and tends to overfit despite EarlyStopping.

In practice, this model mainly plays the role of a *deep baseline*: it validates that a simple convolutional architecture can learn meaningful patterns, but it is not the best candidate for deployment.

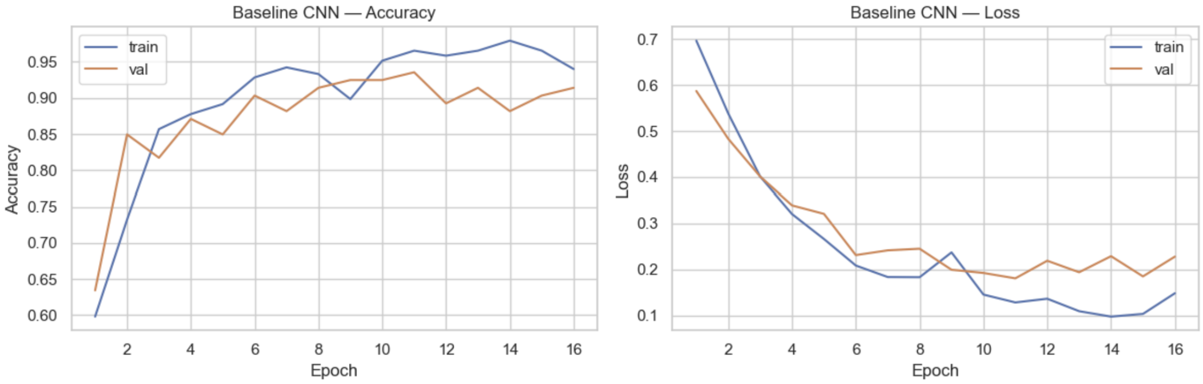

Figure 6: Baseline CNN: training and validation curves (Accuracy and Loss).

4.4 Advanced CNN with ResNet50 Transfer Learning

The transfer learning model based on ResNet50 (Model B) clearly improves over the baseline CNN:

- Accuracy: 92.5%
- Precision (PNEUMONIA): 96.4%
- Recall (PNEUMONIA): 91.4%
- F1-score: 93.8%
- Inference time: ≈ 90.6 ms per exam

On the test set, it misclassifies 5 PNEUMONIA cases as NORMAL (False Negatives) and 2 NORMAL cases as PNEUMONIA (False Positives).

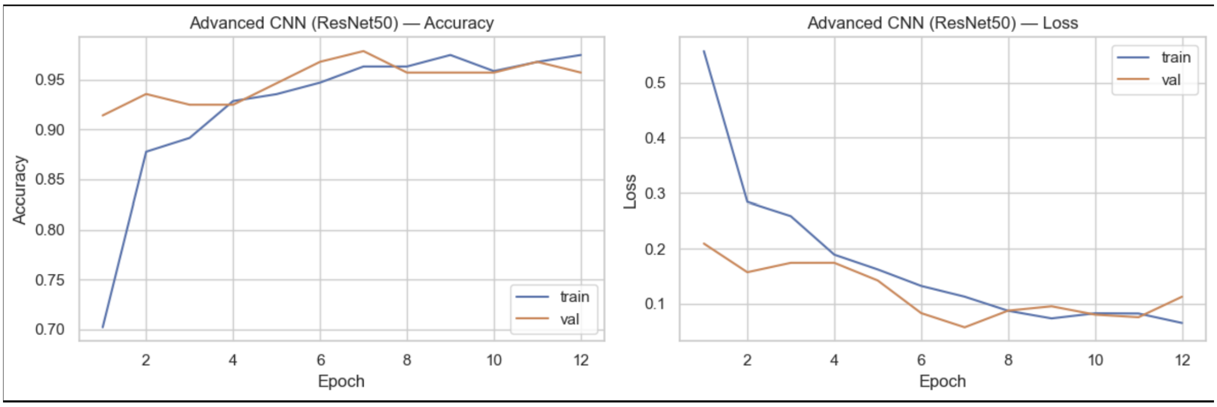

Figure 7: Advanced CNN (ResNet50): training and validation curves (Accuracy and Loss).

The learning curves in Figure 7 show very high training and validation Accuracy (around 97–99%), and validation Loss decreasing steadily with only a light increase at the end. This indicates that data augmentation, partial freezing of the backbone and learning-rate scheduling are effective in limiting overfitting, despite the model complexity (about 4.5 million trainable parameters on top of frozen ImageNet features).

In summary, transfer learning allows us to recover the performance of the best classical models while working directly at the pixel level, at the price of a significantly higher inference time.

4.5 Hybrid Ensemble: Impact on Patient Safety

To combine the strengths of both worlds, we build a hybrid ensemble that averages the probabilities of:

- Model A: Logistic Regression on PCA-compressed deep features,
- Model B: ResNet50-based transfer learning CNN.

On the test set, the ensemble achieves:

- **Accuracy**: 94.6% (88/93 exams correctly classified)
- **Recall (PNEUMONIA)**: **96.6%** (56/58 cases correctly detected)
- **Precision (PNEUMONIA)**: 94.9% (56/59 predicted PNEUMONIA exams were true positives)

The corresponding confusion matrices for the three main models (Logistic Regression, ResNet50, Ensemble) are shown in Figure 8.

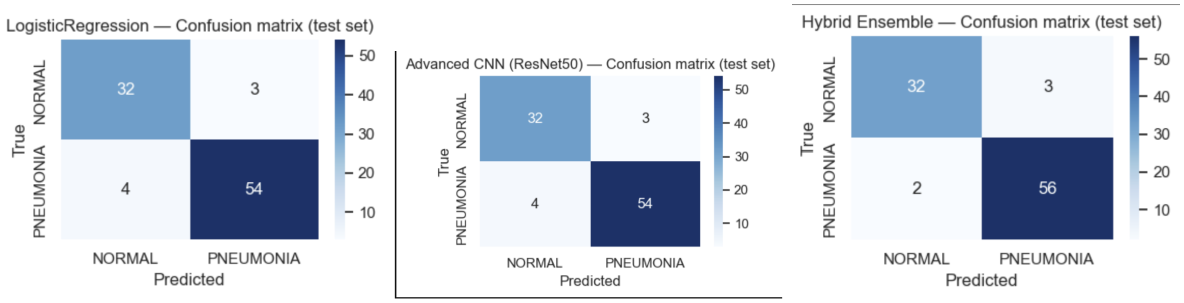

Figure 8: Confusion matrices on the test set for Logistic Regression (left), ResNet50 (centre) and the hybrid ensemble (right).

For the ensemble alone, the confusion matrix is:

$$\begin{pmatrix} 32 & 3 \\ 2 & 56 \end{pmatrix}$$

Only **2** pneumonias are missed (False Negatives) and just **3** healthy patients are over-alerted (False Positives).

Figure 9 summarises the evolution of Recall for the three approaches.

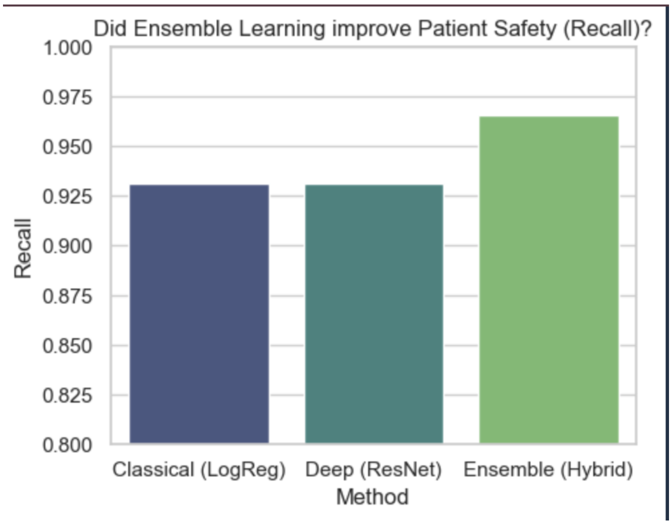

Figure 9: Recall on PNEUMONIA for the classical model (Logistic Regression), the deep model (ResNet50) and the hybrid ensemble.

Compared to the individual models:

- Recall increases from 93.1% with Logistic Regression and ResNet50 to 96.6% with the ensemble;
- Accuracy remains high and very close to the best single model;
- the number of False Negatives is reduced without exploding the number of False Positives.

4.6 Discussion of Errors and Trade-offs

From a clinical perspective, the most critical errors are the remaining False Negatives. Manual inspection of these cases (see Error Analysis section) suggests that they correspond either to borderline opacities, low image quality or atypical presentations. In other words, they are also challenging for a human reader.

Overall, the experiments confirm our modelling choices:

- Classical models on deep features are very competitive and extremely fast; this justifies using Logistic Regression as a lightweight but strong first opinion.
- Transfer learning with ResNet50 improves the deep baseline and brings an independent, non-linear view directly from the images.
- The hybrid ensemble slightly improves Recall and reduces critical errors, which is consistent with our “safety net” positioning of the system.

While the test set is relatively small and results should be interpreted with caution, the combination of high Recall, limited False Positives and reasonable inference time suggests that this approach could realistically support radiologists in prioritising pediatric chest X-rays.

5 Conclusion and Perspectives

5.1 How We Tackled the Clinical Business Case

The initial question was not "can we get a high accuracy on a public dataset?", but rather: *can we build a decision support tool that helps avoid missed cases of pediatric pneumonia in an emergency department, under time pressure?*

This clinical framing guided all our modelling choices:

- We treated the task as a **triage** problem: the model assigns a risk score to each chest X-ray so that suspicious exams can be read first.
- Because missing a sick child is more serious than raising an extra alert, we used **Recall on the Pneumonia class** as the primary metric and systematically analysed confusion matrices rather than Accuracy alone.
- We audited the dataset distribution and explicitly handled the class imbalance (1 : 1.69 in favour of Pneumonia) with `class_weight='balanced'` in our classical models.
- We adopted a **two-branch strategy**: a classical model (Logistic Regression on ResNet50 deep features + PCA) and an advanced CNN (ResNet50 fine-tuned on our data), then combined them in a hybrid ensemble.

On the held-out test set, both branches individually reached strong performance. The Logistic Regression model achieved a Recall of 93.1 % (54/58 pneumonias detected), while the fine-tuned ResNet50 reached the same Recall of 93.1 %. The hybrid ensemble, which averages their probability outputs, further reduced the number of missed cases: it correctly identifies **56 out of 58** pneumonias (Recall $\approx$ 96.6 %) with an Accuracy of 94.6 %.

In concrete terms, on our test cohort this means:

- moving from 4–5 missed pneumonias with the individual models to only **2** False Negatives with the ensemble;
- keeping the number of unnecessary alerts low (only **3** False Positives among 35 healthy children).

Given the business case, this trade-off is clinically sensible: we accept a slightly more complex model and a higher inference time to further reduce the risk of sending a sick child home undiagnosed, without overwhelming clinicians with false alarms.

5.2 From Notebook to Hospital: What Would Change in Practice?

Despite these encouraging numbers, our system remains a prototype trained on a public, curated dataset. Several additional steps would be required before any real-world deployment:

- **Integration into the workflow.** The risk score should appear directly in the radiologist's worklist within the RIS/PACS, for example as a coloured flag or an additional column. A tool that lives outside the existing workflow is unlikely to be used consistently.

- **Prospective validation.** Our evaluation is retrospective and the images are relatively homogeneous. In reality, X-rays come from multiple machines, protocols and hospitals. A prospective study over several months would be necessary to measure the true impact on missed diagnoses and reading time.
- **Human–AI collaboration.** The model is meant to act as a second reader, not as a replacement. One practical scenario would be that exams flagged as "high risk" must be reviewed by a senior radiologist before a child is discharged.

Past experience with Computer-Aided Detection in mammography has shown that good ROC curves are not enough: when CAD marks were used without clear guidelines, they increased sensitivity but also false positives and unnecessary biopsies. This illustrates that *how* a tool is integrated into clinical practice can be as important as its raw performance.

5.3 What We Would Add With More Time and Resources

With additional data, time and computing power, several extensions would be natural:

- **Explainability.** Integrating Grad-CAM or similar techniques to highlight which lung regions drive the decision. This would help detect failure modes (e.g. when the network focuses on the diaphragm or on text markers instead of the lungs) and increase clinicians' trust.
- **Larger and more diverse datasets.** Training and testing on multi-centre data with different age groups and acquisition devices would allow us to measure robustness and reduce the risk of centre-specific overfitting.
- **Calibrated thresholds by scenario.** Rather than a fixed threshold at 0.5, the decision threshold could be tuned together with clinicians for different contexts (night shifts, winter epidemics, intensive care), where the acceptable balance between sensitivity and specificity changes.
- **Additional clinical inputs.** Combining the X-ray with simple clinical variables (age, fever, oxygen saturation) in a multimodal model would better approximate how radiologists actually reason and could further reduce False Negatives.
- **Monitoring in production.** If deployed, the system would need a monitoring dashboard tracking performance over time (drift, changes in case-mix, increase in false positives), with a mechanism for periodic re-training when performance degrades.

5.4 Limitations and Lessons Learned

Several limitations of our work are important to acknowledge:

- the dataset is small and does not reflect the true prevalence of pneumonia in the general pediatric population;
- multiple images may come from the same patient, but we treated them as independent samples;

- we did not include any explainability module, and the ResNet component largely remains a black box;
- our evaluation uses only image data and ignores the clinical context in which X-rays are normally interpreted.

Even with these limitations, the project allowed us to implement a complete end-to-end pipeline: from dataset audit and splitting, to feature extraction, model selection, transfer learning and ensemble design under a clinically motivated cost constraint. The main lesson is that good metrics are only one part of the story. Equally important are the modelling choices that reflect real clinical priorities (Recall vs. Precision) and the way such a system would actually fit into the hospital workflow.

In that sense, this project is less an end point than a starting point towards safer and more transparent AI-assisted radiology for pediatric pneumonia.

References

[1] World Health Organization. *Pneumonia.* Fact sheet, 2019. Available at `https://www.who.int/news-room/fact-sheets/detail/pneumonia`.

[2] A. P. Brady. Error and discrepancy in radiology: inevitable or avoidable? *Insights into Imaging*, 3(3):227–238, 2012.

[3] D. S. Kermany, M. Goldbaum, W. Cai, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131.e9, 2018.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] R. R. Selvaraju, M. Cogswell, A. Das, et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.