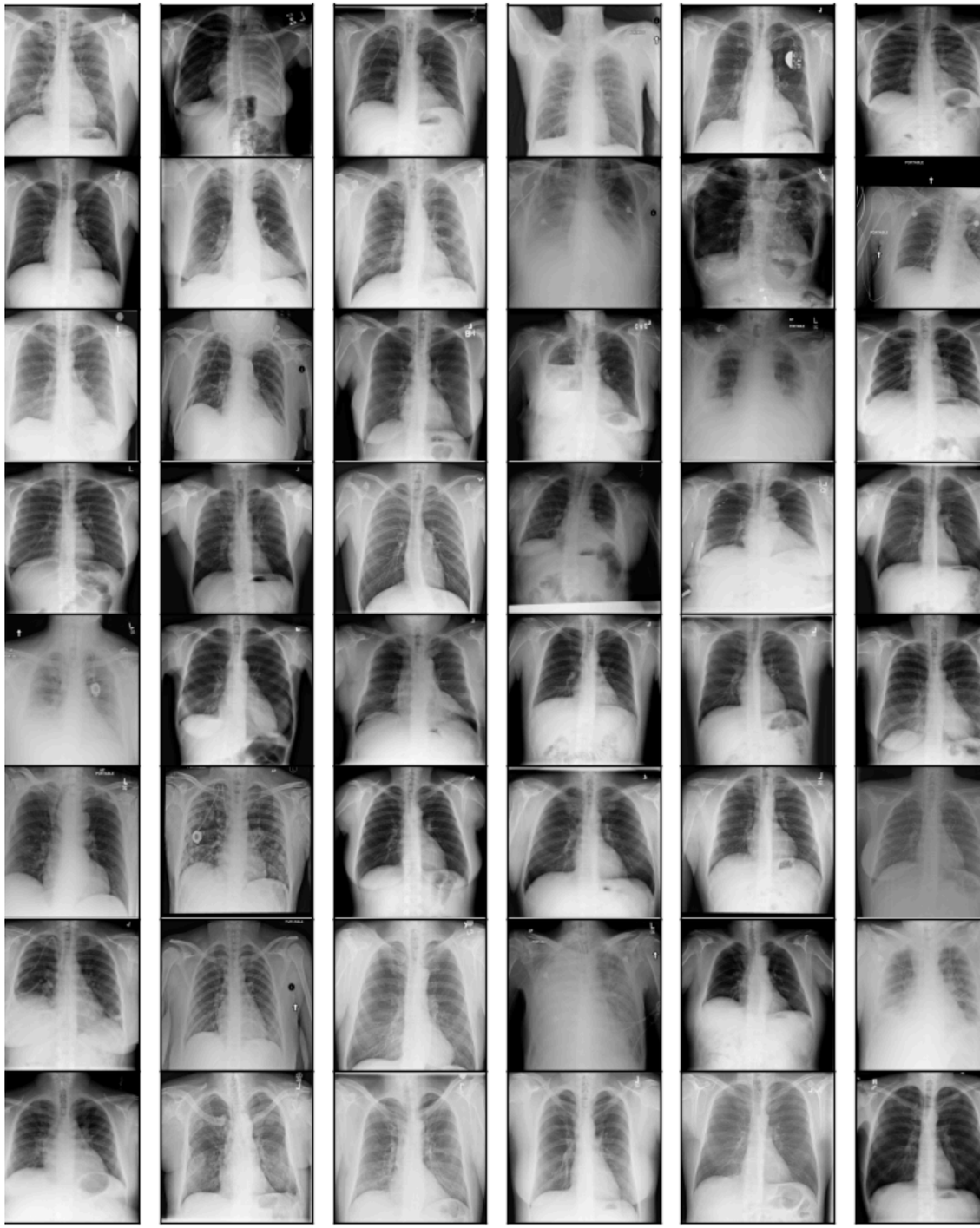
COMPUTATIONAL VISION MODEL FOR MULTI-LABEL CLASSIFICATION FOR CHEST X-RAY DATASET - DENSENET

JOSÉ MANUEL ENRÍQUEZ RODRÍGUEZ 2132982
CARLOS EDUARDO GARCÍA HERRERA 2133000
MARTIN ALEXIS MARTÍNEZ ANDRADE 2049334
DANIEL ROJAS VILLAREAL 2132983
DIEGO ADRIÁN MORENO DUARTE 2132863

INTRODUCTION

This project proposes a model based on DenseNet121 for the multi-label classification of thoracic diseases from chest X-rays in the NIH Chest X-Ray dataset. The architecture is complemented with a Feature Pyramid Network (FPN) to capture multi-scale information and a custom Classification Wrapper to adapt the model's output to the multi-label structure of the problem.

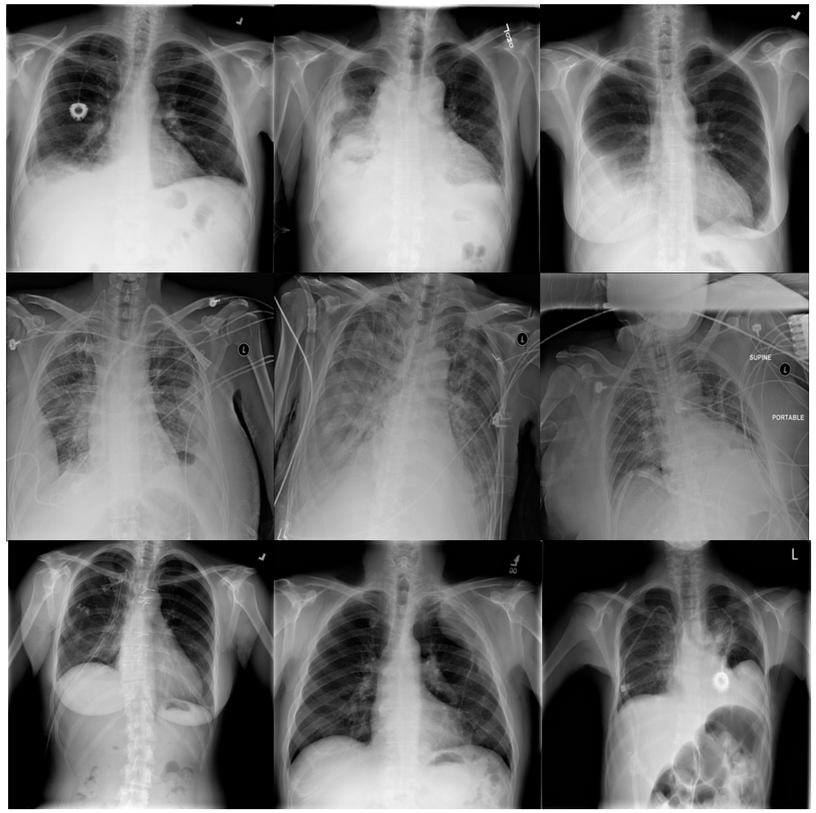


EXPECTED RESULTS

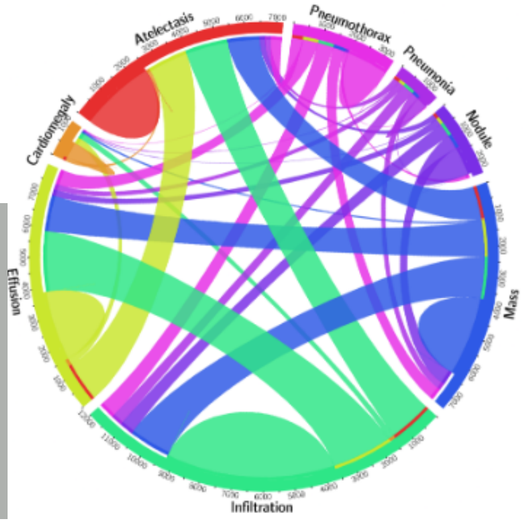
The main objective of this project is to develop and train a multi-label classification model using DenseNet121 as the primary backbone for feature extraction, aimed at detecting multiple thoracic pathologies from chest X-rays in the NIH Chest X-Ray dataset.

DATASET DESCRIPTION

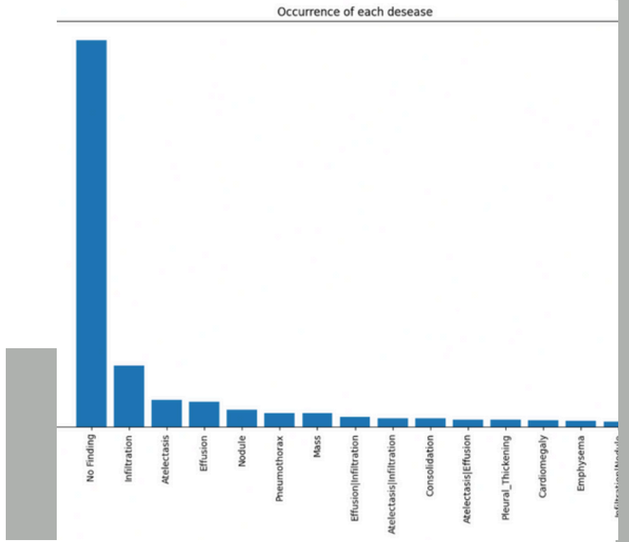
The ChestX-ray8 dataset contains 112,120 frontal chest X-ray images from 30,805 unique patients.



Each record in the dataset can be classified into the following diseases:
Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural Thickening, Cardiomegaly, Nodule, Mass, Hernia, No Findings



The following graph shows the frequency of diseases in the dataset.



DATASET

DEVELOPMENT ENVIRONMENT

The selected execution environment was Google Colab, as it offers high-performance resources for training Machine Learning models, both in its free version and in the Pro version. A Google Colab Pro subscription was acquired to access more powerful hardware and benefit from longer execution

Especifications:

CPU: Intel(R) Xeon(R) CPU @ 2.20GHz (6 cores / 12 threads)

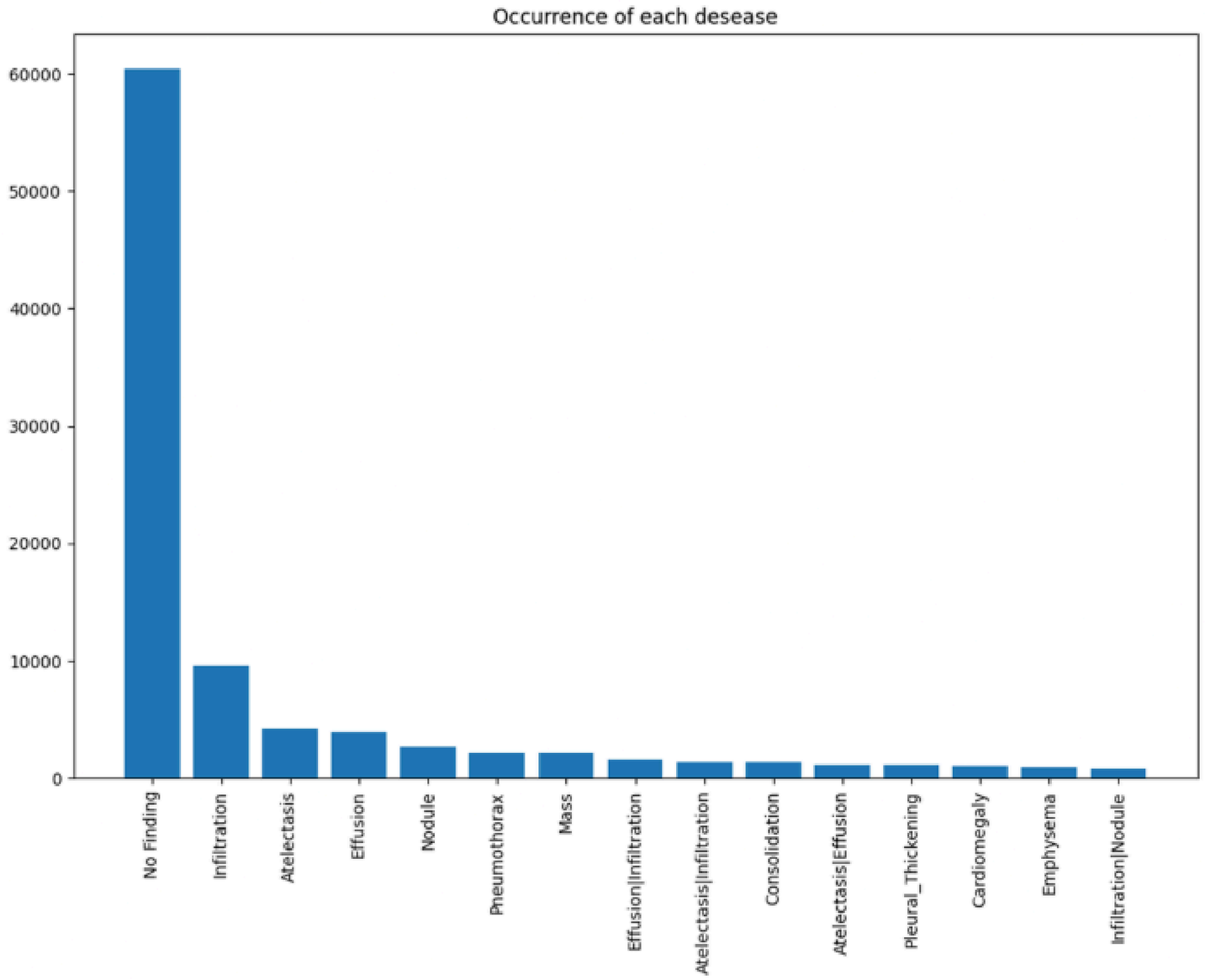
RAM: 53 GB

GPU: NVIDIA L40 (22.5 GB)

Storage: 235.7 GB

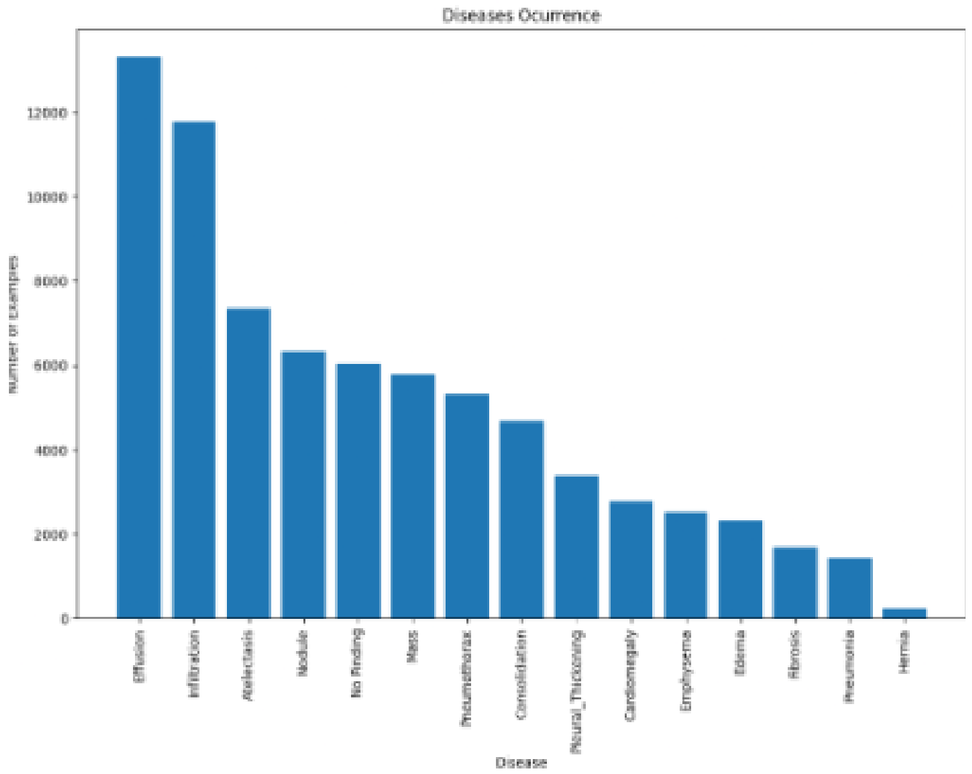
DATA PROCESSING

Original Distribution

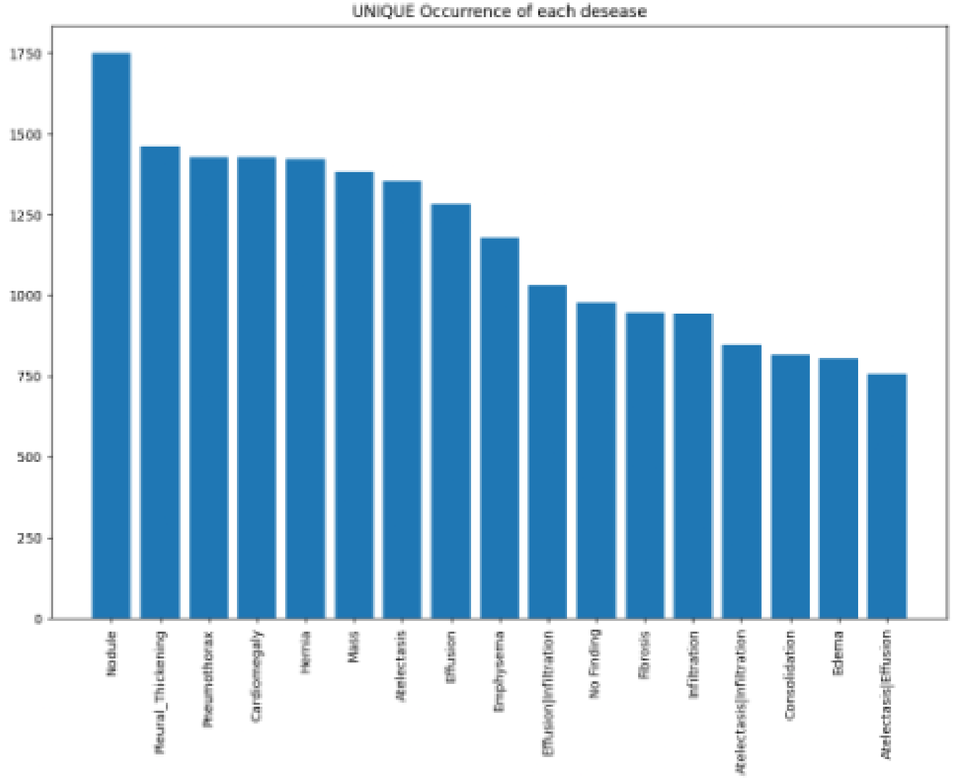


DATA PROCESSING

Final Dataset



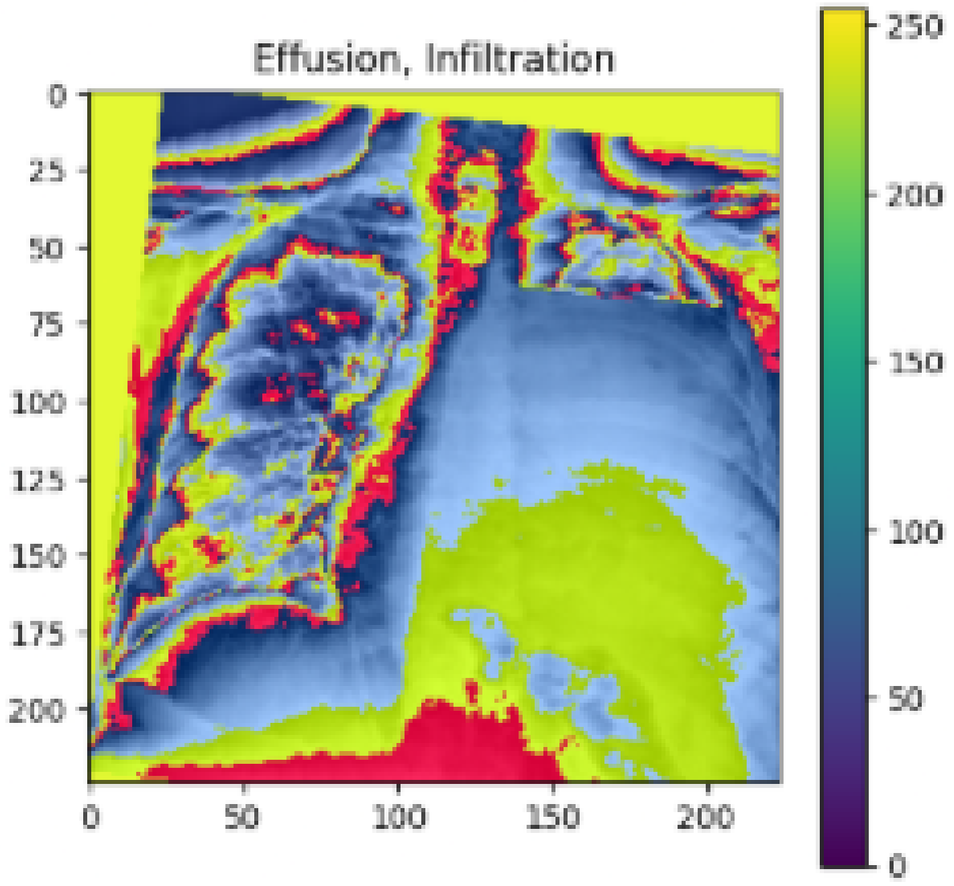
Train Dataset



DATA AUGMENTATION

Transformations for Training: train_transform

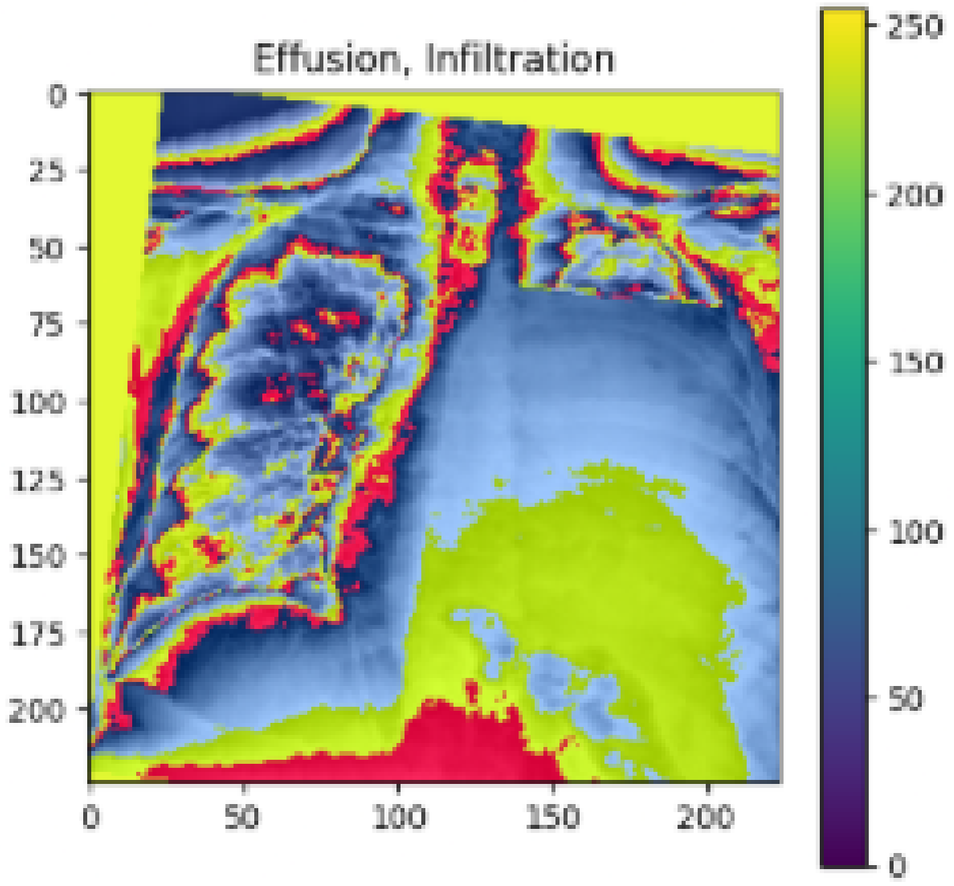
- Conversion of images to tensors with pixel values normalized between 0 and 1.
- Random cropping of approximately 90% of the original size, introducing spatial variation.
- Resizing to a fixed size (640px x 640px), ensuring uniformity in model input.
- Application of random affine transformations, including slight rotations and translations to simulate different acquisition angles.
- Application of Gaussian blur with kernel size of 3 and sigma in the range [0,2, 1.5] with a 40% probability to simulate noise in the images.
- Normalization using ImageNet dataset statistics.



DATA AUGMENTATION

Transformations for Validation and testing:
val_test_transform

- Conversion of images to tensors with pixel values scaled between 0 and 1.
- Cropping 90% of the original size to remove edge artifacts, maintaining spatial consistency.
- Resizing to match the dimensions expected by the model.
- Normalization using the same ImageNet statistics, ensuring inference conditions are consistent with training.



DATASET SPLIT

TRAINING SET (TRAIN_DF)

72% of the complete dataset.

VALIDATION SET (VALID_DF)

8% of the complete dataset.

TEST SET (TEST_DF)

20% of the complete dataset.

MODEL ARCHITECTURE

CONSISTS ON 4 MAIN BLOCKS:

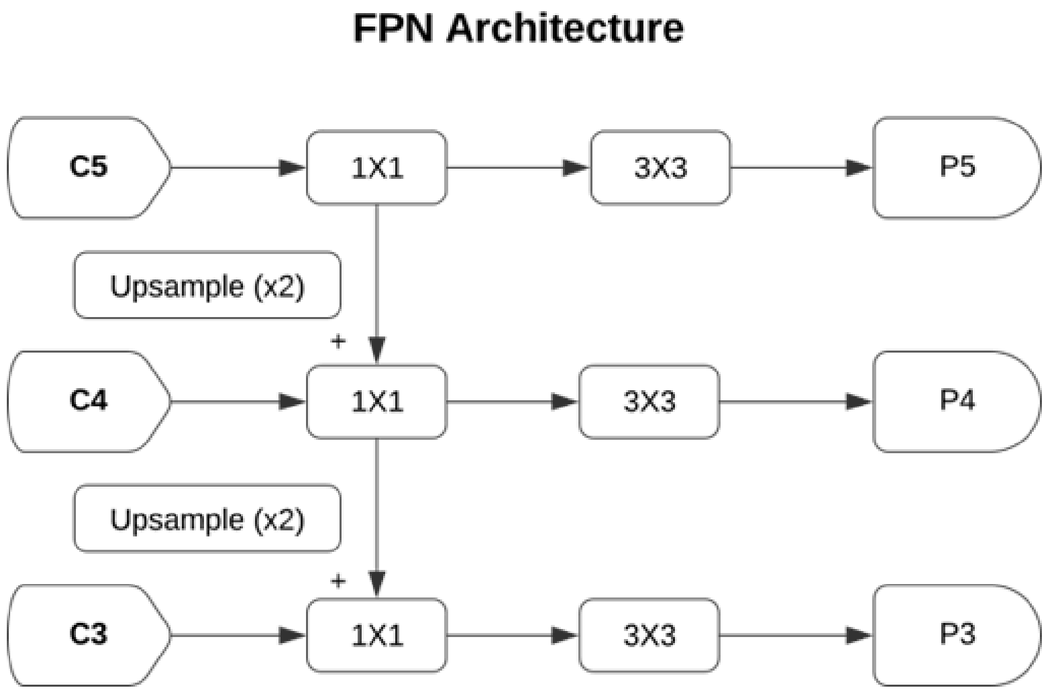
- BACKBONE-FPN
- IMAGE PROJECTION
- TABULAR DATA PROJECTION
- DATA FUSION

MODEL ARQUITECTURE

FPN - Feature Pyramid Network

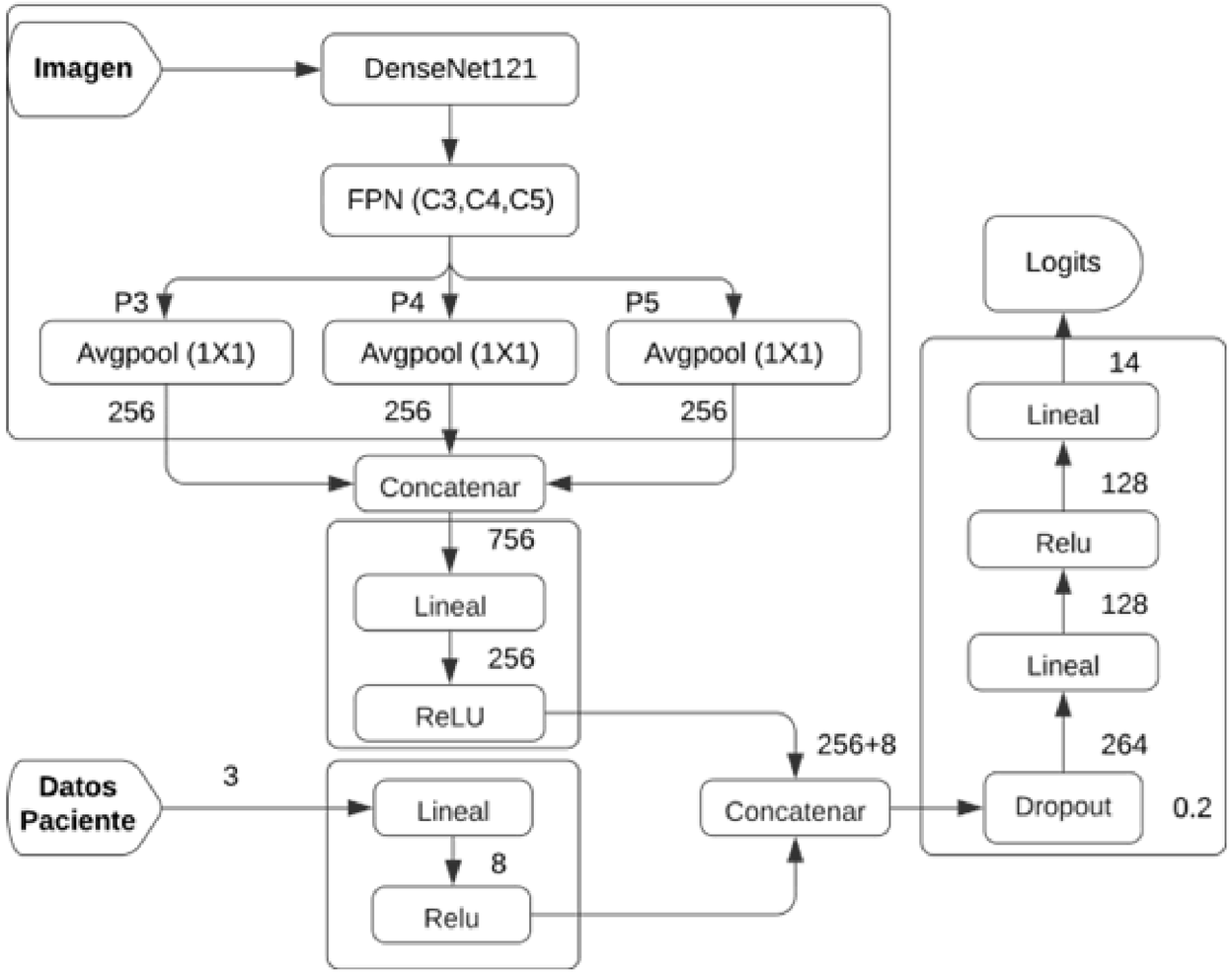
It was decided to implement a Feature Pyramid Network (FPN) in the model architecture because there is positive evidence supporting the use of this type of network for feature extraction tasks.

For context, an FPN provides a top-down pathway to build higher-resolution layers from a semanti-cally rich layer.



MODEL ARQUITECTURE

Complete Model



TRAINING PROCESS

MODEL ARQUITECTURE

Training parameters

- BatchSize

A batch size of 160 was set, which fully utilized the VRAM of the Nvidia L4 GPU. If you want to
train the model in an environment with lower specifications, it is recommended to reduce the batch
size. For the T4 GPU, we tested with a batch size of 100.

- Image Resolution

The image resolution was set to 224x224 pixels, as it represents a balance between quality and
performance, and DenseNet was originally trained at this resolution. Tests were also conducted with
resolutions of 580, 512, and 480 pixels, but the training process
was drastically affected, and the feature extractor’s performance was negatively impacted.

MODEL ARQUITECTURE

Training parameters

- Custom loss weights

According to the number of observations in the training dataset for each of the diseases, a custom weight is calculated: a higher weight is assigned to classes with fewer positive examples (less frequent), and a lower weight to the more common classes. These weights force the model to learn more from the less frequent classes.

Disease	Weight
Hernia	3.0943
Nodule	2.1768
Fibrosis	3.0209
Edema	2.8127
Cardiomegaly	2.5823
Effusion	1.6668
Pneumothorax	2.3087
Pleural Thickening	2.4494
Consolidation	2.4346
Emphysema	2.6907
Pneumonia	3.0943
Mass	2.2459
Atelectasis	1.8179
Infiltration	1.6402

MODEL ARQUITECTURE

Training parameters

It is important to highlight that the model underwent 3 training cycles. In the first 2 cycles, the weights from the previous table were used, but for the third cycle, the weights were manually set for the following diseases because the model was not correctly identifying them.

Disease	Weight
Pneumonia	4.5
Fibrosis	3.8
Consolidation	4.3

MODEL ARQUITECTURE

Training parameters

- Optimization Criterion

Binary Cross Entropy with Logits Loss (nn.BCEWithLogitsLoss) was used, as it is a function that
combines:

– Sigmoid activation.
– Binary Cross Entropy Loss.

- Epochs

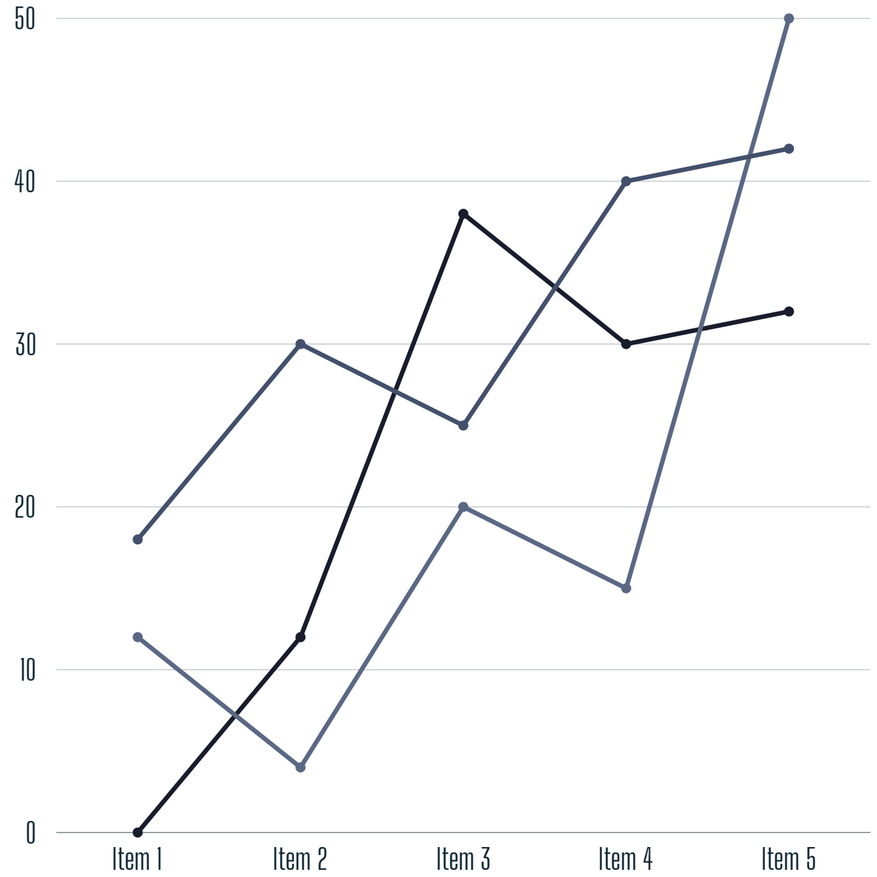
Three training cycles were performed, each consisting of 5 epochs.

- Learning Rate

A learning rate of 0.001 was set along with a scheduler, which is specified below.

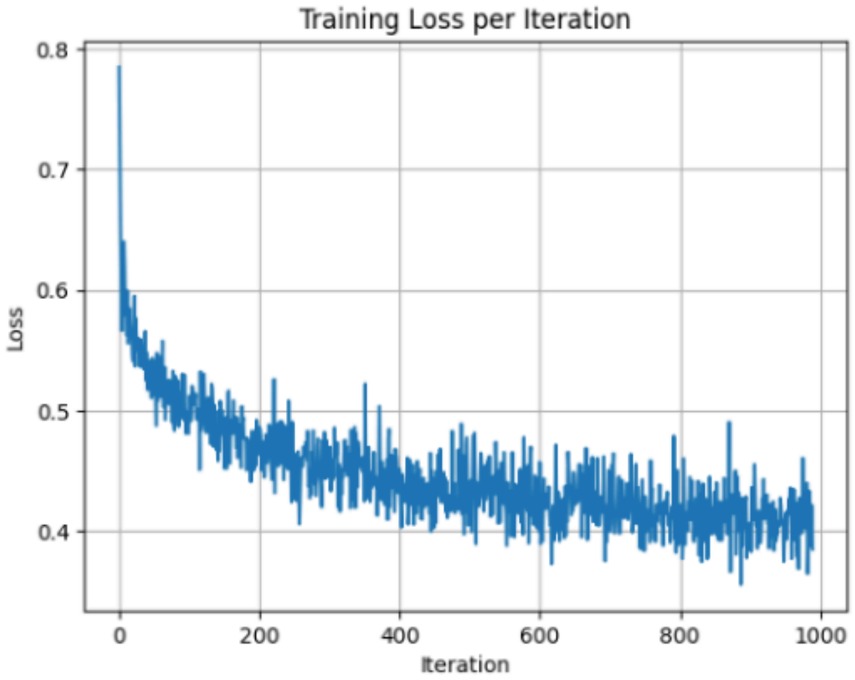
Scheduler: It monitors the validation loss (val loss) metric during training. If this metric does not improve for 2 epochs, the learning rate is reduced by half.

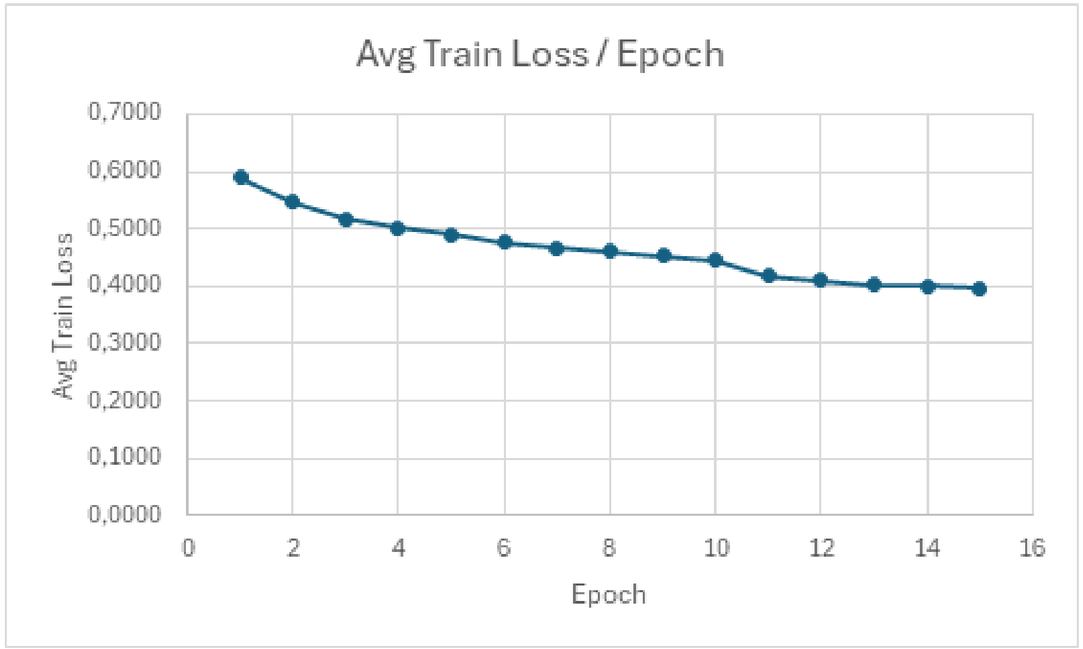
TRAINING METRICS



MODEL ARQUITECTURE

Training cycle





MODEL ARQUITECTURE

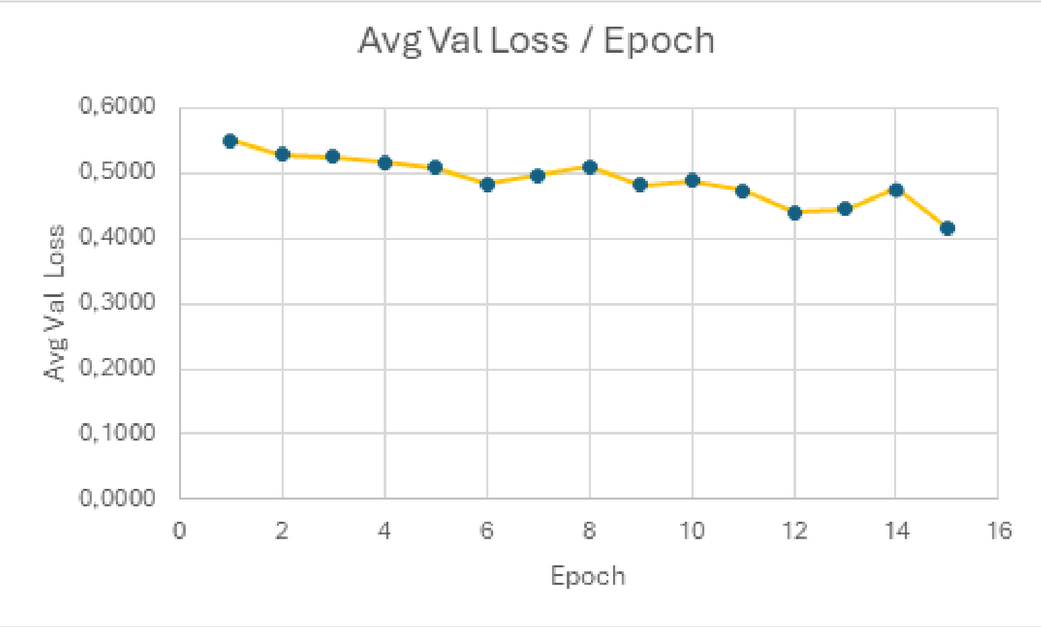
Validation cycle

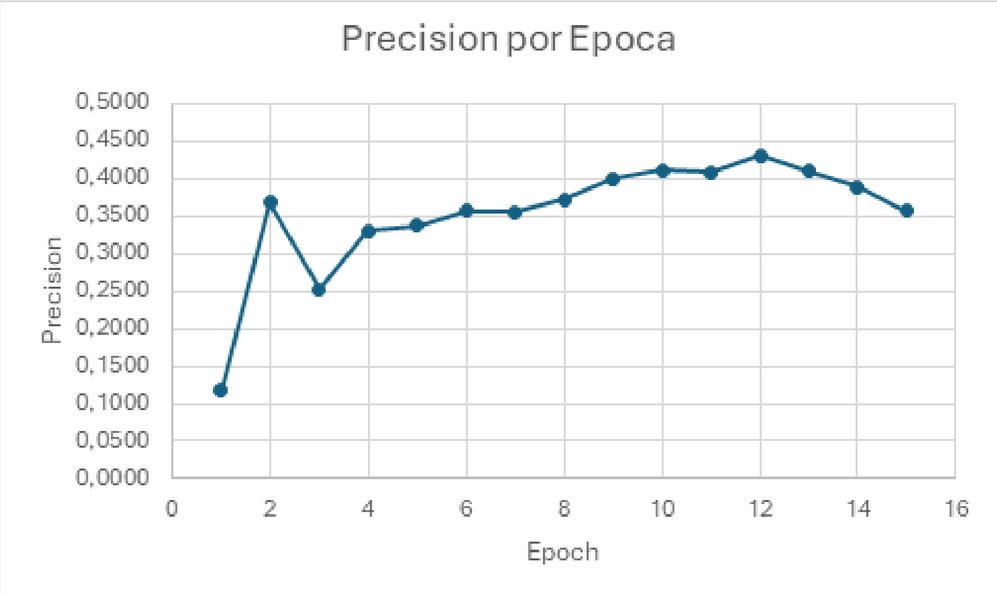
Our model outputs logits. Before making pre-dictions with these data, it is necessary to apply the torch.sigmoid function, which outputs the probabilities of each class being present. However, due to the imbalance in the dataset, it is recommended to use a dynamic threshold for each class (instead of a fixed probability for all) to decide whether the disease is present according to its customized threshold. We conducted tests calculating these thresholds and have three functions that compute the thresholds, but at the end, we decided to only implement one:

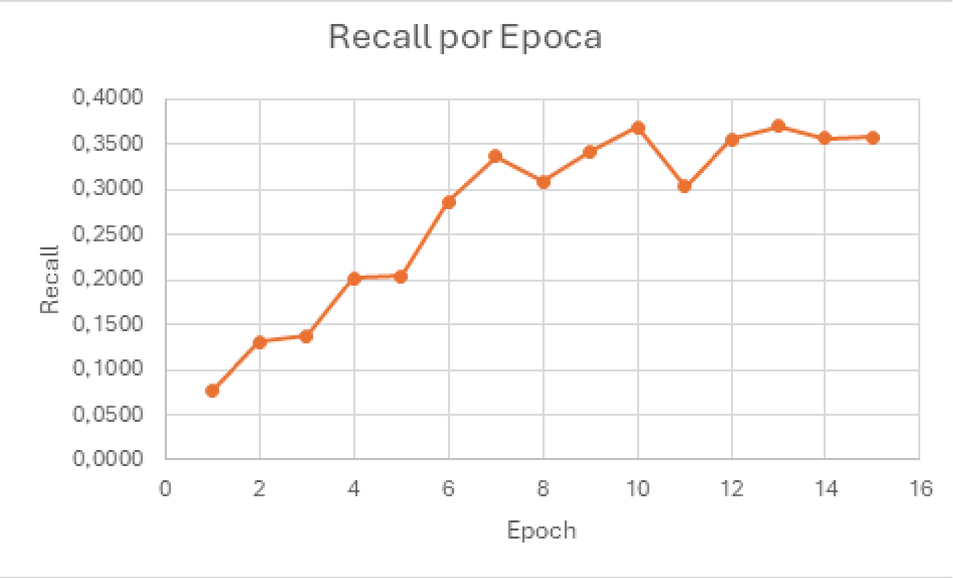
find_optimal_thresholds_with_dynamic_adjustment: This variant also aims to maximize the F1-score under a minimum precision constraint but adds a dynamic adjustment if the performance (measured by F1-score) is insufficient

MODEL ARQUITECTURE

Validation cycle

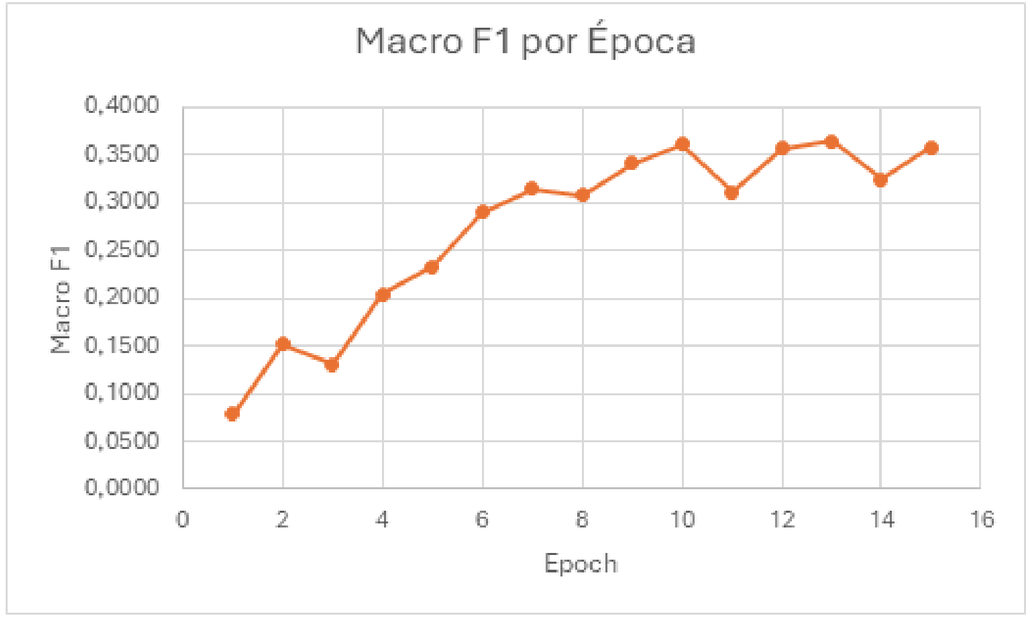


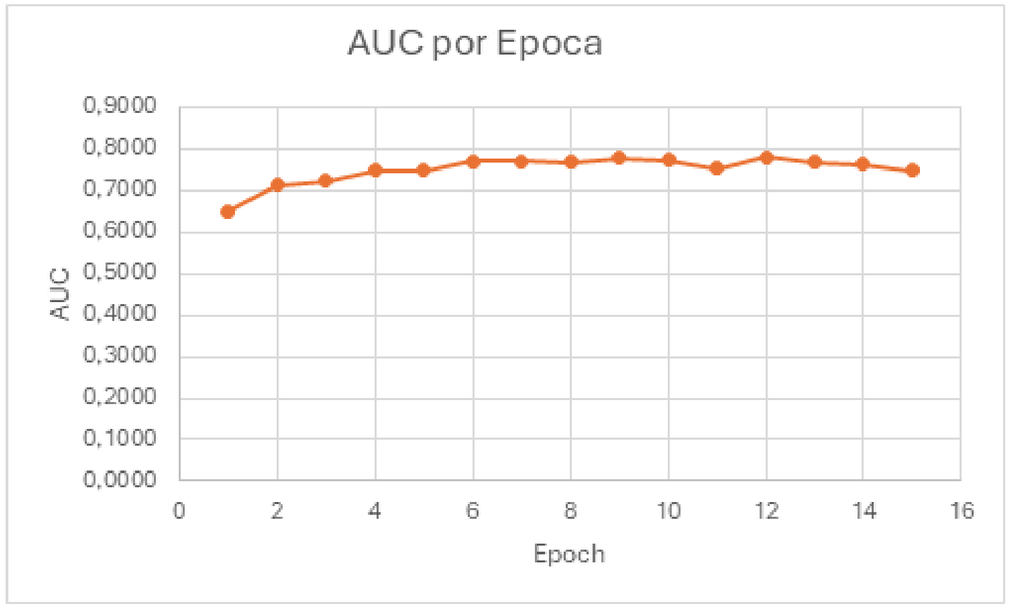




MODEL ARQUITECTURE

Validation cycle





MODEL ARQUITECTURE

Validation cycle

Metrics summary

E	Avg Loss	Val Loss	Precisión (mean)	Recall (mean)	F1 Score (mean)	Macro F1	AUC (mean)
1	0.5895	0.5504	0.1171	0.0770	0.0785	0.0785	0.6495
2	0.5466	0.5296	0.3679	0.1307	0.1520	0.1520	0.7138
3	0.5181	0.5257	0.2527	0.1375	0.1302	0.1302	0.7231
4	0.5009	0.5175	0.3303	0.2021	0.2041	0.2041	0.7484
5	0.4894	0.5083	0.3376	0.2039	0.2330	0.2330	0.7493
6	0.4772	0.4844	0.3569	0.2865	0.2897	0.2897	0.7698
7	0.4672	0.4972	0.3555	0.3365	0.3147	0.3147	0.7717
8	0.4606	0.5107	0.3717	0.3091	0.3074	0.3074	0.7669
9	0.4533	0.4820	0.4006	0.3428	0.3416	0.3416	0.7779
10	0.4452	0.4883	0.4121	0.3696	0.3617	0.3617	0.7723
11	0.4174	0.4739	0.4089	0.3032	0.3110	0.3110	0.7526
12	0.4095	0.4399	0.4316	0.3556	0.3571	0.3571	0.7794
13	0.4030	0.4457	0.4095	0.3708	0.3649	0.3649	0.7687
14	0.4003	0.4775	0.3896	0.3568	0.3249	0.3249	0.7639
15	0.3972	0.4485	0.3570	0.3513	0.3577	0.3577	0.7485

MODEL ARQUITECTURE

Training Process

After each validation cycle, the best model so far is saved.
The criterion we considered most important for selecting the “Best model,” was:

- **Macro F1**
  - What it measures: It is the average of the F1-score for each class, giving equal weight to every label regardless of how frequent or rare it is.
  - Why it matters: In multilabel classification, rare labels tend to be ignored if a global metric like accuracy or micro F1 is used. Macro F1 forces the model to perform well on all labels equally.
  - Advantage: It favors models that have balanced performance between common and rare classes.
  - Ideal when all classes are equally important.

MODEL ARQUITECTURE

Resulting Files

- **validation metrics [Model's name].csv**
It contains the model's metrics obtained using the test dataset (test df).

- **configuracion.txt**
Contains the configuration and logs of the training process.

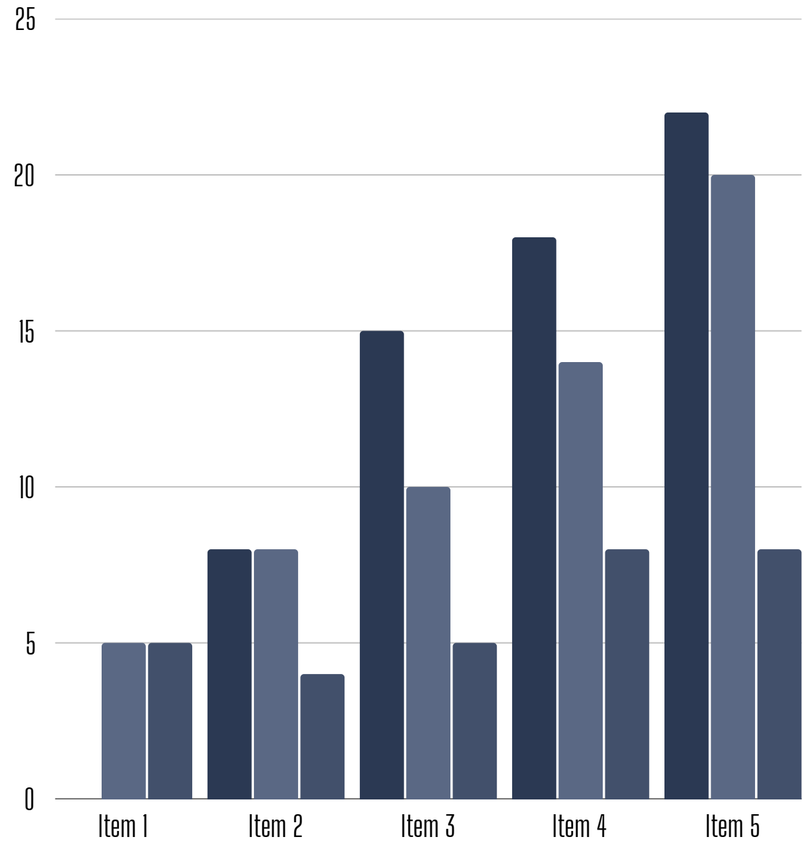
- **optimizer classification.pt**
Contains the current state of the optimizer.

- **model classification.pt**
Contains the weights of the model architecture.

- **thresholds.json**
Contains the custom thresholds for each disease.

- The files for the final model are located in the following archive: FINAL DenseNetTuned.zip.

RESULTS & VALIDATION METRICS



ACCEPTANCE THRESHOLDS

THE THRESHOLD VALUES OBTAINED PER DISEASE REFLECT THE MODEL'S INDIVIDUAL PERFORMANCE FOR EACH CLASS, BASED ON ITS ABILITY TO BALANCE PRECISION AND F1-SCORE.

Disease	**Weight**
Hernia	20
Nodule	50
Fibrosis	65
Edema	75
Cardiomegaly	40
Effusion	50
Pneumothorax	25
Pleural_Thickening	45
Consolidation	50
Emphysema	40
Pneumonia	45
Mass	25
Atelectasis	35
Infiltration	45

VALIDATION METRICS

Exact Match Ratio	Hamming Loss	Recall Micro	Recall Macro	Label Ranking Avergae	Accuracy
14.21%	13.26%	0.43	0.38	64%	86.74%

F1 SCORE (CLASS)

THE F1 SCORE EVALUATES THE PERCENTAGE OF TRUE POSITIVES WITH RESPECT TO TRUE POSITIVES, FALSE NEGATIVES, AND FALSE POSITIVES; IT AVERAGES ACROSS ALL CLASSES IF MACRO, AND OVER ALL INSTANCES IF MICRO

Class	F1 Score
Hernia	0.4237
Nodule	0.2261
Fibrosis	0.1538
Edema	0.1548
Cardiomegaly	0.5475
Effusion	0.6020
Pneumothorax	0.4736
Pleural Thickening	0.2565
Consolidation	0.0259
Emphysema	0.4416
Pneumonia	0.0330
Mass	0.3814
Atelectasis	0.4695
Infiltration	0.4147

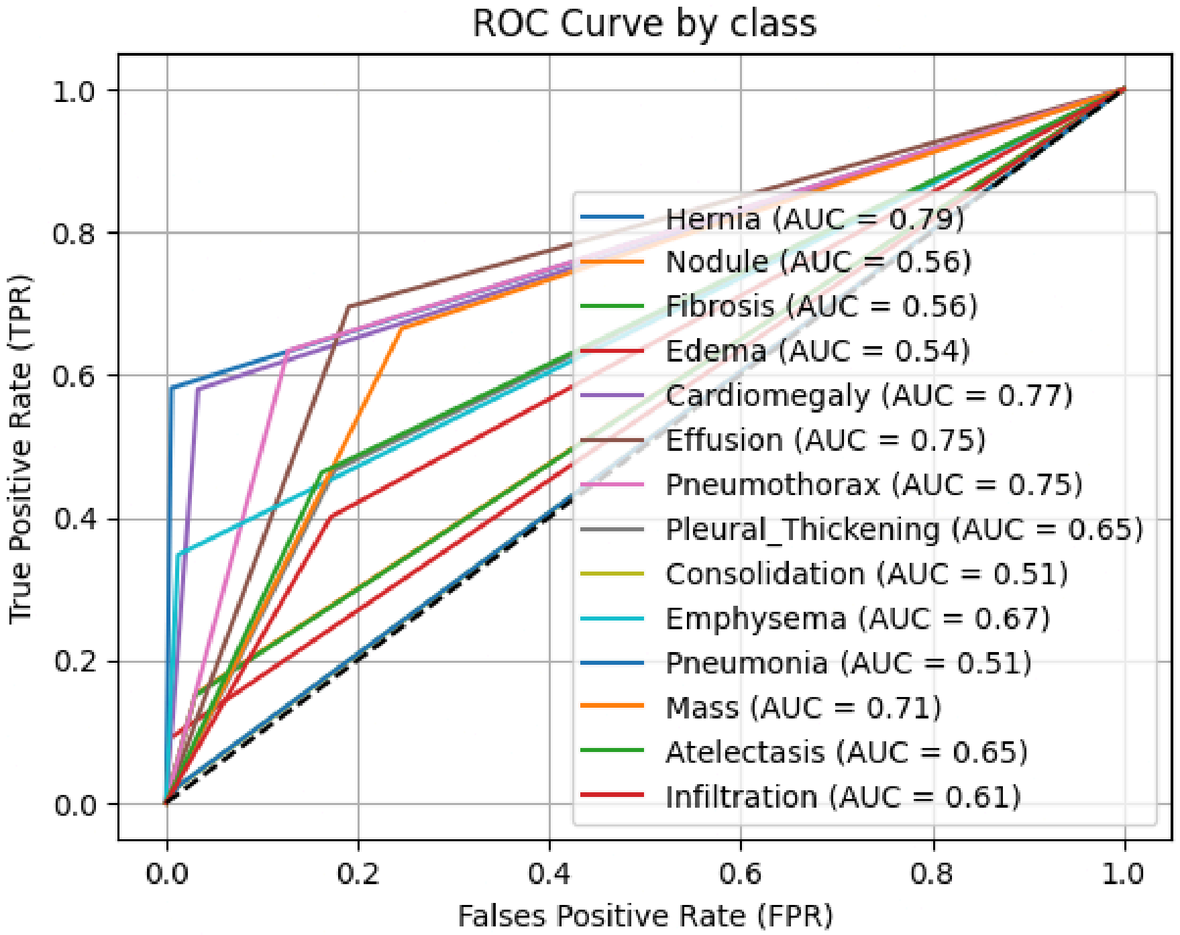
CONFUSION MATRICES

CONFUSION MATRICES SHOW THE NUMBER OF TRUE POSITIVES AND TRUE NEGATIVES, AS WELL AS FALSE POSITIVES AND FALSE NEGATIVES PREDICTED FOR EACH CLASS

Clase	Matrices de confusión
Hernia	$\begin{pmatrix} 8494 & 50 \\ 18 & 25 \end{pmatrix}$
Nodule	$\begin{pmatrix} 7186 & 230 \\ 982 & 177 \end{pmatrix}$
Fibrosis	$\begin{pmatrix} 8023 & 252 \\ 265 & 47 \end{pmatrix}$
Edema	$\begin{pmatrix} 8146 & 32 \\ 372 & 37 \end{pmatrix}$
Cardiomegaly	$\begin{pmatrix} 7815 & 270 \\ 211 & 291 \end{pmatrix}$
Effusion	$\begin{pmatrix} 5301 & 1250 \\ 621 & 1415 \end{pmatrix}$
Pneumothorax	$\begin{pmatrix} 6676 & 975 \\ 343 & 593 \end{pmatrix}$
Pleural_Thickening	$\begin{pmatrix} 6541 & 1401 \\ 344 & 301 \end{pmatrix}$
Consolidation	$\begin{pmatrix} 7749 & 25 \\ 802 & 11 \end{pmatrix}$
Emphysema	$\begin{pmatrix} 8040 & 101 \\ 291 & 155 \end{pmatrix}$
Pneumonia	$\begin{pmatrix} 8229 & 110 \\ 242 & 6 \end{pmatrix}$
Mass	$\begin{pmatrix} 5701 & 1863 \\ 343 & 680 \end{pmatrix}$
Atelectasis	$\begin{pmatrix} 5451 & 1057 \\ 1117 & 962 \end{pmatrix}$
Infiltration	$\begin{pmatrix} 5372 & 1117 \\ 1257 & 841 \end{pmatrix}$

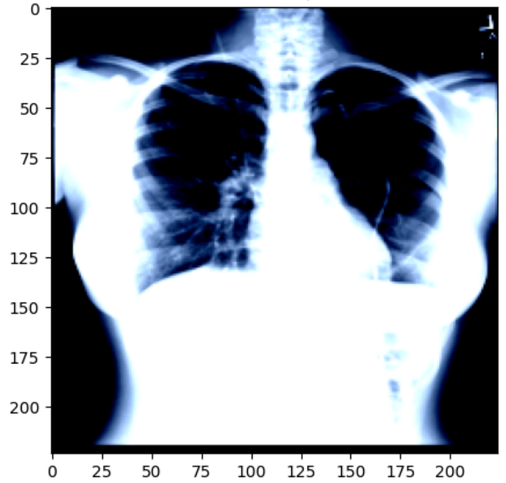
ROC CURVE

THE ROC CURVE (RECEIVER OPERATING CHARACTERISTIC) PLOTS THE VALUES BETWEEN THE FALSE POSITIVE RATE (ON THE X-AXIS) AND THE TRUE POSITIVE RATE (ON THE Y-AXIS), OBTAINED BY MOVING THE MINIMUM PREDICTED PROBABILITY THRESHOLD FROM 0 TO 1 TO CONSIDER A PREDICTION AS POSITIVE (I.E., THAT THE CLASS IS CONSIDERED PRESENT).
IN A GOOD ROC CURVE, THE CURVE SHOULD TEND TOWARDS THE UPPER LEFT CORNER.

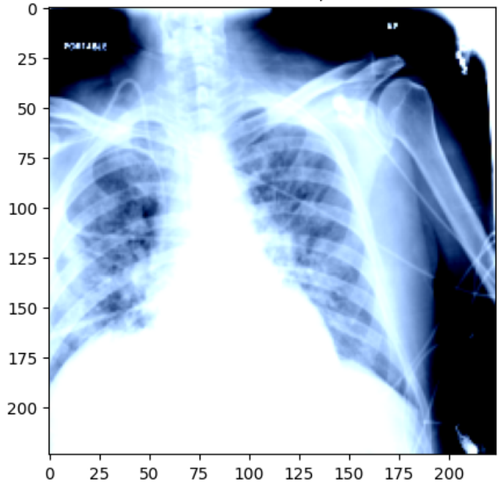


PREDICTION MADE BY THE MODEL

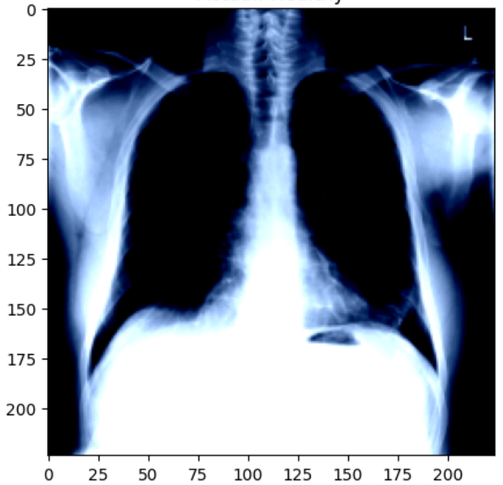
Predicted: Effusion, Pneumothorax, Pleural_Thickening, Mass
Actual: Pneumonia, Atelectasis



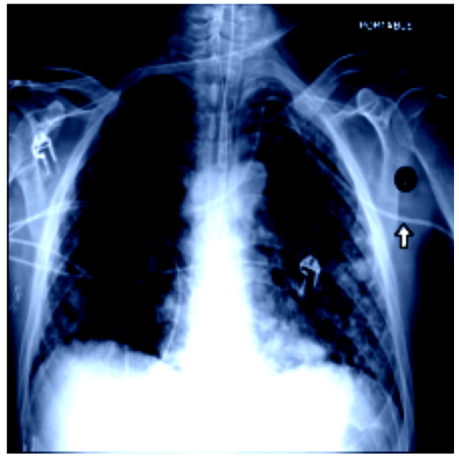
Predicted: Effusion, Infiltration
Actual: Consolidation, Infiltration



Predicted: Healthy
Actual: Healthy



Perfect Match
Pred: Nodule, Infiltration
True: Nodule, Infiltration



Perfect Match
Pred: Infiltration
True: Infiltration

CONCLUSION

CONCLUSION

In conclusion, the multilabel classification model demonstrated solid performance on well-represented classes such as Effusion, Atelectasis, and Infiltration, indicating that the model effectively learns when trained with sufficient data. Notably, it also achieved acceptable results on rare and complex classes like Hernia and Pneumothorax, suggesting its ability to generalize despite data imbalance. However, very low F1-scores in classes such as Pneumonia and Consolidation reveal challenges related to overlapping features and limited training examples. The impact of class imbalance was evident, especially in underrepresented conditions like Fibrosis and Pleural Thickening, emphasizing the need for mitigation strategies such as reweighting, oversampling, or synthetic data generation. Moreover, the model exhibited a mismatch between precision and recall across different classes, which should be adjusted depending on clinical priorities—often favoring recall to minimize false negatives. Lastly, the implementation of class-specific thresholding proved beneficial in optimizing metrics like F1-score, particularly in the context of high class imbalance typical of multilabel medical imaging tasks.

REFERENCES

[1] Hasanah, U., Cahyana, A. (December 27, 2023). CheXNet and feature pyramid network: a fusion deep learning architecture for multilabel chest X-Ray clinical diagnoses classification. Retrieved from PubMed: https://pubmed.ncbi.nlm.nih.gov/38150139/

[2] Hui, J. (March 26, 2018). Understanding feature pyramid networks for object detection (FPN). Retrieved from Medium: https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c

[3] Ito Aramendia, A. (March 1, 2024). DenseNet: a complete guide. Retrieved from Medium: https://medium.com/@alejandro.itoaramendia/densenet-a-complete-guide-84fedef21dcc

[4] Liz, H., Huertas-Tato, J., S´anchez-Monta˜n´es, M., Del Ser, J., & Camacho, D. (n.d.). Deep learning for understanding multilabel imbalanced Chest X-ray datasets. Retrieved from ar5iv.labs.arxiv: https://ar5iv.labs.arxiv.org/html/2207.14408