POLITECNICO
MILANO 1863

Applied AI in Biomedicine

Team "M2S" Project Report

Mehrshad Alipoor, Shahryar Namdari Ghareghani, Maurizio Tirabassi

244246, 246980, 248986

May 21, 2025

1 Introduction

This work develops four deep-learning classifiers for lung computed tomography (CT) images: two models process entire axial slices (one for five-level malignancy scoring and one for binary benign versus malignant classification) and two analogous models focus on zoomed-in nodule regions. The five-class models predict malignancy scores from 1 (least severe) to 5 (most severe), while the binary models group scores 1–3 as benign and 4–5 as malignant.

Our pipeline begins by evaluating a range of pretrained convolutional neural network (CNN) feature extractors to identify the most effective backbone. For each selected extractor, we attach a classification head and fine-tune the combined network using task-specific data augmentations and class-balancing strategies. We then perform an extensive hyperparameter search, varying batch size, optimizer, classifier head architecture, and learning-rate schedules.

1.1 Paradigm Motivation

Machine learning provides a way to solve problems without explicit algorithmic solutions. It does so by learning from data: a parameterized function maps inputs to outputs, and its parameters are optimized through numerical methods to fit observed relationships and generalize to unseen samples, while balancing flexibility to prevent overfitting.

Within this framework, data are represented as

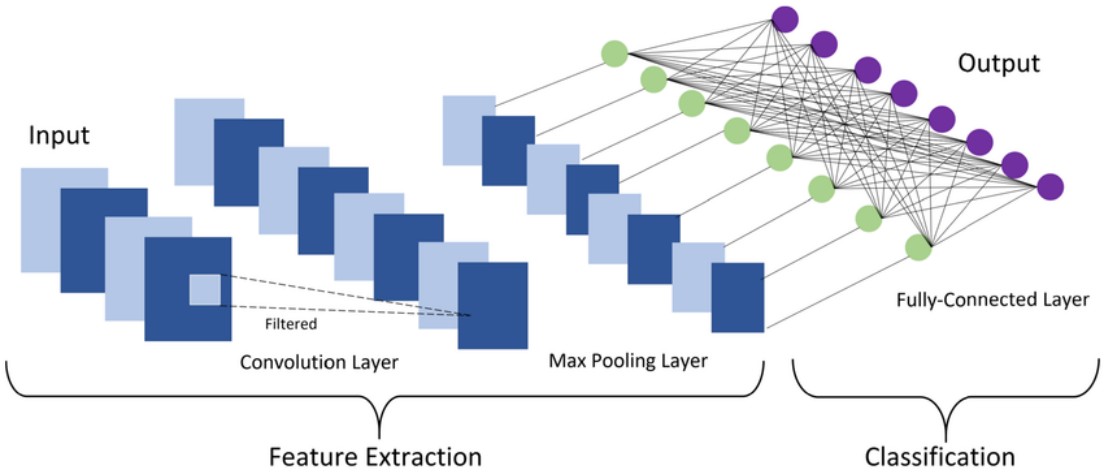

Figure 1: Schematic of a convolutional neural network functioning as a feature extractor for a classification task.

tensors, i.e. n-dimensional arrays of numerical values. In the present work, each CT image was handled as a two-dimensional tensor. Classifying images automatically requires defining decision boundaries in the very high-dimensional pixel space. Converting each image into a lower-dimensional feature vector simplifies this task. Features can be handcrafted or learned automatically.

Fully connected layers are unsuitable for image processing because their large number of parameters risks overfitting. Under the deep-learning approach, convolutional neural networks perform data-driven feature extraction, as represented in Figure 1.

2 Data & Preprocessing

2.1 Data Analysis

The dataset is constituted by 2363 pairs of full slices and nodule slices, each labeled with a malignancy score from 1 to 5. The dataset is very small, so it was already acknowledged that it would pose a challenge to properly train a five-class model that generalizes, especially in the possible presence of class imbalance.

In this paragraph, we present relevant statistics needed to understand the nature of the data. This step is crucial, as it will determine which aspects must be addressed when preprocessing the dataset (extended upon in Section 2.2).

2.1.1 Image Shape Variability

The dimensions of both full slices and nodule-only crops vary as shown in Table 1.

Table 1: Minimum, maximum, and mean ± standard deviation of image dimensions (H×W) for full slices and nodule crops.

Type	Min	Max	Mean ± Std
Fullslice	512×512	512×512	$512.00 \pm 0.00 \times 512.00 \pm 0.00$
Nodule	45×44	138×126	$55.70 \pm 10.49 \times 55.71 \pm 10.49$

All full slices share an identical resolution of 512×512 pixels, whereas nodule crops differ widely in size. Because pretrained feature extractors require a fixed input size, all slices will be resized, and the nodule crops will also need to be padded to preserve their aspect ratios.

2.1.2 Intensity Value Interpretation

CT images encode pixel intensities in Hounsfield Units (HU), where water is 0 HU and air is –1000 HU [5]. The distribution of HU for both full slices and nodule crops is summarized in Table 2.

Table 2: Minimum, maximum, and mean ± standard deviation of Hounsfield Units (HU) for fullslice and nodule images.

Type	Min	Max	Mean ± Std
Fullslice	-32768	32713	-783.5 ± 798.97
Nodule	-3024	3071	-591.2 ± 352.07

In our dataset, full slices span nearly the entire 16-bit range, while nodule crops occupy a much narrower window. To prevent extreme values from skewing network training and to focus on clinically relevant tissue contrasts, we will apply intensity clipping before normalization in the preprocessing pipeline.

2.1.3 Class Distribution

Figure 2 presents two label distributions: one for the five-class malignancy scale and one for the benign vs. malignant classification, which are identical for both the full-slice and nodule datasets (they share the same labels and counts).

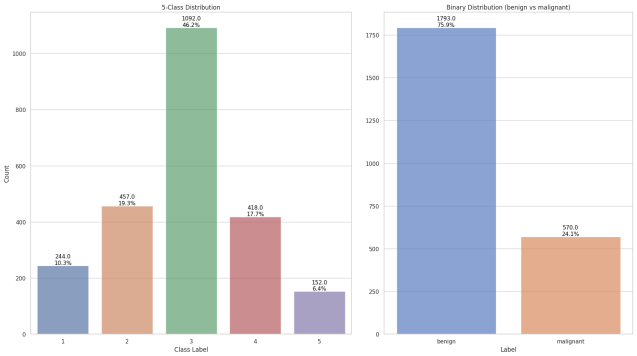

Figure 2: Five-class (left) and binary (right) malignancy distributions for full-slice and nodule datasets.

The dataset exhibits pronounced class imbalance: class 3 is overrepresented, while class 5 is underrepresented. Binarizing the labels maintains

a pronounced imbalance. In lung nodule analysis, this reflects the relatively low prevalence of highly malignant lesions compared to benign or indeterminate ones [11].

2.1.4 Noise Estimation

Under the assumption that CT image noise is locally homogeneous, noise levels were estimated by sampling two uniform regions on a single slice: air background and lung parenchyma, and plotting their Hounsfield-unit histograms [5].

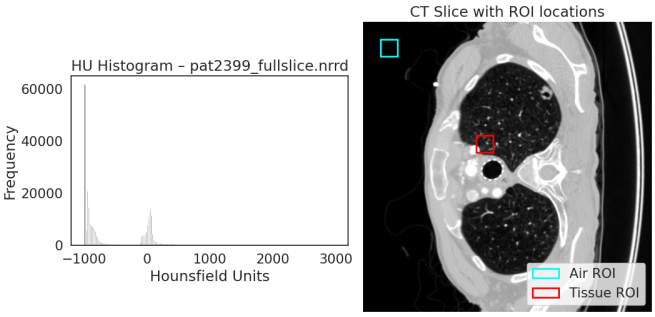

Figure 3: HU histograms for homogeneous ROIs: air (mean = –997.0 HU, σ = 4.2 HU) and lung tissue (mean = –555.3 HU, σ = 785.1 HU)

Figure 3 indicates minimal noise in air and physiological variability in tissue, thus not requiring the application of any denoising filter. Applying one could blur diagnostically relevant details.

To validate this observation across the dataset, Laplacian variance (LV), i.e. a proxy for noise and sharpness [7], was employed to quantitatively characterize noise.

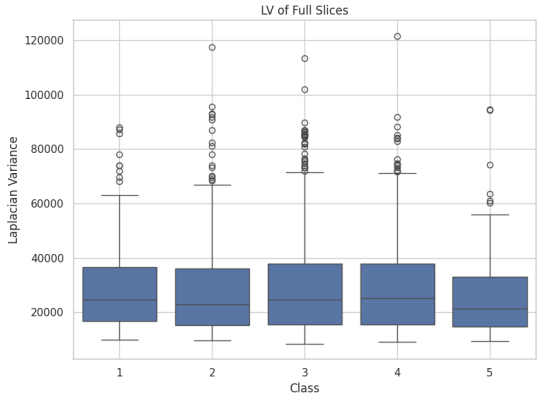

Figure 4: Boxplot of Laplacian variance by malignancy class for full-slice images.

Table 3: Number of images above the 90th, 95th, and 99th LV percentiles by slice type.

Type	P99	P95	P90
Fullslice	24	119	237
Nodule	24	119	237

As reported in Table 3, only 24 out of 2363 full-slice images exceed the 99th percentile, demonstrating that extreme noise cases are rare. Since excluding more than 1–2% of images risks discarding clinically important examples, all images, including high-LV outliers, have been retained.

2.2 Data Preprocessing

Effective data preprocessing is essential to ensure stable model training and reliable downstream performance. Although end-to-end learning ideally handles raw inputs, numerical transformations, such as feature scaling, remain critical to prevent noisy or extreme values from destabilizing optimization.

Within this work, preprocessing focused on four main steps: HU value clipping, padding & resizing, and z-score normalization. All choices (clipping thresholds, padding strategy, resize dimension, normalization method) were selected based on recommendations from CT preprocessing studies [12, 5, 2].

Preprocessing produced two parallel datasets: full and cropped nodule slices, which serve as the starting point for all subsequent experiments. Although preprocessing was applied before splitting the data, each image was processed independently (no global statistics were shared) so there was no risk of information leakage.

2.2.1 Hounsfield Units Clipping

Clipping HU values to a diagnostic window removes non-informative extremes, such as air and bone, while enhancing tissue contrast in the range of interest [2, 5]. Following these guidelines, all pixels were constrained to the interval $[-1000, 400]$, filtering out air and high-density artifacts that lie outside the diagnostic spectrum and could otherwise bias network activations.

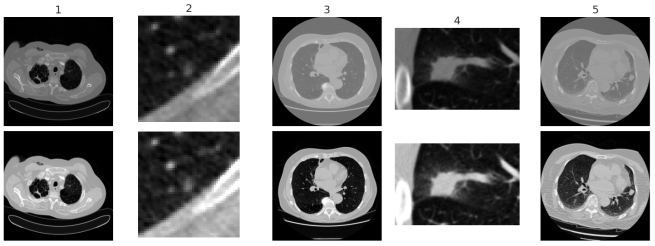

Figure 5: Example slice before (above) and after (below) HU clipping to $[-1000, 400]$.

Some studies recommend a tighter window, i.e. $[-600, 400]$, to isolate lung parenchyma more aggressively during nodule preprocessing [12]. Figure 6 illustrates this approach. Although our primary pipeline uses the wider $[-1000, 400]$ range to preserve perinodular context, we still report on the $[-600, 400]$ protocol here because it remains widely cited and relevant in CT preprocessing literature.

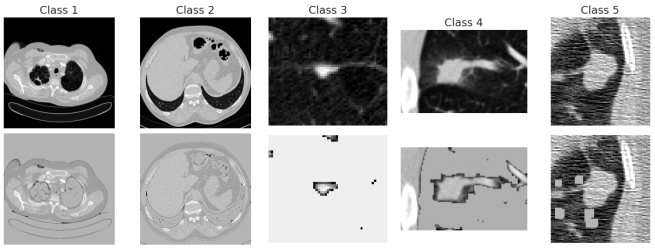

Figure 6: Before (above) and after (below) lung segmentation through a more aggressive HU clipping.

2.2.2 Padding and Resizing

All full-slice images were resized to the standard input size of 224×224. Nodule crops were first made square by symmetric zero padding, i.e. adding equal numbers of zero-valued pixels to the shorter side, then bilinearly resized to 224×224.

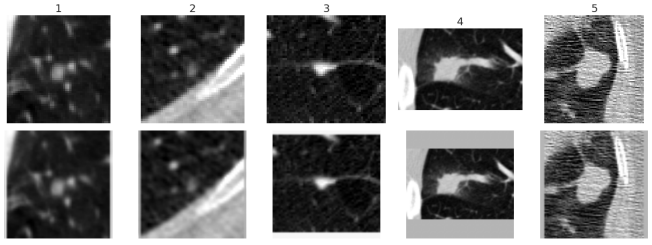

Figure 7: Example nodule crops before (above) and after (below) symmetric zero-padding and bilinear resizing to 224×224.

Zero padding is especially well suited to CT lung nodule classification because a padding value of zero corresponds to the Hounsfield unit for air outside the body, avoiding the introduction of anatomically implausible intensities that could confuse feature extractors.

2.2.3 Z-Score Normalization

Prior to model input, pixel intensities of each slice were Z-score normalized, i.e. subtracting the mean and dividing by the standard deviation, to enforce zero mean and unit variance across all pixels. This standardization ensures that each input contributes equally during optimization and helps stabilize gradient updates.

2.3 Outlier Detection

Before model training, we evaluated the preprocessed datasets for potential outliers. Directly identifying outliers in high-dimensional image space is challenging, so we leveraged lower-dimensional feature representations extracted by convolutional neural networks. Pretrained CNN backbones are well established for capturing semantically meaningful embeddings of images, even prior to fine-tuning. We applied t-distributed stochastic neighbor embedding (t-SNE) to project these feature vectors into two dimensions, enabling visual inspection of cluster structure and potential outliers (see Figure 8).

We selected MobileNetV3-Large, ResNet-50, and VGG-16 for outlier detection due to their complementary feature extraction capabilities [9, 1], all pretrained on the ImageNet dataset:

- **VGG-16** employs small convolutional filters to extract mid-level features, providing a balance between detail and abstraction.
- **ResNet-50** utilizes residual connections to learn complex, high-level abstractions, effectively capturing detailed patterns.
- **MobileNetV3-Large** captures fine-grained, low-level features, making it suitable for large-scale preliminary analyses.

By integrating these architectures, we obtain diverse feature representations (from low-level details to high-level abstractions) enhancing the robustness of our approach.

Through visual inspection, no clear outlier clusters were detected. Moreover, in medical

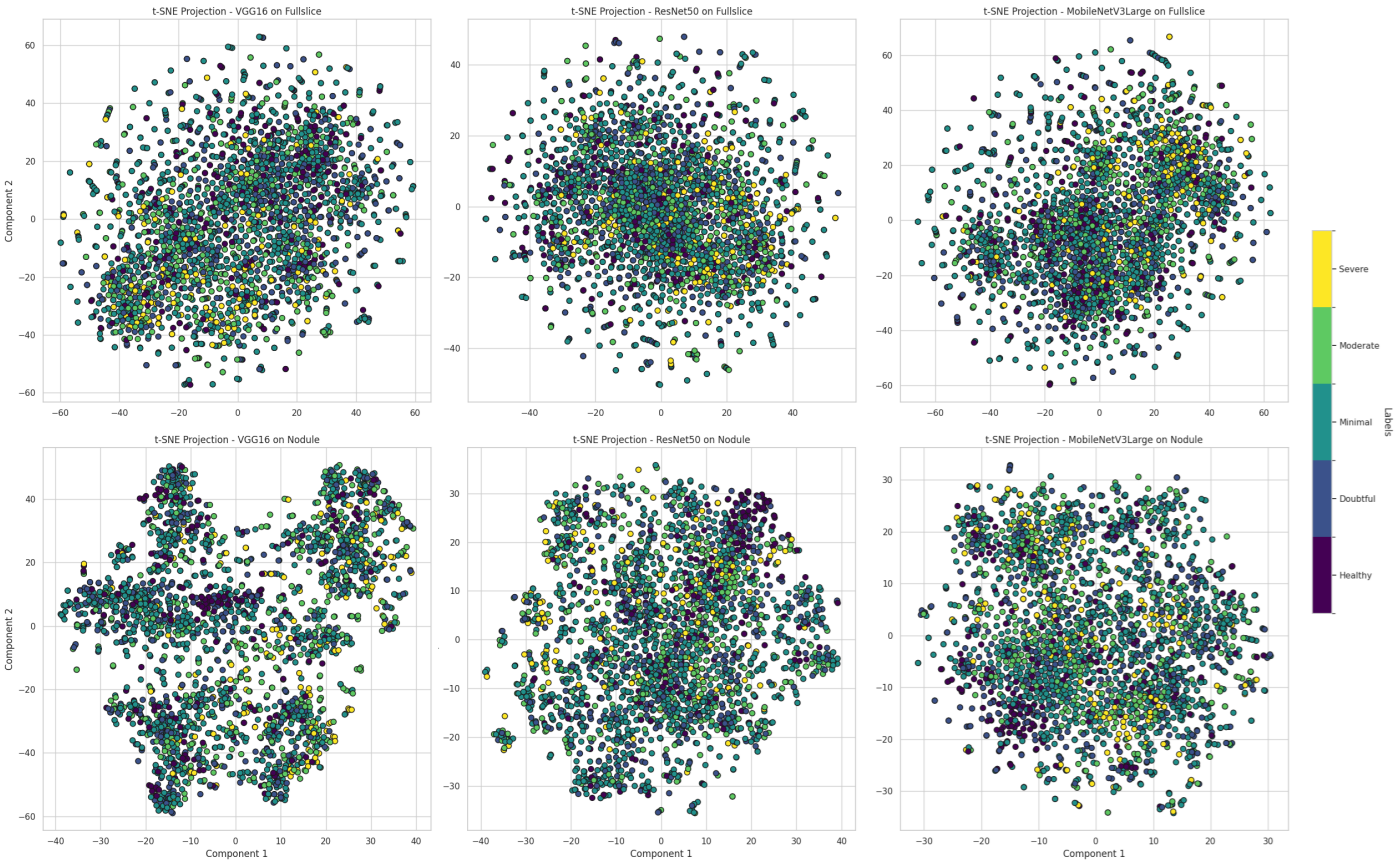

Figure 8: t-SNE projections for full slices nodules (rows) and for each one of the three backbones (columns).

imaging, such cases may reflect rare but clinically important variations. For these reasons, we chose to retain all samples and not pursue further outlier exclusion.

2.4 Dataset Splitting

We split each preprocessed dataset into training (80%), validation (10%), and test (10%) sets by stratifying on the original five-class malignancy labels. Stratification preserves the clinical prevalence of each class so that both validation and test sets reflect real-world distributions. Any balancing interventions are going to be applied only to the training set.

The split for binary classifications was generated after the five-class split. If we merged labels before splitting, the test or validation subsets risked lacking representation from some original classes. By first creating five-class folds and then collapsing them to binary, we ensure that every held-out set contains the full spectrum of pathology.

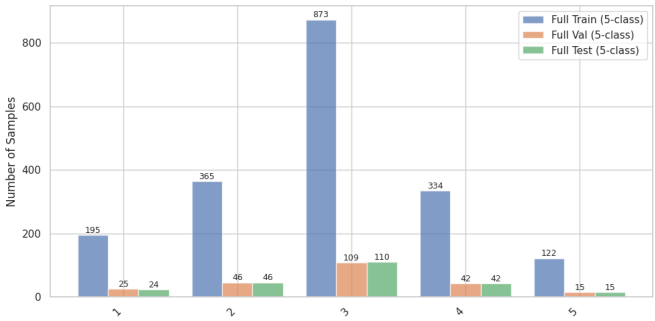

Figure 9: Dataset split for 5 class classification.

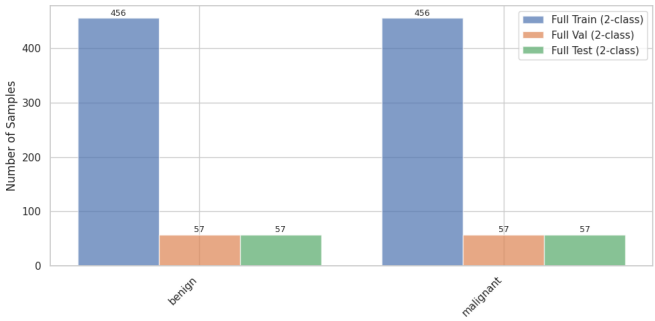

Figure 10: Dataset split for binary classification.

2.4.1 Evaluation Metric Selection

Given that our validation and test sets remain imbalanced, we cannot rely on overall accuracy or similar metrics (these would be driven almost entirely by the majority class and mask performance on the rarer, high-risk cases). Instead, we adopt Weighted Area Under the Precision-Recall Curve (W-AUPRC) as the main metric (since it captures precision-recall trade-offs without being skewed by abundant true negatives) and use the F1 score for intermediate tasks (e.g., addressing class imbalance), as it provides a single, balanced measure of precision and recall.

3 Methodology & Results

3.1 Feature Extractor Screening

We define the feature extractor as the network backbone that maps input images to a lower-dimensional embedding space, where Euclidean distances correlate with semantic similarity. To select the most effective backbone for our task, we evaluated several state-of-the-art CNN architectures [9, 1] on the imbalanced, preprocessed datasets to identify those that are inherently better at representing minority classes. The employed backbones were pretrained on the ImageNet dataset.

Following the methodology of Yosinski et al. [13], we measured each model's feature quality by fine-tuning progressively larger portions of the network (unfreeze percentages) and computing the area under the balanced-accuracy vs. unfreeze-percentage curve. This procedure was carried out separately for both full and nodule slices.

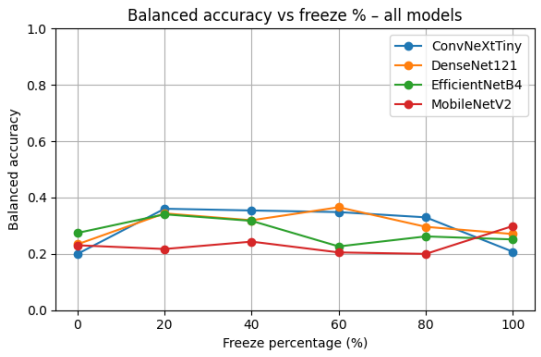

Figure 11: Balanced-accuracy AUC as a function of unfreeze percentage for full slices.

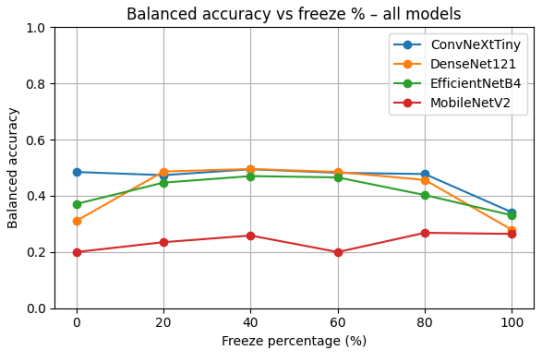

Figure 12: Balanced-accuracy AUC as a function of unfreeze percentage for nodule slices.

Table 4: Comparison of balanced-accuracy AUC (%) values for each backbone on full-slice and nodule-crop datasets.

Model	Full	Nodule
ConvNeXtTiny	**31.92**	**46.81**
DenseNet121	31.54	44.37
EfficientNetB4	28.18	42.73
MobileNetV2	22.61	23.89

Table 4 summarizes the resulting AUC values for each backbone when using a simple linear classifier head. ConvNeXt-Tiny consistently achieved the highest AUC on both datasets, indicating its superior ability to extract discriminative features from lung CT images. Consequently, we selected ConvNeXt-Tiny as our backbone for the development of our final models.

3.2 Addressing Class Imbalance

Class imbalance can bias model training toward majority classes, leading to poor performance on underrepresented categories [4]. To mitigate this, we created a balanced training set offline by upsampling minority classes through data augmentations. Generation of synthetic samples with generative adversarial networks (GANs) was also experimented with.

3.2.1 Upsampling with Data Augmentation

We upsampled each minority class to match the size of the majority class using two augmentation strategies: traditional and advanced (as detailed in Section 3.3). To avoid overfitting, we limited

the upsampling factor to the majority class size; excessive upsampling (e.g., 1.5× majority) risks generating redundant samples without novel anatomical variation [10].

The classification model comprised the best-performing pretrained feature extractor identified earlier, followed by a single dense layer. We performed 3 epochs of transfer learning and 7 epochs of fine-tuning.

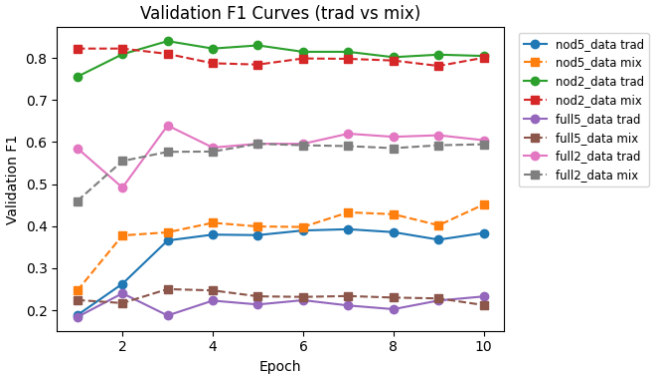

Figure 13: F1-score trajectories during fine-tuning for binary and five-class datasets.

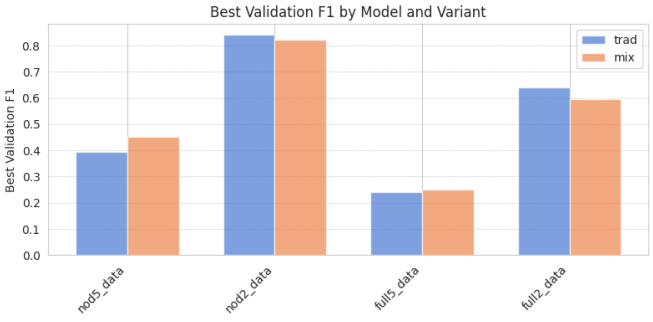

Figure 14: Highest F1-score values during fine-tuning for binary and five-class datasets.

Figure 13 shows F1-score trajectories on validation data: traditional augmentations yielded higher performance on the binary classification task, whereas advanced augmentations performed marginally better on the five-class malignancy scale. Although the overall differences are small, this trend suggests that augmentation complexity should be matched to classification granularity, and insights gained on one task (e.g., five-class) may translate to related tasks (e.g., binary).

It can already be noticed in Figure 14 that the binary classification models outperform the five-class classification ones.

3.2.2 GAN-Based Synthesis

We conducted preliminary experiments with GAN-based minority-sample synthesis but did not pursue this method further due to insufficient data to learn a reliable image manifold (Figure 15 illustrates some clearly failed attempts). Consequently, all final experiments relied solely on augmentation-based upsampling.

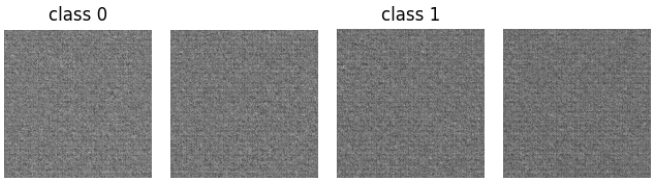

Figure 15: Examples of failed attempts at GAN-based sample synthesis.

3.3 Data Augmentation

Data augmentation was used both to upsample minority classes offline (mitigating class imbalance) and to provide online regularization during training [6]. Augmentation types and their parameters were selected based on literature guidelines.

We applied controlled rotations (±15°) and horizontal flips ($p = 0.5$) to introduce variation while preserving anatomical validity. Because all CT slices share identical orientation (identity direction cosines), these transforms simulate slight patient repositioning and valid left-right lung swaps. Although larger rotations (up to ±90°) can help in some applications, lung-cancer CT scans are already standardized, so limiting rotation to ±15° best balances diversity and realism [3]. Random zooms and crops (scale 0.8–1.0) further mimic scanner variability without distortion [3].

In addition, we incorporated MixUp and CutMix to enhance robustness in CT tasks [8]:

- **MixUp** interpolates pairs of images and their labels, reducing sensitivity to label noise.
- **CutMix** swaps a patch between two images and combines their labels, delivering stronger regularization.

Figure 16 shows these augmentations applied to both full and nodule slices.

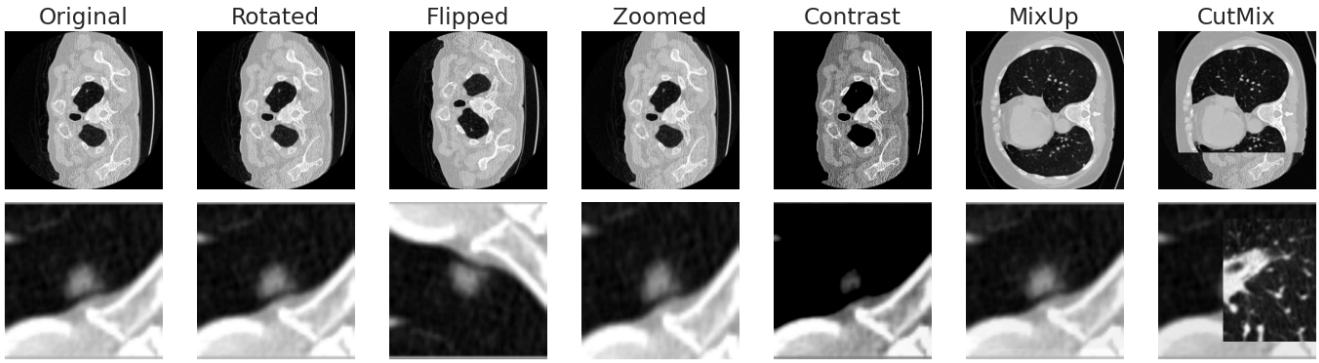

Figure 16: Examples of augmentations applied to a full-slice (above) and a nodule crop (below).

3.3.1 Test-Time Augmentation

Test-time augmentation (TTA) applies the same transformations used during training to each test image, then votes on the resulting model outputs to yield a more robust final prediction. It boosts accuracy and AUPRC without changing the underlying architecture or retraining.

TTA was employed to evaluate our top models and to generate the project's final predictions.

3.4 Hyperparameter Search

Building on prior observations that five-class classification performance translates well to the binary setting, we conducted an extensive hyperparameter search on both full-slice and nodule datasets (five-class labels), applying advanced augmentation upsampling to the training set. Our initial experiments focused on three factors: batch size, optimizer choice, and the architecture of the fully connected classifier head.

3.4.1 Batch Size

Full-slice training works best with a small batch (16 from Table 6), which adds helpful noise that prevents overfitting on its varied images. Nodules, being more alike, train more stably with a slightly larger batch (32 from Table 7) that smooths out gradient updates without losing detail.

Table 6: Comparison of performance metrics by batch size for the full slice dataset.

Batch	Acc	BalAcc	W-F1	W-AUPRC
16	**0.44**	**0.35**	**0.43**	**0.42**
32	0.35	0.30	0.36	0.40
64	0.39	0.32	0.38	0.40

Table 7: Comparison of performance metrics by batch size for the nodule dataset.

Batch	Acc	BalAcc	W-F1	W-AUPRC
16	0.51	0.43	0.49	0.46
32	**0.51**	**0.43**	**0.49**	**0.50**
64	0.49	0.45	0.49	0.49

3.4.2 Optimizer

Adam adapts quickly to the complex patterns in full slices and outperforms its weight-decay variant there (see Table 8). For nodules, however, AdamW's built-in decay keeps the simpler model from fitting noise (see Table 9).

Table 8: Comparison of performance metrics by optimizer for the full slice dataset.

Opt	Acc	BalAcc	W-F1	W-AUPRC
AdamW	0.41	0.28	0.37	0.38
Adam	**0.43**	**0.33**	**0.43**	**0.43**
RMSprop	0.49	0.31	0.41	0.41
SGD	0.06	0.20	0.02	0.31

Table 9: Performance metrics by optimizer for the nodule dataset. AdamW* refers to weighted Adam with 10^{-4} decay.

Opt	Acc	BalAcc	W-F1	W-AUPRC
AdamW	**0.55**	**0.47**	**0.54**	**0.54**
Adam	0.50	0.45	0.50	0.53
AdamW*	0.54	0.43	0.52	0.51
RMSprop	0.47	0.47	0.48	0.48
SGD	0.37	0.43	0.37	0.45

3.4.3 Classifier Head

Full-slice models need a two-layer head (see Table 10) to add enough nonlinear flexibility for separating intricate features. Nodules, by contrast, do better with a single linear layer (see Table 11), since extra depth only risks overfitting.

Table 10: Comparison of performance metrics by dense layer configuration for the full slice dataset.

Head	Acc	BalAcc	W-F1	W-AUPRC
256,128,5	0.43	0.32	0.41	0.40
128,5	**0.42**	**0.31**	**0.40**	**0.41**
5	0.42	0.30	0.39	0.39

Table 11: Performance metrics by dense layer configuration for the nodule dataset.

Head	Acc	BalAcc	W-F1	W-AUPRC
256-128-5	0.42	0.45	0.43	0.51
128-5	0.52	0.46	0.51	0.50
5	**0.56**	**0.47**	**0.54**	**0.54**

3.5 Final Training

Having identified the best batch size, optimizer, and head configuration, we then compared three fine-tuning regimes, i.e. Conservative, Moderate, and Aggressive, each differing in the fraction of backbone layers unfrozen, total epochs, and learning-rate schedule, in order to find the best performing model for each task.

All regimes begin with 20 epochs of head-only training at a learning rate of 1e−3. Table 5 summarizes the subsequent stages: the percentage of backbone layers unfrozen at each step, the number of epochs per stage, and the linear decay of the learning rate from its initial to final value.

A moderate unfreezing of backbone layers lets the full-slice network adjust without losing its pretrained strengths (see Tables 12,13). Nodules, which overfit more easily, benefit from a conservative approach that keeps most layers frozen (see Tables 14,15).

Table 12: Performance of the full slice binary classifier.

Mode	Acc	BalAcc	W-F1	W-AUPRC
Con.	0.78	0.62	0.75	0.77
Mod.	**0.78**	**0.68**	**0.78**	**0.81**
Agg.	0.65	0.54	0.66	0.68

Table 13: Performance of the full slice 5-class classifier.

Mode	Acc	BalAcc	W-F1	W-AUPRC
Con.	0.31	0.29	0.32	0.36
Mod.	**0.43**	**0.29**	**0.39**	**0.41**
Agg.	0.45	0.22	0.32	0.35

Table 14: Performance of the nodule binary classifier.

Mode	Acc	BalAcc	W-F1	W-AUPRC
Con.	**0.82**	**0.76**	**0.82**	**0.88**
Mod.	0.84	0.76	0.84	0.86
Agg.	0.85	0.75	0.84	0.87

Table 5: Subsequent fine-tuning stages after an initial 20 epochs at 1e−3 head-only. Each mode unfreezes the backbone by the listed percentages, for the given epochs, with LR decaying linearly from start to end.

Mode	Unfreeze (%)	Epochs	LR (start → end)
Conservative	30% → 50% → 70% → 90%	[25,25,30,30]	5e−5 → 5e−6
Moderate	50% → 90%	[25,30]	5e−4 → 5e−5
Aggressive	90%	[30]	1e−3 → 1e−4

Table 15: Performance of the nodule 5-class classifier.

Mode	Acc	BalAcc	W-F1	W-AUPRC
Con.	**0.57**	**0.49**	**0.55**	**0.55**
Mod.	0.46	0.44	0.46	0.54
Agg.	0.48	0.50	0.48	0.47

Figure 17 shows the marginal boost provided by TTA in W-AUPRC across all architectures.

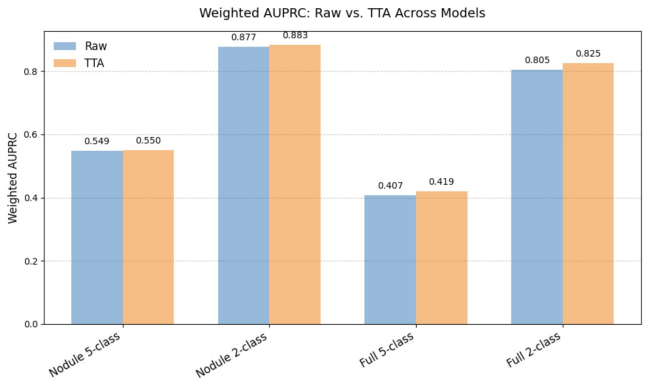

Figure 17: Comparison of raw predictions versus TTA-augmented predictions, measured by weighted AUPRC.

The CT-slice and nodule datasets used for training, together with our final trained models, are publicly available `here`.

3.6 Explainability & Grad-CAM++

Gradient-weighted class activation mapping++ (Grad-CAM++) produces class-specific heatmaps by weighting the final convolutional feature maps with the positive gradients of the target score, upsamples them, and overlays the result on the original CT slice.

This visualization was employed to confirm that each model attends to clinically relevant regions. The figures below present Grad-CAM++ outputs for successful predictions of each model type.

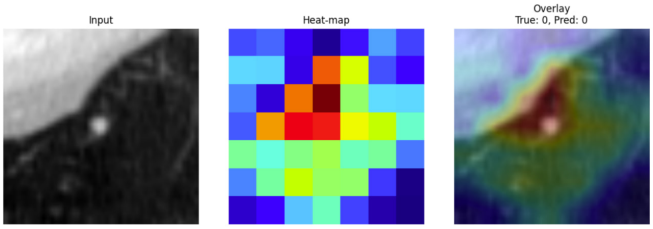

Figure 18: Nodule binary classifier (Grad-CAM++).

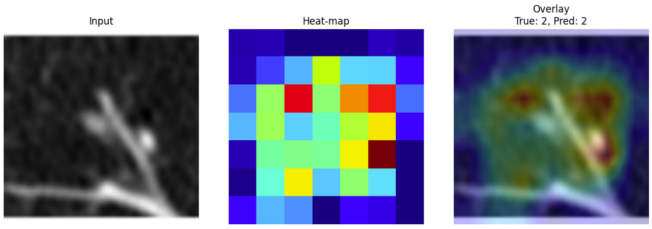

Figure 19: Nodule 5-class classifier (Grad-CAM++).

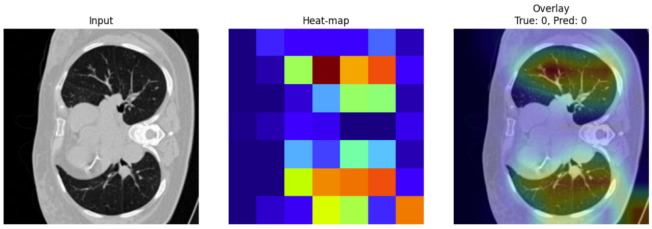

Figure 20: Full-slice binary classifier (Grad-CAM++).

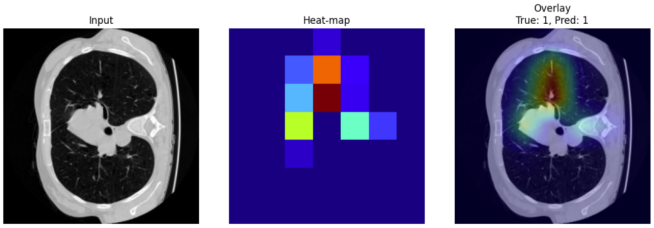

Figure 21: Full-slice 5-class classifier (Grad-CAM++).

4 Discussion

The classification head's inability to establish clear boundaries among five malignancy levels was principally driven by suboptimal feature representations from generic ImageNet-pretrained backbones. t-SNE projections performed on these pretrained CNNs before fine-tuning revealed weak clustering of samples by malignancy grade, underscoring the need for medical-domain pretraining to initialize more diagnostically relevant embeddings. Limited sample size and class imbalance further challenged feature learning. Models trained on nodule-crop slices achieved marginally higher scores than full-slice models, as cropping reduced background variability and directed the feature extractor toward lesion-specific details. Detailed class-specific evaluation metrics for the five-class task (e.g. micro- and macro-averages, confusion matrix analysis) were not reported, limiting insight into misclassification patterns.

Hyperparameter optimization was extensive for network architecture, batch size, optimizer type, and classifier head architecture, but did not encompass

data augmentation hyperparameters, relying on generic best practices rather than being tailored. The absence of k-fold cross-validation prevented assessment of embedding robustness across data splits. Finally, retraining on the full dataset after hyperparameter tuning was not performed, foregoing refinement of feature encodings with all available samples.

5 Conclusion

Four CNN classifiers were fine-tuned on full-slice and nodule-crop CT images for binary and five-class malignancy prediction. ConvNeXt-Tiny achieved weighted AUPRCs of 0.88 for nodule-crop binary classification and 0.81 for full-slice. Five class-tasks yielded weighted AUPRCs of 0.55 for nodules and 0.41 for full slices, reflecting challenges in fine-grained scoring starting from generic pretrained features. Key limitations included reliance on non-medical pretraining, absence of cross-validation, and lack of full-dataset retraining. Future work should employ medical-domain backbone pretraining, expand hyperparameter searches to augmentation, report comprehensive class-specific performance metrics, implement cross-validation, and retrain on all samples to improve feature discriminability and clinical applicability.

References

[1] M. M. Ansari, S. Kumar, U. Tariq, M. Belal Bin Heyat, F. Akhtar, M. A. Bin Hayat, E. Sayeed, S. Parveen, and D. Pomary. Evaluating CNN Architectures and Hyperparameter Tuning for Enhanced Lung Cancer Detection Using Transfer Learning. *Journal of Electrical and Computer Engineering*, 2024(1):3790617, Jan. 2024.

[2] W. Chen, Y. Wang, D. Tian, and Y. Yao. CT Lung Nodule Segmentation: A Comparative Study of Data Preprocessing and Deep Learning Models. *IEEE Access*, 11:34925–34931, 2023.

[3] E. Goceri. Medical image data augmentation: Techniques, comparisons and interpretations. *Artificial Intelligence Review*, 56(11):12561–12605, Nov. 2023.

[4] Haibo He and E. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, Sept. 2009.

[5] S. Masoudi, S. A. Harmon, S. Mehralivand, S. M. Walker, H. Raviprakash, U. Bagci, P. L. Choyke, and B. Turkbey. Quick guide on radiology image pre-processing for deep learning applications in prostate cancer research. *Journal of Medical Imaging*, 8(01), Jan. 2021.

[6] L. Perez and J. Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning, 2017.

[7] A. Ranjbaran, A.-H. Hassan, M. Jafarpour, and B. Ranjbaran. A laplacian based image filtering using switching noise detector. *SpringerPlus*, 4:119, 2015.

[8] A. Rao, J.-Y. Lee, and O. Aalami. Studying the Impact of Augmentations on Medical Confidence Calibration. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2454–2464, Paris, France, Oct. 2023. IEEE.

[9] M. Rawashdeh, M. A. Obaidat, M. Abouali, D. E. Salhi, and K. Thakur. An Effective Lung Cancer Diagnosis Model Using Pre-Trained CNNs. *Computer Modeling in Engineering & Sciences*, 143(1):1129–1155, 2025.

[10] C. Shorten and T. M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60, Dec. 2019.

[11] M. E. Tschuchnig and M. Gadermayr. Anomaly detection in medical imaging – a mini review, 2021.

[12] J. Wang, N. Sourlos, S. Zheng, N. Van Der Velden, G. J. Pelgrim, R. Vliegenthart, and P. Van Ooijen. Preparing CT imaging datasets for deep learning in lung nodule analysis: Insights from four well-known datasets. *Heliyon*, 9(6):e17104, June 2023.

[13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? 2014.