

Automated classification of intraductal carcinoma of the prostate using deep learning on non-linear multiphoton microscopy images

James Yap [101276054],^{1*} Ajaan Nalliah [101325463],¹ Vipra Gajera [101271761]¹

¹COMP 4107 (Winter 2026), Carleton University, 125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada

Project code: <https://github.com/2026W-COMP4107/project-group-5/commits/main/>

ABSTRACT

Intraductal carcinoma of the prostate (IDC-P) is an aggressive form of prostate cancer where cancer cells grow within existing prostate ducts and acini, often associated with high-grade, advanced-stage disease. Research by the LAMPE Lab achieved great results detecting IDC-P by applying texture-based analysis paired with a support vector machine (SVM) on multimodal nonlinear optical images. This project applies modern Deep Learning methodologies on the LAMPE data set, with the goal of automating feature selection during analysis, and to increase explainability of black-box models by generating attention heatmaps to highlight regions and structures in the images with the most influence on the final classification result. Practical techniques such as Transfer Learning and Consistent Rank Logits (CORAL) were explored to improve classification accuracy.

Key words: Neural Network – Ordinal Classification – Transfer Learning

1 INTRODUCTION

The effectiveness of deep representation learning in high-stakes medical diagnostics, particularly for classifying Intraductal Carcinoma of the Prostate (IDC-P), an aggressive subtype of prostate cancer, is studied. Our goal was to evaluate whether Convolutional Neural Networks (CNNs) could autonomously identify discriminative spatial features in multimodal nonlinear optical (NLO) imaging more effectively than handcrafted texture-based analysis. The original approach used by LAMPE researchers involved extracting first and second-order statistics from a Gray-Level Co-occurrence Matrix (GLCM), which is then interpreted by a Support Vector Machine (SVM). We implemented our own deep learning pipeline on the same data set used in the original study to compare the results of deep learning methods on multimodal data. Three distinct but coregistered imaging channels: collagen, protein, and lipid, were used within a single multimodal input tensor to allow the model to learn complex biochemical relationships that are difficult to define manually.

2 DATASET

This project was done in collaboration with the LAMPE Lab of the Carleton University Physics Department. The dataset includes labeled images of Healthy, Low Grade Carcinoma (LGC), High Grade Carcinoma (HGC), and Intraductal Carcinoma of the Prostate (IDC-P) tissue samples from a cohort of 50 patients who participated in the Canadian Prostate Genome Project (CPC-GENE). This project utilizes co-registered Stimulated Raman Scattering (SRS) and Second Harmonic Generation (SHG) microscopy to provide complementary biochemical and structural data. While SRS images lipids at

Table 1. Dataset details before segmentation, augmentation, or deletion of empty patches.

FOV size:	210 μm $\times$ 210 μm
FOV resolution:	512 $\times$ 512
Channels:	3 (1450 cm^{-1} SRS, 1668 cm^{-1} SRS, SHG)
Total FOVs:	201

Table 2. Class distribution of LAMPE Lab data set. Patients are uniquely identified by the position of biopsy cores on the glass slides. A total of 5 glass slides, each containing a 6×9 array of prostate cores, except for the fifth slide, which had a 7×12 array (Gagnon et al. 2025). For example, “1_C2” refers to the biopsy core on slide 1, row C, column 2.

Index	Label	# FOVs	Indices	Patients
0	Healthy	13 FOVs	[0 - 12]	1_C2, 5_C2, 5_D6, 5_E2
1	LGC	34 FOVs	[13 - 46]	1_B3, 1_C4, ... (16 total)
2	HGC	84 FOVs	[47 - 130]	1_B2, 1_B3, ... (28 total)
3	IDC	70 FOVs	[131 - 200]	1_B1, 1_B3, ... (18 total)

1450 cm^{-1} and proteins at 1668 cm^{-1} , SHG specifically maps non-centrosymmetric structures like collagen fibers.

2.1 Multimodality & Coregistration

The three imaging channels listed in Table 1 are coregistered and are hence superimposable. Each channel provides information about different specific tissue types, as explained in Section 2. This per-tissue isolation is a theoretical advantage of multiphoton microscopy over the more traditional H&E-stained brightfield microscopy approach. For example, a classifier may learn to apply varying weights to images of each tissue type situationally. Additionally, Gagnon et al. (2025) found that chemical information captured in SRS, which is known to be important in classifying IDC-P, is not visible in H&E.

* E-mail: contact@jamesyap.org

2.2 Resonant Background Subtraction

The SRS data set images in (Gagnon et al. 2025) included vibrationally resonant Raman signals and non-vibrationally resonant background signals, which arise due to optical effects such as cross-phase modulation, two-photon absorption, and thermal lensing. The original authors had computed background-subtracted images using these on-Raman and off-Raman resonant frequency bands.

statistical test to see if off-res matters

3 MEASURES OF PERFORMANCE

3.1 Precision, Accuracy, ROC AUC

The initial approach was to treat this project as an ordinal regression task of four classes. The classes increased in severity: Healthy $\rightarrow$ LGC $\rightarrow$ HGC $\rightarrow$ IDC-P. The `sklearn.metrics.classification_report` utility was used to compute performance metrics such as precision, recall (or true positive rate), F1 score, Receiver Operating Characteristic Area Under the Curve (ROC AUC), and accuracy for every fold taken during training and evaluation. Due to the class imbalance in the data set, more focus is given to the macro and weighted averages, as well as the AUC value. Standard statistical tests were then leveraged to discriminate between hyperparameter choices as discussed in Appendix A.

In a later iteration, however, discussions with the author (Justin Gagnon) of Gagnon et al. (2025) revealed that the mathematical relationship of IDC-P data points need not be ordinal in the latent space. Instead, IDC-P is a spatial growth pattern that is not simply a "higher-grade" carcinoma. While the Healthy, LGC, and HGC classes are naturally ordinal, IDC-P is a biological outlier almost always co-exist with High-Grade invasive cancer but is morphologically distinct (Guo & Epstein 2006).

3.2 Ordinal Regression vs. Anomaly Detection

Consequently, our analysis was pivoted away from ordinal classification for the reasons discussed above. A different approach of treating this as an "anomaly detection" task was tested, where the primary goal became to identify minority IDC-P instances from the majority LGC and HGC classes.

Multiple other benefits arose from this change in scope. Namely, the Healthy class consists of only four unique biopsy cores. This severely limits the number of splits that can be done during K-fold cross validation to prevent data leakage and intra-core bias (see Table 2). By focusing on discriminating IDC-P variants from regular high/low grade carcinoma, we eliminated the Healthy class bottleneck. Additionally, modern techniques such as Focal Loss can be applied to penalize false-negative IDC-P classifications at a higher proportion.

Methodologies and results obtained from both ordinal classification and binary anomaly detection approaches are included in the following sections for completeness.

3.3 Recall

False negatives are significantly more expensive than false positives, as pathologists can easily dismiss false positives during diagnosis. On the other hand, failing to diagnose patients with IDC-P results in a detrimental delay in the treatment timeline, leading to fatalities that could have been prevented and caught earlier. As a result, recall of

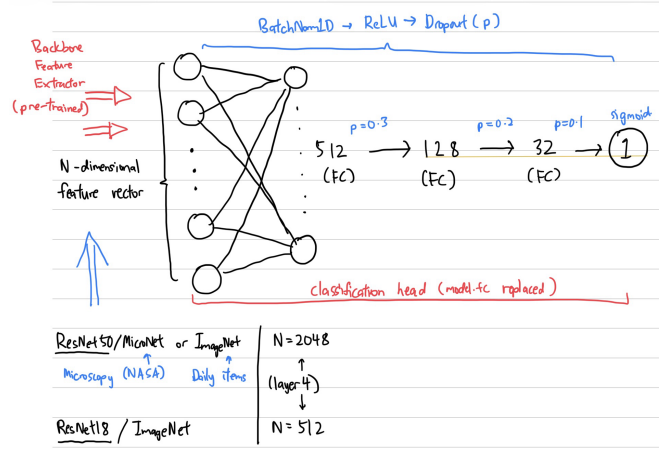


Figure 1. Enter Caption

IDC-P was prioritized as the primary measure of performance. The recall equation is given by equation (2).

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{All Positives}} \quad (1)$$

$$= \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (2)$$

4 METHODS, OBSERVATIONS, SIMULATIONS ETC.

4.1 Backbone Feature Extractors

Due to the small size of the acquired dataset, we used transfer learning with various backbone CNNs as feature extractors. We chose light-weight architectures such as ResNet18, VGG-16, and VGG-19. The classifier head is then replaced with custom fully connected layers (see Fig. 1). The choice of these backbone CNNs was heavily motivated by the findings obtained in a similar study on ovarian cancer tissue by Huttunen et al. (2018). Phased unfreezing (see Section 4.2) was implemented to fine-tune the backbone feature extractor model. The learning rate used on the unfrozen feature layers in the backbone model is taken as 10% of the learning rate used on the fully connected classifier layer. This was done to prevent catastrophic forgetting of learned parameters in the backbone model.

Fig. 18 shows the results obtained using various backbone CNN feature extractors. For rapid iteration, an SVM classification head was used to evaluate the relative performance of various backbone feature extractors, CNNs and ViTs alike. For why SVMs were used for backbone evaluation instead of FC classification heads, see Section 4.12 for justification. The main reason is due to computational speed and negligible performance differences. Once the best backbone feature extractor is identified, the SVM is replaced with an FC classifier.

Additional backbone feature extractors tested include ResNet50/ImageNet (He et al. 2015), ResNet50/MicroNet by NASA (Stuckner et al. 2022), ResNet18/ImageNet (Allena 2022), DINOv2_ViT-S/14 by Meta (Oquab et al. 2023; Darcet et al. 2023; Jose et al. 2024), ConvNeXt, EfficientNetV2, and owkin/phikon-v2 (Filiot et al. 2024). See Fig. 18 for numeric results.

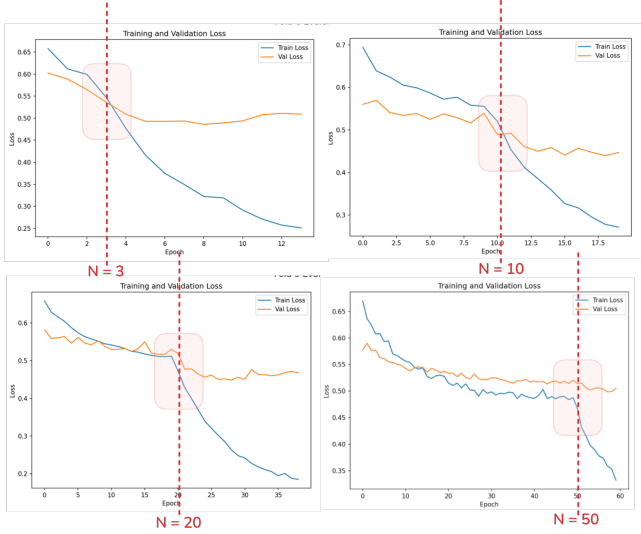


Figure 2. Training and validation loss when unfreezing ResNet18 (layer4) at epoch numbers 3, 10, 20, and 50 until early stopping is triggered. Severe overfitting responses is visible immediately after unfreezing. Train loss collapses to 0 while validation loss stays the same or even increases.

4.2 Phased Unfreezing

To use ImageNet-trained ResNet18 as a feature extractor, the CNN backbone was frozen during training. However, the final stage of residual blocks known as `layer4` in the PyTorch implementation was unfrozen to allow the model to learn data set specific features. This is because the original ImageNet labels comprise general items such as "mammal", "dog", or "vehicle", as seen in [Deng et al. \(2009\)](#). By contrast, the data set used in this project is based on microscopy images taken from biopsy cores. We had also replaced its classification head with a custom fully connected (FC) layer, tailored to the 4 ordinal classes that we expect to produce: Healthy, LGC, HGC, IDC-P.

Initial training efforts yielded rapidly decreasing training losses, but validation losses remained high, and even increased at times. This indicated overfitting — it was then theorized that the over-powerful model was blindly memorizing noise in the training set, even with a light-weight architecture. To combat this, phased unfreezing of `layer4` was implemented. Fig. 2 shows the train/val loss curves when `layer4` is unfrozen at various N epochs. Training continued until validation loss stops decreasing for a specified number of epochs (known as early stopping with patience).

4.3 Group K-Fold with Stratification

The LAMPE data set provides unique identifiers for each FOV image. One biopsy core is associated with one unique patient, with one or more FOV images taken from each biopsy core. The biopsy cores are uniquely identified by their position on the glass slide during imaging. Group K-Fold from scikit-learn was used to ensure all FOV images from a single patient stays entirely either in the train set or the validation set, but not both. This is to verify that the model is not memorizing patient-specific patterns and is able to generalize to unseen biopsy cores, as will be required in field applications.

Additionally, stratification is applied to encourage an even representation from each class. This avoids over-representation from one

```
import albumentations as A
from albumentations.pytorch import ToTensorV2

self.transform = A.Compose([
    # A.ElasticTransform(
    #     alpha=10,
    #     sigma=6,
    #     # alpha_affine=100 * 0.03,
    #     border_mode=cv2.BORDER_REFLECT_101,
    #     p=1.0,
    # ),
    A.RandomRotate90(p=0.5),
    A.HorizontalFlip(p=0.5),
    A.VerticalFlip(p=0.5),
    ToTensorV2() # Converts back to (C, H, W) and makes it a Tensor
])
```

Figure 3. The Albumentations library was used for its image augmentation capabilities. The code snippet above shows an example usage within the PyTorch Dataset class.

class and under-representation from the other during sampling. See Appendix A for more information.

4.4 AdaBoost (Error Correction), Bagging, EasyEnsemble

`imblearn.ensemble.EasyEnsembleClassifier` is an ensemble method provided by scikit-learn. Ensembles attempt to build better learning algorithms by combining multiple simpler algorithms ([Schapire 2013](#)). EasyEnsemble uses AdaBoost with error correction and bagging to build a strong classifier from many weak classifiers. Since neural networks are considered strong learners, they are more at risk of overfitting, which boosting may worsen. However, given the class imbalance even in the binary task of HGC vs IDC, EasyEnsemble with bagging can randomly under-sample the majority class and train multiple models. Each model sees the full minority class and a unique sub-portion of the majority class.

Using the approach described above, class imbalance is no longer detrimental. We could even combine the LGC and HGC classes, turning the problem into detecting anomalous IDC patterns in a flood of non-IDC data points. This point may be investigated in detail in future work.

4.5 Affine Geometric Augmentation

There are 28 unique biopsy cores in the majority class (HGC), and only 4 in the minority class (Healthy), as seen in Table 2. The class imbalance is further exaggerated at the sub-image level, blowing up the 1 : 7 imbalance ratio up to 1 : 12. This is because many more empty images with insufficient collagen imaged by SHG are discarded from the "Healthy" class compared to the "HGC" class.

The resolution of full FOV images are 512×512 (see Table 1). ResNet18, however, expects an input shape of 224×224 ([Al-lena 2022](#)). Therefore, stateless, probabilistic affine augmentation is paired with PyTorch's `WeightedRandomSampler` to balance out the four classes. This includes taking random crops and rotations at a continuous interval. Around $\sigma = 0.02$ units of Gaussian noise was also added to expand the augmentation space to infinity. Effectively, no data points will be duplicated exactly.

Fig. 4 shows an overly jagged and erratic training loss curve, where validation loss is smooth and decreases exponentially. Since augmentation and regularization like Dropout are only active during training, this is evidence that the amount of augmentation required

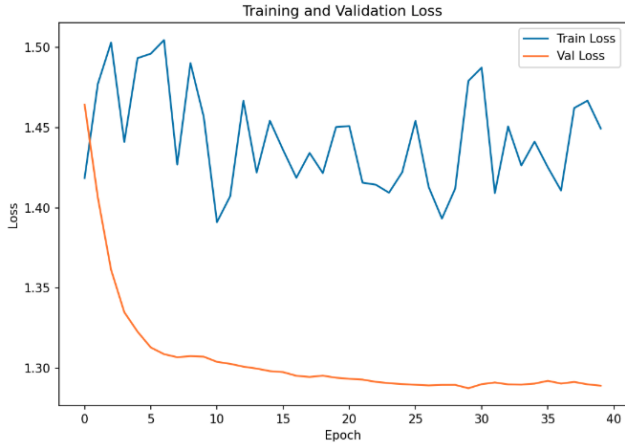


Figure 4. Loss curves with heavy augmentation to counteract 1:12 class imbalance. Validation loss is a smooth curve which decreases exponentially, and eventually plateaus. Training loss is extremely jagged and highly erratic. This is evidence of overwhelming augmentation and regularization, active only during training.

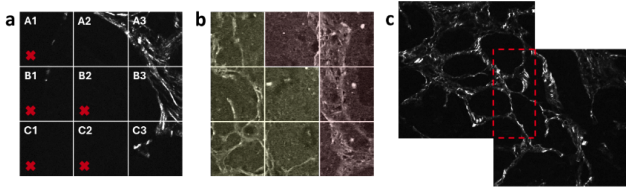


Figure 5. SHG and SRS fields of view before segmentation. a) An SHG image segmented into 3×3 , with patches discarded due to insufficient collagen. b) An FOV images that contains more than one tissue type — HGC (yellow) and IDC (red). c) Two FOVs that are overlapping. Reprinted from [Gagnon et al. \(2025\)](#).

to balance out the classes are overwhelming and fail to represent the true underlying distribution.

4.6 Patches

The LAMPE researches pre-processed their data set by dividing each FOV image into 3×3 patches ([Gagnon et al. 2025](#)). These patches were then manually reviewed and discarded if their SHG channels contained insufficient information (see Fig. 5). We investigated the performance of using full FOV images versus using sub-images as inputs to the CNN feature extractor. See Table 5 and Section 4.8 for analysis of results.

4.7 Robust Min-Max

Robust MinMax is a custom utility developed to perform percentile-based noise removal. It works in conjunction with resonant background subtraction (see Section 2.2) to increase the Signal-to-Noise (SNR) ratio while eliminating outliers. Later experiments found that Robust MinMax should only be applied on post-hoc visualizations and should never be used as a pre-processing step. This is because Robust MinMax is applied on the patch/sub-image level, and destroys and relative magnitudes between samples. Dataset-wide Z-score stan-

```
def robust_minmax(arr: np.ndarray, p_min=5.0, p_max=99.5, axis=None) -> np.ndarray:
    """
    Normalizes using percentiles to ignore outliers, supporting axis-wise scaling.

    Args:
        arr: Input array to normalize.
        p_min: Lower percentile.
        p_max: Upper percentile.
        axis: Axis or axes along which to compute percentiles.
            e.g., (1, 2) for per-channel scaling on (C, H, W).
    """
    # 1. Compute percentiles. keepdims=True is crucial for broadcasting!
    lo = np.percentile(arr, p_min, axis=axis, keepdims=True)
    hi = np.percentile(arr, p_max, axis=axis, keepdims=True)

    diff = hi - lo

    # 2. Handle the case where the range is near zero to avoid DivisionByZero
    # We use np.where to keep the logic vectorized
    normalized = np.where(diff < 1e-8, 0.0, (arr - lo) / diff)

    # 3. Clip and ensure float32
    return np.clip(normalized, 0.0, 1.0).astype(np.float32)
```

Figure 6. Implementation of the robust_minmax utility. Crucially, this utility is only used for post-hoc visualizations and should never be applied as a pre-processing step. Dataset-wide Z-score standardization should be used instead. Applying robust_minmax at the patch level has been shown to ruin key features and severely degrade model performance, as seen in Section 5

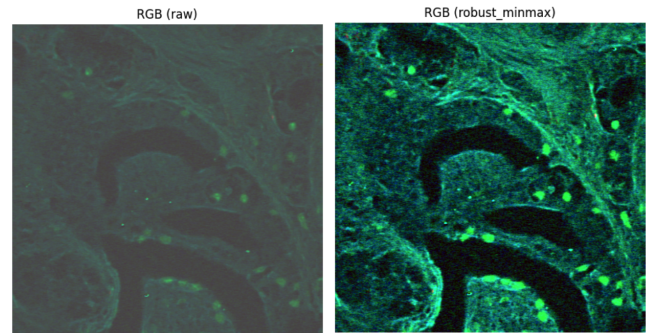


Figure 7. Visual sample of robust_minmax applied to FOV image #110. Visibility is increased, with outliers clipped using percentile-based noise removal.

dardization supersedes Robust MinMax during pre-processing (see equation 3).

$$z = \frac{x - \mu}{\sigma} \quad (3)$$

4.8 AB-MIL

Attention-based Deep Multiple Instance Learning (AB-MIL) is a variation of supervised learning where a single class label is assigned to a bag of instances. As seen in Section 4.6, we experimented with breaking down full FOV images into smaller sub-images called "patches" to better fit the expected input size of the ResNet architecture. As a side effect, the number of data points is increased. Combined with the fact that [Ilse et al. \(2018\)](#) benchmarked performance gains on two real-life histopathology datasets similar to LAMPE, AB-MIL made for a great choice to experiment with in this project.

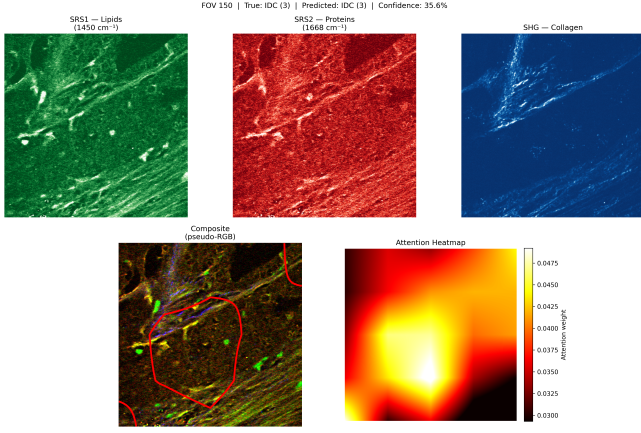


Figure 8. An example heatmap generated by using the AB-MIL approach. This approach was eventually abandoned due to bad performance and inaccurate heatmaps. Grad-CAM (see Section 4.13) supersedes AB-MIL in a future iteration for model explainability. As explained in Section 4.6, AB-MIL performs worse because sub-images do not contain enough detectable structural information in contrast to full FOV images. AB-MIL treats each sub-image independently and does not examine the spatial relationships between patches. In future work, a deep-dive comparison could be made to the original work of Gagnon et al. (2025) to figure out why sub-image classification worked so well for their texture-based analysis compared to our deep learning approach. A reasonable starting point would be to investigate the training dataset for the backbone feature extractors used in this project, i.e., ImageNet by Stanford and MicroNet by NASA.

4.9 Self-Attention and ViT

The possibility of using Self-Attention-based architectures such as TransMIL and Vision Transformers (ViT) was also considered. These architectures were engineered to solve the weak supervision problem that arises when overlapping tissue from different classes appear within the same FOV data point. This introduces noise and impacts the separability of the training set, harming the model's ability to learn concise decision boundaries between the 4 ordinal class labels.

However, TransMIL specializes in dealing with whole slide images (WSI) (Shao et al. 2021) that are typically in the gigapixel (10^9 pixels) resolution range with FOVs of 15×15 mm (Jiang et al. 2023). By contrast, the FOV size used in our data set is $210 \mu\text{m} \times 210 \mu\text{m}$, with images constructed by averaging 10 successfully imaged frames at a resolution of 512×512 (Gagnon et al. 2025). Additionally, most pre-trained models specializing in pathology are trained on H&E-stained brightfield microscopy images, in contrast to the label-free non-linear multiphoton approach used in our data set. These significant discrepancies eliminated the suitability of TransMIL and similar architectures for our use case.

4.10 K-1 Ordinal

K-1 ordinal regression was used to exploit the inherent ordering of classes in the LAMPE data set. $K - 1 = 3$ neurons were used in the output layer to classify the ordered severity of cancer in the input image. Table 3 shows the mapping used in this technique.

From empirical results obtained over multiple trials, this approach exhibited the "ordinal hedging" effect. The model overwhelmingly preferred the middle classes like LGC and HGC, and avoids the extremes (Healthy and IDC). Initially, this phenomenon was believed to be caused by a class imbalance in the data set. However, this issue

Table 3. K-1 ordered mapping of output layer activations for LAMPE data set classes. This approach uses three neurons in the output layer to discriminate between the four classes.

Class	Activations	Notes
Healthy	[0, 0, 0]	Not worse than any.
LGC	[1, 0, 0]	Worse than Healthy.
HGC	[1, 1, 0]	Worse than Healthy and LGC.
IDC-P	[1, 1, 1]	Worse than everything.

persisted even after heavy geometric augmentation to counteract the class imbalance. It is possible that the geometric augmentation process was unable to expand the data set size in a way that faithfully represents the true underlying distribution due to bias.

More importantly, extreme predictions are statistically punished. To predict "Healthy" ([0, 0, 0]) or "IDC" ([1, 1, 1]), the model must get all three independent classifiers to align on the correct side of the 0.5 threshold used during inference. Given standard variance and noise, it is highly probable that at least one logit flips incorrectly, causing a collapse to the middle.

Additionally, the model frequently violates the monotonicity of the ordinal target. For example, predictions such as [0, 1, 0] imply that a given data point is more severe than "LGC" but less severe than "Healthy", which is illogical. This happens because there is no mathematical guarantee to force monotonicity in the predictions, and is a common problem with most ordinal binary decomposition approaches (Lázaro & Figueiras-Vidal 2023a).

4.11 CORAL, CORN, BONN

The ordinal hedging effect due to statistically difficult extreme predictions and monotonicity violation mentioned in Section 4.10 could be addressed using Consistent Ranked Logits (Cao et al. 2020), Rank-Consistent Ordinal Regression (Shi et al. 2023), or with Bayesian Ordinal Neural Networks (Lázaro & Figueiras-Vidal 2023b).

Consistent Ranked Logits (CORAL) achieves rank consistency among its output layer tasks by imposing a weight-sharing constraint on the output neurons, forcing the model to learn optimal bias terms (Cao et al. 2020). These bias units then act as monotonic decision boundaries.

Weight sharing reduces the expressibility of CORAL regression units, acting as an aggressive form of regularization. Ranked Consistent Ordinal Regression (CORN) achieves rank consistency by estimating a series of conditional probabilities using conditional training subsets, and then applying the chain rule of probability to calculate the final unconditional rank probabilities (Shi et al. 2023). This way, CORN achieves rank consistency without imposing restrictive weight-sharing constraints, increasing the expressive power of the model.

Bayesian Ordinal Neural Networks (BONN) condenses data points onto a one-dimensional output space using a single output neuron, enforcing a monotonic decision rule to match the ordinal structure of the corresponding pattern (Lázaro & Figueiras-Vidal 2023b). BONNs also deal with class imbalance using the Bayesian Decision Theory, providing a unified solution to solve both problems highlighted in Section 4.10 natively. However, BONN introduces a novel loss function, which, due to its heavy mathematical rigor, has been excluded from the scope of this project. Here, we deal with class imbalance and the ordinality of the data in two independent and separate parts of our analytical pipeline. A future implementation may revisit this method for its potential.

4.12 Support Vector Machines (SVM) & Principal Component Analysis (PCA)

The original paper's by [Gagnon et al. \(2025\)](#) found great success using texture-based analysis + SVM approach. Robust techniques such as the Error-Correcting Output Codes (ECOC) with Hamming distance was used to improve SVM performance, especially on multi-label classification tasks.

We investigated the effectiveness of SVMs by building a hybrid model. This model uses a pre-trained CNN feature extractor to convert images into flat vectors of condensed features in a latent space. An SVM is then fitted to these feature vectors and their labels. Techniques such as geometric augmentation and StratifiedGroupKFold were also applied.

The key difference is that we have now replaced the fully connected (FC) classification head of the backbone CNN with an SVM. The ResNet50 backbone produces a 2048-dimensional vector, while ResNet18 produces a 512-dimensional vector. Principal Component Analysis (PCA) is then applied to further reduce the effective dimension of the latent space. An SVM is then fitted to these labelled, compressed representations. `sklearn.svm.SVC` and `sklearn.decomposition.PCA` from scikit-learn were used for this task.

The hypothesis is that SVMs should perform better on sparse data sets. SVMs create a mapping with a wide as possible gap between separating categories. Data points closest to the gap are called "support vectors". Other points in the train set are then discarded. Thus, SVMs should, in theory, efficiently handle sparse data. Another benefit to using SVMs is the implementation and computation simplicity. scikit-learn provides an extremely ergonomic interface for using SVMs and PCA, including parameters such as `class_weight='balanced'` to deal with class imbalance. From our anecdotal experience, SVM fitting by scikit-learn takes only a couple of seconds, while training a monolithic CNN with fully connected classifiers takes some ~10 minutes to complete. This computational efficiency enables rapid iteration, greatly reducing the friction of hyperparameter tuning during the pre-processing and feature extraction steps.

Our tests revealed that SVMs provided no visible accuracy improvements over FC classifiers. However, as explained in Section 3.3, we prioritized recall as the primary measure of performance. Our empirical tests show that SVMs consistently achieved a higher recall score for the anomalous IDC class than FC classifiers. This, and the fact that SVMs computationally cheaper, make them a viable option for serious consideration (see Fig. 9).

4.13 Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) is a gradient-based feature attribution method that is used to produce visual explanations for decisions from CNN-based models ([Selvaraju et al. 2020](#)). It uses gradients flowing into the final convolutional layer to produce a coarse localization map highlighting important regions in the image during classification. For this project, the Grad-CAM implementation for PyTorch used is due to [Gildenblat & contributors \(2021\)](#).

Fig. 10 and Fig. 11 are examples of Grad-CAM heatmaps on IDC classes, while Fig. 12 shows an HGC instance. For a trained pathologist, these activation maps may provide explanation and establish appropriate trust in the predictions of the model. As explained in Section 3.1 and Section 3.2, IDC-P samples are morphologically distinct from regular prostate cancer tissue. Hence, Grad-CAM pro-

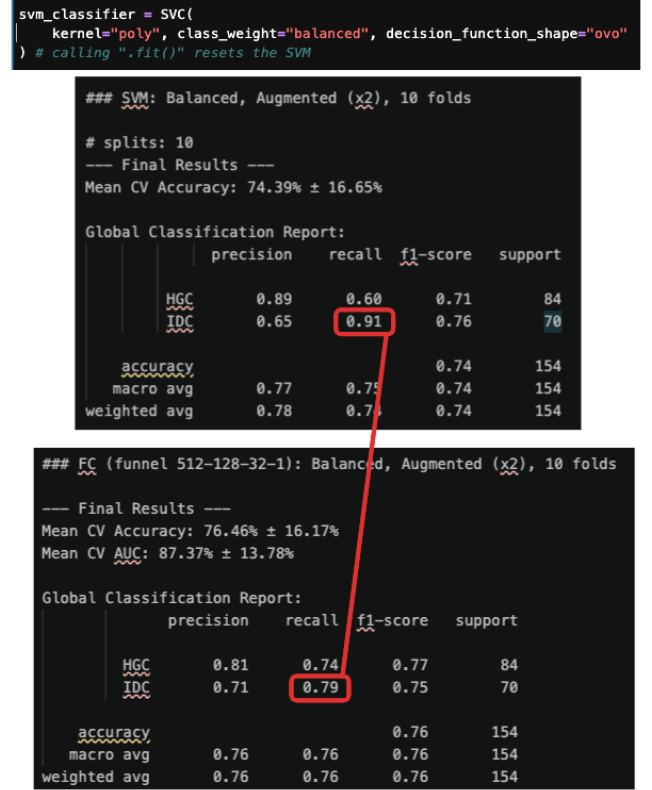


Figure 9. Tables comparing an SVM classifier with an FC classifier. The accuracy difference between the two are not statistically significant, but SVMs clearly shows higher recall scores on IDC than FC classifiers.

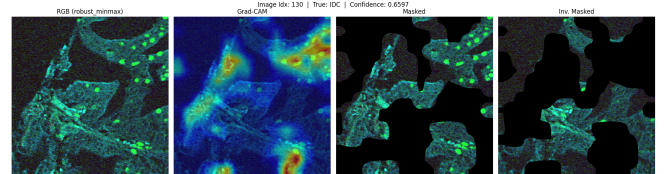


Figure 10. FOV image at index 130 with Grad-CAM heatmap. True label: IDC, predicted: IDC with 65.97% confidence.

vides visualizations of these discriminatory features that the trained model uses to detect anomalous IDC-P signatures.

We are not trained pathologists, so qualitative analysis on these results are non-trivial. However, looking at Fig. 10, the heatmap appears to highlight regions in the image which exhibit unique structural difference from HGC patterns. In a future work, a trained pathologist could inspect these heatmaps to comment on the models performance.

5 RESULTS

Fig. 14 shows the loss curves over a 10-fold run. The red dots mark positions at which validation loss is at the lowest — these are the points at which the model states are saved. Fig. 15 shows the loss curves of the four-class labeling task. Fig. 17 and Table 4 compare the performance of various architecture, hyperparameter, and pre-

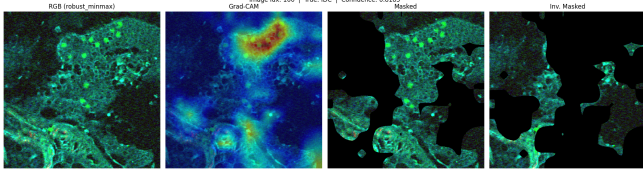


Figure 11. FOV image at index 100 with Grad-CAM heatmap. True label: IDC, predicted: IDC with 81.85% confidence.

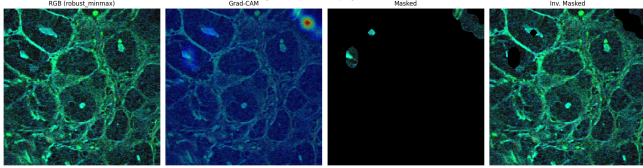


Figure 12. FOV image at index 50 with Grad-CAM heatmap. True label: HGC, predicted: HGC with 77.16% confidence.

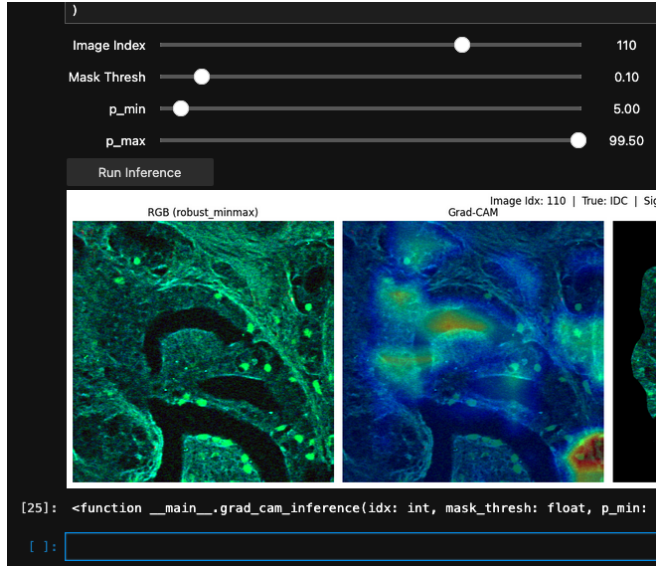


Figure 13. A Jupyter Notebook utility that provides interactive sliders and buttons to run one-click inference on the dataset.

processing choices. Fig. 18 and Table 18 compare the performance of various SVM parameters and backbone feature extractor choices.

6 CONCLUSIONS

This project shows that deep learning is a viable technique for detecting IDC-P patterns among regular HGC images. Using texture-based analysis and an SVM, Gagnon et al. (2025) achieved mean accuracies of 94.2% with subimages and 96.7% with FOV majority-consensus. By contrast, our deep learning approach achieved a mean accuracy of $80.67\% \pm 3.84\%$ and a mean AUC score of $90.18\% \pm 2.49\%$ at the

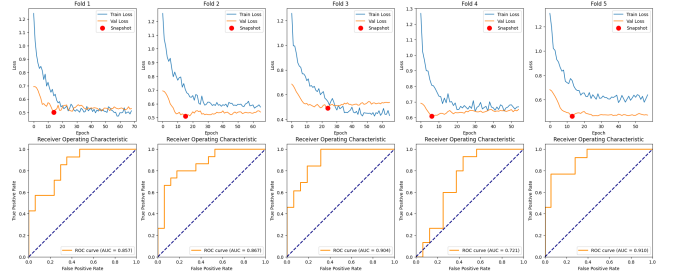


Figure 14. Unfreezing of layer4 in ResNet50 feature extractor is automated. Unfreezing mechanism activates as soon as plateau in validation loss is detected, given a set "patience" hyperparameter. Validation loss is minimum at the red dots, at which model snapshots are taken.

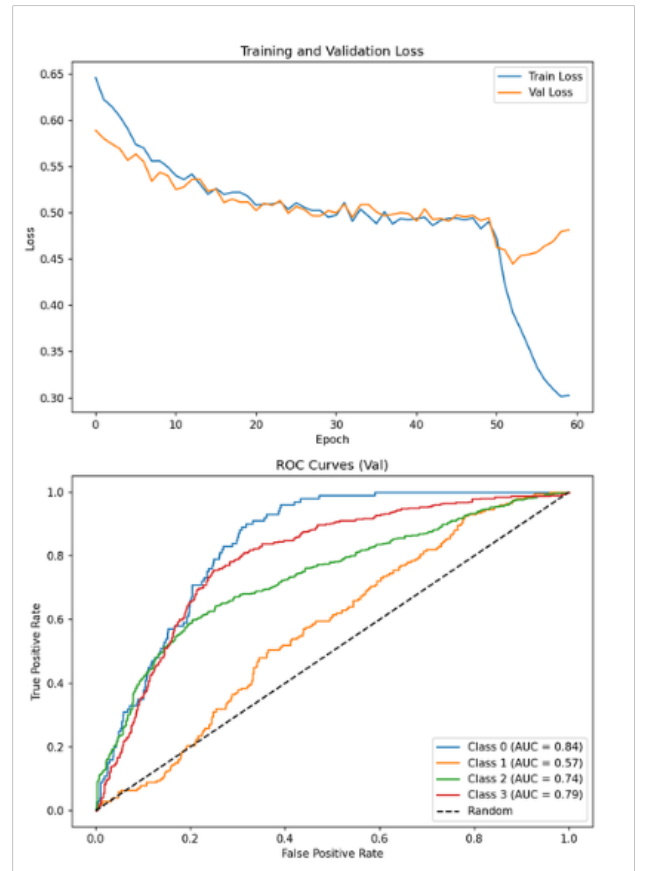


Figure 15. Loss curves taken from an example run using continuous geometric augmentation to boost data set size. Unfreezing of ResNet18 backbone happens at epoch 50, with immediate overfitting response following after. ROC AUC curves show performance of model on individual classes. This approach is later dropped in favor of the binary anomaly detection approach. See Section 3.2. Also see Fig. 14 for binary AUC curves.

FOV level, with significant performance degradation at the subimage level.

With deep learning, we showed the possibility of generating heatmaps to highlight IDC-P patterns in an FOV image. This increases model explainability and establishes appropriate trust in the

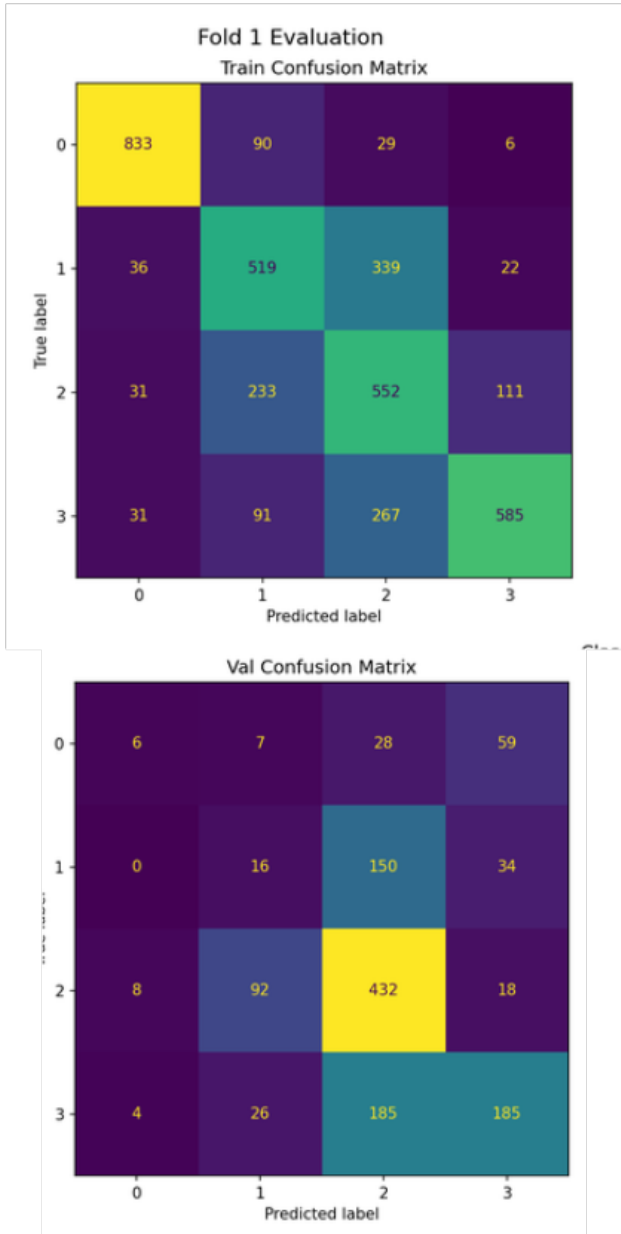


Figure 16. Train and validation confusion matrices for the same run as in Fig. 15. Having high numbers on the diagonal is good. The train confusion matrix shows significantly higher performance compared to the validation confusion matrix — this is a classic marker of severe overfitting; it likely happened when the ResNet18 backbone was unfrozen. This approach is later dropped in favor of the binary anomaly detection approach. See Section 3.2.

model’s predictions. Pathologists may verify that the model recognizes morphological IDC-P markers instead of memorizing noise.

Additionally, no manual feature engineering is required when using deep learning. This is an improvement on the original method of using hand-crafted, texture-based feature selection; with fine-tuned transfer learning, the model acts as its own feature extractor with autonomous representation learning. All pre-processing steps are standard and independent, increasing viability in field deployments where complexity and time cannot be afforded.

Possible future works include those listed in above sections,

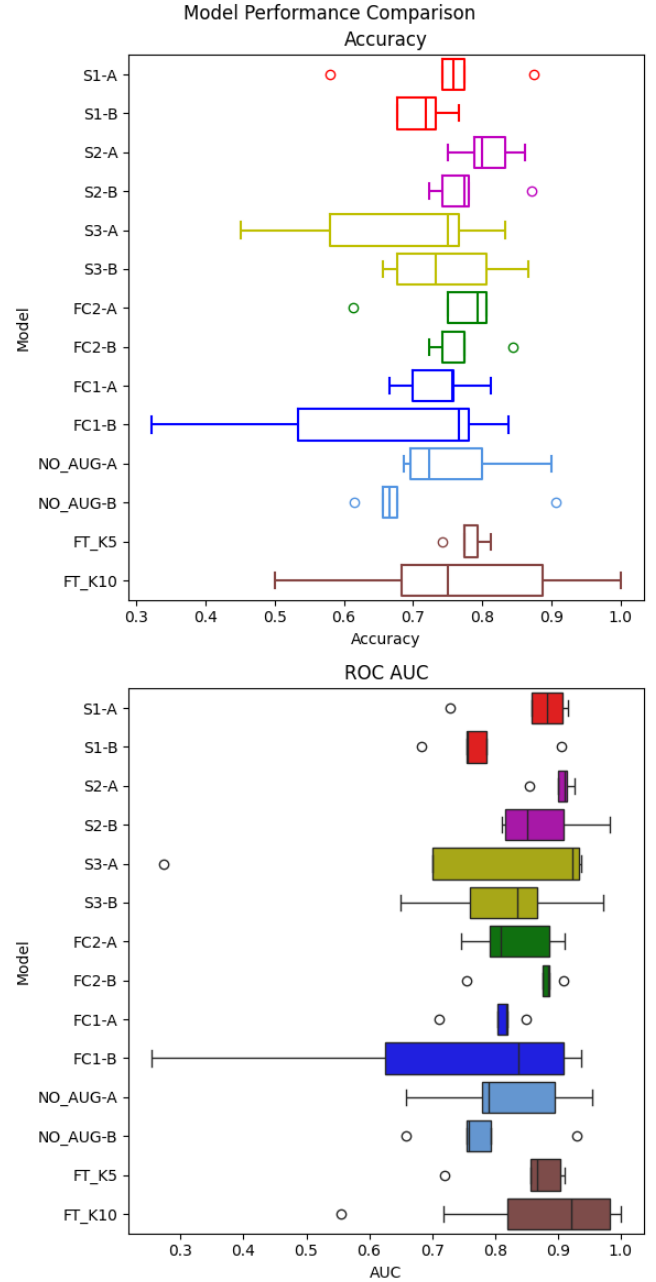


Figure 17. Performance of various hyperparameters and FC classifier architectures. Table 4 describes each hyperparameter/architectural choice in detail.

namely: BONN for a significantly more robust ordinal regression (see Section 4.11), a more specialized ViT with self-attention (see Section 4.9), and various ensemble techniques (i.e., AdaBoost, EasyEnsemble) to deal with class imbalance (see Section 4.4).

One could also experiment with combining multiple weak classifiers to form one strong master classifier. This enables the grouping of LGC and HGC data points to be discriminated against IDC, significantly increasing the trainable data set size.

As mentioned in Section 4.13, a trained pathologist could inspect and interpret Grad-CAM visualizations highlighting unique IDC patterns; this confirms whether or not the model was successful in learning identifiable IDC biomarkers. A future work could

Table 4. Complementary details for box plots in Fig. 17. "A" and "B" are two trials using the same hyperparameters. Trials are repeated to reduce variance.

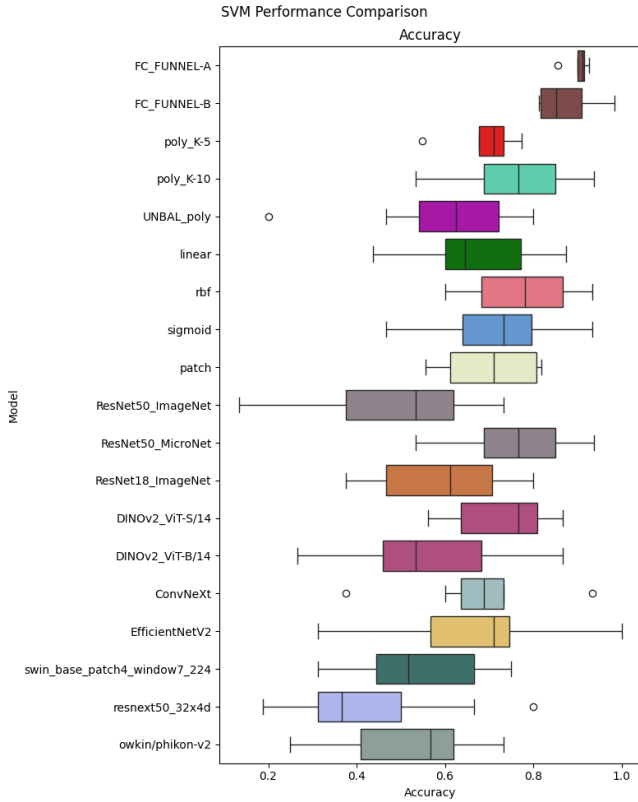
Label	Description
S1-A S1-B	Lossless augmentation (rotations), balanced via WeightedRandomSampler, Funnel arch: 512-128-32-1, sampler factor x1.
S2-A S2-B	Sampler factor x2
S3-A S3-B	Sampler factor x3
FC2-A FC2-B	ResNet50, classification head replaced. Fully connected, 2 layers deep.
FC1-A FC1-B	Fully connected, 1 layer deep
NO_AUG-A NO_AUG-B	No geometric augmentation or balancing
FT_K5 FT_K10	Fine-tuning of layer4 after plateau (MicroNet + ResNet50), K=5 and K=10 in K-Fold Cross Validation.

Table 5. Complementary details for box plots in Fig. 18. These are Accuracy data points, but it might have been more informative if these were Recall plots. N.B. the "patch" box seems unassuming and similar to other boxes. However, its recall score is an abysmal 0.20 — accuracy alone is scarcely sufficient as a performance metric. Default backbone used is ResNet50/MicroNet unless specified.

Part	Description
FC	Fully connected classifier (NN, not SVM)
FUNNEL	Funnel architecture: 512-128-32-1
A/B	Multiple runs with the same hyperparameters
K5/10	5 or 10 fold cross validation (default 10)
UNBAL	omitted class_weight="balanced"
poly/linear/rbf/sigmoid	SVM kernel used (default poly)
patch	3 x 3 sub-images with bad SHG removed

Table 6. FC Model Performance - Accuracy. See code repository for truncated outputs.

Model	Fold 1	Fold 2	Fold 3	Fold 4	Truncated
S1-A	0.759	0.774	0.875	0.581	...
S1-B	0.677	0.767	0.733	0.677	...
S2-A	0.862	0.800	0.833	0.750	...
S2-B	0.871	0.781	0.724	0.774	...
S3-A	0.767	0.750	0.833	0.581	...
S3-B	0.867	0.677	0.733	0.806	...
FC2-A	0.806	0.806	0.750	0.613	...
FC2-B	0.844	0.774	0.724	0.774	...
FC1-A	0.667	0.759	0.700	0.812	...
FC1-B	0.781	0.533	0.767	0.839	...
NO_AUG-A	0.697	0.724	0.688	0.900	...
NO_AUG-B	0.906	0.615	0.676	0.656	...
FT_K5	0.774	0.812	0.793	0.742	...
FT_K10	0.733	1.000	1.000	0.750	...

**Figure 18.** Hyperparameter tuning using a PCA-SVM classifier on feature vectors produced by various backbone feature extractors. FC_FUNNEL-A and FC_FUNNEL-B are taken from S2-A and S2-B in Fig. 17 for comparison against the FC classifier.

create generative models to generate synthetic IDC patterns to further bolster training set size, and to improve model explainability. Lastly, as the LAMPE researchers collect more microscopy images, the performance of our model can be improved upon re-training.

Table 7. FC Model Performance - ROC AUC. See code repository for truncated outputs.

Model	Fold 1	Fold 2	Fold 3	Fold 4	Truncated
S1-A	0.907	0.882	0.917	0.729	...
S1-B	0.683	0.906	0.756	0.786	...
S2-A	0.912	0.915	0.900	0.855	...
S2-B	0.983	0.910	0.851	0.812	...
S3-A	0.933	0.937	0.923	0.700	...
S3-B	0.973	0.650	0.760	0.868	...
FC2-A	0.887	0.792	0.810	0.746	...
FC2-B	0.910	0.887	0.885	0.754	...
FC1-A	0.804	0.819	0.710	0.820	...
FC1-B	0.838	0.625	0.910	0.938	...
NO_AUG-A	0.790	0.659	0.780	0.955	...
NO_AUG-B	0.931	0.758	0.793	0.754	...
FT_K5	0.857	0.867	0.904	0.721	...
FT_K10	0.982	1.000	1.000	0.812	...

ACKNOWLEDGEMENTS

We thank LAMPE Lab researchers for their continuous and ongoing support of this project: Dr. Sangeeta Murugkar and her PhD students Justin Gagnon and Parminder Riarh, for providing the labeled and pre-processed data set, supplementary information, guidance, and domain knowledge to this project. We also thank the instructor and teaching assistants of COMP 4107 for their feedback and various deep learning techniques taught during class.

Table 8. SVM Model Performance - Accuracy. See code repository for truncated outputs.

Model	Fold 1	Fold 2	Fold 3	Fold 4	Truncated
FC_FUNNEL-A	0.912	0.915	0.900	0.855	...
FC_FUNNEL-B	0.983	0.910	0.851	0.812	...
poly_K-5	0.710	0.733	0.774	0.677	...
poly_K-10	0.938	0.800	0.933	0.867	...
UNBAL_poly	0.625	0.800	0.533	0.200	...
linear	0.875	0.800	0.600	0.800	...
rbf	0.875	0.933	0.733	0.867	...
sigmoid	0.812	0.733	0.733	0.933	...
patch	0.818	0.636	0.585	0.792	...
ResNet50_ImageNet	0.625	0.533	0.600	0.133	...
ResNet50_MicroNet	0.938	0.800	0.933	0.867	...
ResNet18_ImageNet	0.625	0.800	0.600	0.800	...
DINOv2_ViT-S/14	0.812	0.867	0.800	0.600	...
DINOv2_ViT-B/14 (1)	0.750	0.533	0.867	0.467	...
DINOv2_ViT-B/14 (2)	0.750	0.467	0.533	0.267	...
ConvNeXt	0.688	0.600	0.933	0.733	...
EfficientNetV2	0.688	0.733	1.000	0.800	...
swin_base_patch4	0.312	0.667	0.333	0.733	...
resnext50_32x4d	0.312	0.667	0.533	0.800	...
owkin/phikon-v2	0.625	0.733	0.600	0.400	...
robust_minmax	0.562	0.800	0.467	0.467	...

DATA AVAILABILITY

The data set belongs to and is provided by the LAMPE Lab at Carleton University.

AUTHOR CONTRIBUTIONS

JY managed communications and data transfer with the LAMPE research lab. JY conceived the idea of using ordinal regression and binary anomaly detection methods, AN implemented and obtained empirical results. AN suggested using Grad-CAM for explainability, JY and VG worked on Grad-CAM visualizations. JY experimented with AB-MIL and attention-based techniques. JY liaised with LAMPE for domain knowledge to write the MatReader data set parser. JY selected and executed the Mann-Whitney U test for a statistical-based hyperparameter tuning process. VG conducted the backbone-model experimentation, including testing multiple CNN architectures such as ResNet18, ResNet50, VGG-16, and EfficientNetB0. VG performed extensive hyperparameter exploration, evaluated different learning-rate schedules and dropout configurations for ResNet-based models. AN implemented the patch-based autoencoder pipeline including preprocessing, patch extraction, and reconstruction based training. AN implemented FOV-level evaluation framework with mean-logit aggregation, along with the integrated autoencoder encoder into downstream classification pipeline. AN ran experiments to evaluate frozen fine-tuned encoder representations against supervised CNN baseline and implemented several improvements for training stability/performance (training for longer, checkpointing models, early stopping). All authors contributed to the interpretation of the results. All authors took part in the revision process and approved the final copy of this report.

REFERENCES

- Allena A. S., 2022, 10, 2320
 Cao W., Mirjalili V., Raschka S., 2020, *Pattern Recognition Letters*, 140, 325–331

- Darcet T., Oquab M., Mairal J., Bojanowski P., 2023, *Vision Transformers Need Registers*
 Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L., 2009, 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp 248–255
 Filiot A., Jacob P., Kain A. M., Saillard C., 2024, Phikon-v2, A large and public feature extractor for biomarker prediction ([arXiv:2409.09173](https://arxiv.org/abs/2409.09173)), <https://arxiv.org/abs/2409.09173>
 Gagnon J. R., Allen C. H., Diop M.-K., Dallaire F., Leblond F., Trudel D., Murugkar S., 2025, *Sci. Rep.*, 15, 39874
 Gildenblat J., contributors 2021, PyTorch library for CAM methods, <https://github.com/jacobgil/pytorch-grad-cam>
 Guo C. C., Epstein J. I., 2006, *Modern Pathology*, 19, 1528
 He K., Zhang X., Ren S., Sun J., 2015
 Huttunen M. J., Hassan A., McCloskey C. W., Fasih S., Upham J., Vnderhyden B. C., Boyd R. W., Murugkar S., 2018, *J. Biomed. Opt.*, 23, 1
 Ilse M., Tomczak J. M., Welling M., 2018
 Jiang P., Liu J., Luo Q., Pang B., Xiao D., Cao D., 2023, *J. Digit. Imaging*, 36, 1110
 Jose C., et al., 2024, DINOv2 Meets Text: A Unified Framework for Image- and Pixel-Level Vision-Language Alignment
 Lázaro M., Figueiras-Vidal A. R., 2023a, *Pattern Recognition*, 137, 109303
 Lázaro M., Figueiras-Vidal A. R., 2023b, *Pattern Recognition*, 137, 109303
 Oquab M., et al., 2023, DINOv2: Learning Robust Visual Features without Supervision
 Schapire R. E., 2013, *Explaining AdaBoost*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 37–52, [doi:10.1007/978-3-642-41136-6_5](https://doi.org/10.1007/978-3-642-41136-6_5), https://doi.org/10.1007/978-3-642-41136-6_5
 Selvaraju R. R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D., 2020, *Int. J. Comput. Vis.*, 128, 336
 Shao Z., Bian H., Chen Y., Wang Y., Zhang J., Ji X., Zhang Y., 2021
 Shi X., Cao W., Raschka S., 2023, *Pattern Analysis and Applications*, 26, 941–955
 Stuckner J., Harder B., Smith T. M., 2022, *Npj Comput. Mater.*, 8

APPENDIX A: STATISTICAL TESTS FOR MODEL PERFORMANCE

The hyperparameter tuning and architectural design choices in this investigation were motivated by non-parametric statistical tests. The Mann-Whitney U Test (see eq. A2) is used when comparing exactly two techniques, while the Kruskal-Wallis test is used when comparing more than two techniques. AUC or accuracy scores are taken from around 5-10 cross validation folds from each trial. Hence, there is not enough data to prove a normal distribution, and so a non-parametric test is suitable. The standard P-value thresholds of $P > 0.05$ for "Not Significant", $P < 0.05$ for "Significant", $P < 0.01$ for "Strongly Significant", and $P < 0.001$ for "Highly Significant" are used to determine statistical significance in both tests, except in scenarios where multiple Mann-Whitney used to compare more than two classes. In that case, the Bonferroni Correction is applied to reduce the risk of Type I errors (false positives) when performing multiple, simultaneous hypothesis tests.

$$U_1 = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1 \quad (\text{A1})$$

$$U_2 = n_1 n_2 + \frac{n_2(n_2 + 1)}{2} - R_2 \quad (\text{A2})$$

Classification performance metrics such as precision, recall (or true positive rate), F1 score, and accuracy are measured for every fold taken during training and evaluation. However, the ROC AUC values have instead been chosen for statistical tests due to its ability to evaluate model performance across all possible cutoff thresholds.

The LAMPE data set provides only four unique biopsy cores for the minority "Healthy" class (see Section 2). This severely limits `StratifiedGroupKFold` to be done over four folds to prevent intra-core bias and data leakage. The number of folds is artificially increased using `RepeatedStratifiedKFold` in conjunction with `StratifiedGroupKFold` and a random seed to reduce variance when performing statistical tests.

This paper has been typeset from a $\text{\LaTeX}$ file prepared by the author.