

# Exploring ResNet18 for Multi-Class Classification and Cross-Domain Transfer Learning

Nicholas Kamra  
Concordia University  
Gina Cody School of Engineering and Computer Science  
yckamra@gmail.com

## Abstract

*This project explores deep learning for image classification using the ResNet18 Convolutional Neural Network (CNN) architecture to address challenges in computational pathology and computer vision. Task 1 trained ResNet18 on colorectal cancer data, achieving 97% test accuracy with feature representations visualized via t-SNE. Task 2 employed transfer learning, using features extracted by the Task 1 ResNet18 encoder and an ImageNet-1k pre-trained encoder. These features were classified using non-linear Support Vector Machines (SVMs), achieving accuracies of 92% and 66% for the ResNet18 encoder, and 98% and 100% for the ImageNet-1k pre-trained encoder on prostate cancer and animal faces datasets, respectively. The results demonstrate the superior generalization of the ImageNet pre-trained model while highlighting the power of CNNs and transfer learning to extract and classify meaningful features across diverse domains. This study underscores the potential of feature extraction and visualization for multi-domain image classification.*

## 1. Introduction

### 1.1. Problem Statement and Importance

Image classification is a fundamental task in computer vision, with applications ranging from healthcare diagnostics to wildlife monitoring. In computational pathology, accurate classification of tissue samples is critical for diagnosing diseases such as colorectal and prostate cancer, directly impacting treatment outcomes and patient care. Similarly, in computer vision, tasks like animal identification are vital for ecological studies and conservation efforts. Despite their importance, these tasks face challenges such as domain-specific variations, limited labeled data, and the need for adaptable models capable of delivering high accuracy across different application fields. This project seeks to address these issues by leveraging deep learning models,

particularly ResNet18, and exploring the potential of transfer learning to generalize across domains effectively.

### 1.2. Challenges and Motivation

Limited data is a common issue in machine learning, even more so in image recognition as access to quality images regarding the classification task at hand can be limited. During training, without enough data, this can lead to the model under fitting and unable to detect the underlying patterns within the images themselves. This is also an issue when training a model with the goal of using it for transfer learning; A small dataset of 6000 images of colorectal cancer tissue samples may be biased and only capture patterns from particular populations of patients causing less generalization or may be too small to truly capture both high and low-level features in the dataset, which does not transfer over well to new datasets - luckily both a non-issue with the datasets used in this study as they were randomly selected from datasets of 100k images and as long as the trained CNN performed well on the training and test sets, the model obtained all the high and low-level features it could have, being limited only by model size and variation of the datasets.

Variation also comes into play when using transfer learning on new datasets. There exists fur patterns, eye shapes, stop signs, or numbers depending on the image dataset we are trying to apply image recognition to. With a dataset specialized to a particular category, such as colorectal cancer, these high and low-level features may not be learned, caused not only by small datasets (and small models), but by specialized datasets without enough variance to capture these patterns that transfer well to other datasets with seemingly little in common.

It is widespread knowledge that some data is just hard to get or is costly and expensive, yet the value to computational pathology being able to create machine models in certain fields such as cancer detection in medicine to animal identification in biology, is priceless. Exploring transfer learning allows us to have our cake and eat it too, en-

abling pre-trained models to be leveraged to apply to other real world datasets without the need of an infeasible amount of specialized images.

### 1.3. High-Level Methodology Overview

This project employs a two-phase methodology to tackle image classification tasks across multiple datasets. The first phase involves training a Convolutional Neural Network (CNN) model, ResNet18, for feature extraction. The classification head of the ResNet18 model was removed, enabling the extracted features to be used for transfer learning. In the second phase, these features were applied to new datasets and classified using a classical machine learning algorithm, with performance compared against a pre-trained ResNet18 model on the ImageNet-1k dataset.

ResNet18 was selected for its efficient feature extraction capabilities, and the PyTorch framework was utilized for model training and feature analysis. Dimensionality reduction of extracted features was performed using t-SNE, achieved with sklearn library, allowing for effective visualization of feature representations. Visualization tasks, including t-SNE plots and learning curves (costs and accuracies), were accomplished using the matplotlib library. Additionally, the sklearn library was employed for calculating precision, recall, and F1 scores for both the CNN-based feature extractors and the classical machine learning classifiers, as well as creating SVM models.

For classification, a non-linear Support Vector Machine (SVM) model was used to replace the heads of both the ResNet18 encoder trained on the project datasets and the pre-trained ImageNet-1k ResNet18 model. Non-linear SVM was chosen for its ability to handle complex, non-linear data relationships effectively. While dimensionality reduction with t-SNE ensured the data was no longer high-dimensional, non-linear SVMs perform particularly well on small datasets and provide robust classification compared to models like Random Forests in many cases.

### 1.4. Results Overview

The ResNet18 model trained on the colorectal cancer dataset demonstrated strong classification performance across all three classes: smooth muscle, normal colon mucosa, and cancer-associated stroma. On the test set, the model achieved an overall accuracy of 97%, with a macro-average F1-score of 0.97. Class-wise performance was consistent, with F1-scores of 0.96, 0.99, and 0.96 for the three classes, respectively. These results highlight the effectiveness of ResNet18 as a feature extractor and classifier for domain-specific pathology datasets.

For the prostate cancer dataset (Dataset 2), the Support Vector Machine (SVM) classifier achieved an accuracy of 92% when features were extracted using the ResNet18 encoder trained in Task 1. In comparison, features ex-

tracted using the pre-trained ResNet18 model on ImageNet achieved a significantly higher accuracy of 98%, demonstrating the advantage of diverse pre-training for pathology tasks.

For the animal faces dataset (Dataset 3), the Task 1 ResNet18 encoder achieved a test accuracy of 66%, reflecting the challenge of generalizing from a domain-specific model to an unrelated domain. However, the pre-trained ImageNet ResNet18 model achieved a perfect test accuracy of 100% on the same dataset, underscoring the value of pre-trained models with greater feature diversity for highly varied datasets.

### 1.5. Related Works

Understanding the scientific research surrounding our models and datasets is essential for contextualizing this project within the broader field and identifying relevant methodologies and insights. Prior studies have explored various applications of convolutional neural networks (CNNs) that inform our work. For example, Kwiatkowska et al. [3] investigated the use of CNNs for detecting malignant melanoma in dermoscopy images, demonstrating the potential of deep learning in medical imaging, particularly for cancer classification tasks. Similarly, research by Gu and Lee [1] examined pre-trained CNN models on ImageNet, revealing how these models can generalize to new tasks by extracting shared features, such as those used to identify pneumonia from X-rays. This work underscores the significance of pre-trained models in leveraging diverse feature representations for domain-specific problems.

Additionally, the use of CNNs in ecological studies has shown promising results in wildlife classification. Islam et al. [2] demonstrated that pre-trained CNNs could accurately classify animal species from camera trap images, showcasing their utility in identifying patterns across diverse datasets. These studies provide valuable context for understanding the challenges and opportunities in applying CNNs and transfer learning to domain-specific problems. They also highlight the broader relevance of leveraging pre-trained models to address data scarcity and enhance classification accuracy in both medical and ecological applications.

## 2. Methodology

### 2.1. Proposed Solution

This project aims to evaluate the effectiveness of a transfer learning pipeline utilizing a ResNet18 architecture, illustrated in Figure 1, with two distinct initialization strategies: one trained on a domain-specific dataset and another pre-trained on ImageNet. The methodology is centered around addressing the limitations posed by small, domain-specific datasets through feature extraction, dimensionality

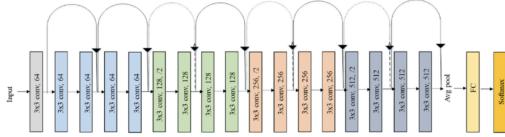


Figure 1. The original ResNet18 architecture before the fully-connected layer and softmax layers (head) have been removed.

reduction, and classical machine learning classification.

The primary strategy involves training a ResNet18 model initialized with random weights on Dataset 1, a colorectal cancer classification dataset. After training, the fully connected layer (head) of the model is removed, converting the ResNet18 into a feature extractor. This trained model is then used to extract feature embeddings from Dataset 2 (prostate cancer classification) and Dataset 3 (animal faces classification). For comparison, a ResNet18 model pre-trained on ImageNet is similarly employed as a feature extractor for the same datasets.

The extracted feature embeddings are subjected to dimensionality reduction using t-SNE, providing interpretable visualizations of the feature distributions across the datasets. This visualization step allows for an assessment of how well-separated the classes are in the feature space, offering insights into the generalization capacity of the two ResNet18 models.

To perform classification, the reduced features are fed into a non-linear Support Vector Machine (SVM) classifier. This approach is applied to four scenarios: the trained ResNet18 model on Datasets 2 and 3, and the ImageNet pre-trained ResNet18 model on Datasets 2 and 3. The use of a non-linear SVM ensures robust handling of non-linear decision boundaries, particularly important for datasets with complex patterns and limited size.

This pipeline integrates the strengths of domain-specific training, transfer learning, and classical machine learning to address the challenges of limited labeled data, domain adaptability, and generalization; As seen in Figure 2, both high and low-level features can be extracted and generalized across vastly different images, detecting simple shapes such as triangles to more complex textures. The comparative evaluation between the trained ResNet18 model and the ImageNet pre-trained model offers valuable insights into the relative merits of each approach for feature extraction and classification in real-world applications.

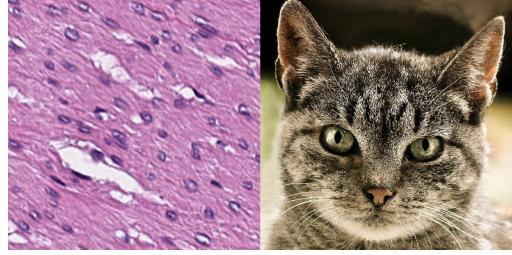


Figure 2. An image of a smooth muscle (MUS) tissue sample from Dataset 1 alongside an image of a cat in Dataset 3, showing the stark differences between cancerous tissues and animal faces; Visually there is not a lot in common to the naked eye but a Convolutional Neural Network (CNN) can detect both high and low-level features across diverse domains.

## 2.2. Implementation Details

### 2.2.1 Dataset Preprocessing

The datasets used in this study span three distinct domains: colorectal cancer classification, prostate cancer classification, and animal face classification. Each dataset was reduced to a manageable size of 6,000 images and organized into three classes to ensure computational efficiency while maintaining sufficient diversity for training and evaluation. These datasets were stored in centralized repositories on Google Drive, facilitating collaborative access and integration into the preprocessing pipeline. Datasets were then loaded into the Google Colab local storage in order to limit the need for repeated database requests during training, as the data is accessed locally after the initial download, significantly reducing transfer times and improving efficiency. Images were provided in .tif format for Dataset 1 and .jpg format for Datasets 2 and 3.

Dataset 1, focused on colorectal cancer classification, was derived from a larger dataset containing 100,000 image patches across eight tissue types. For this project, the dataset was reduced to three classes: smooth muscle (MUS), normal colon mucosa (NORM), and cancer-associated stroma (STR). Dataset 2, aimed at prostate cancer classification, was reduced from 120,000 image patches to three classes: tumor tissue, benign glandular tissue, and benign non-glandular tissue. Dataset 3, targeting animal face classification, originated from a collection of 16,000 images and was similarly reduced to three classes: cats, dogs, and wildlife animals.

The preprocessing pipeline for all three datasets involved organizing images into directories corresponding to their respective classes, retrieving their file paths, and assigning numeric labels to each class for consistency in downstream processing. The datasets were then shuffled to ensure randomness and split into training, cross-validation, and test sets, with 80% allocated to training and the remaining 20%

split evenly between cross-validation and testing. These splits were generated using a stratified approach to maintain class balance across all subsets.

To prepare the images for model input, a standardized set of transformations was applied using the PyTorch `torchvision.transforms` library. Images were resized to 256 pixels on the shorter side, center-cropped to 224×224 pixels to match the input dimensions required by ResNet18, and converted to PyTorch tensors. Finally, pixel values were normalized using the mean and standard deviation of the ImageNet dataset to ensure compatibility with pre-trained models and enhance generalization.

This preprocessing pipeline ensured that all datasets were formatted uniformly, enabling seamless integration into the feature extraction and classification pipelines. The transformations and splits were designed to optimize model training while ensuring robust evaluation on unseen data.

### 2.2.2 ResNet18 Model Training

The ResNet18 architecture, loaded from PyTorch, was trained on the colorectal cancer dataset for feature extraction and classification. The model was initialized with randomly assigned weights, and training was conducted over 10 epochs. This epoch count was selected to provide the model with sufficient opportunities to traverse the cost landscape, allowing it to escape local minima through the momentum-driven optimization of the Adam optimizer. Additionally, this duration ensured computational efficiency by avoiding unnecessary prolongation of training.

The Adam optimizer was chosen for its adaptive learning rate and momentum properties, which enhance convergence stability. A learning rate of  $\eta = 0.0001$  was employed to prevent the optimizer from overshooting the optimal solution, a phenomenon observed with higher learning rates in preliminary experiments. Weight decay ( $\lambda = 1 \times 10^{-4}$ ) was incorporated as a regularization mechanism to mitigate overfitting by penalizing large weights. Gradient clipping was also applied to constrain gradient magnitudes and prevent the occurrence of exploding gradients, thereby stabilizing the optimization process.

A learning rate scheduler was implemented to adjust the learning rate dynamically based on the model’s performance during training. Specifically, the `ReduceLROnPlateau` scheduler monitored the cross-validation loss and reduced the learning rate by a factor of 0.1 when the loss plateaued, promoting finer adjustments as training progressed. Cross-entropy loss was employed as the objective function to minimize the discrepancy between predicted and true class distributions.

Image data were preprocessed and loaded using PyTorch’s `DataLoader`, which facilitated parallelized pre-

processing on CPU cores and efficient transfer of image batches to GPU memory for training. This ensured a seamless data pipeline, minimizing bottlenecks in data handling.

Throughout training, both training and cross-validation losses and accuracies were recorded at the end of each epoch. This allowed for real-time monitoring of model performance, enabling the identification of potential overfitting or underfitting trends and ensuring the model’s capacity to generalize effectively to unseen data. Graphing of loss and dataset accuracies, as well as classification reports were conducted for fine-tuning; Graphing was accomplished with `matplotlib` and the classification reports were calculated by `scikit-learn`.

### 2.2.3 Feature Extraction and Dimensionality Reduction

The two ResNet18-based architectures were employed for this task: the custom-trained model, fine-tuned on the colorectal cancer dataset, and a pre-trained model initialized with ImageNet-1K weights. Both models had their fully connected classification heads removed, leaving only the convolutional layers to function as feature extractors. This adjustment enabled the networks to map input images to 512-dimensional feature vectors, representing abstract features of the images.

The extracted feature vectors were then subjected to dimensionality reduction for visualization. t-distributed Stochastic Neighbor Embedding (t-SNE) was used and performed by `scikit-learn` to reduce the 512-dimensional feature space to two dimensions, enabling qualitative analysis of the feature representations. Parameters for t-SNE included a perplexity of 30, a learning rate of 200, and a maximum of 1000 iterations, chosen to optimize clustering performance and maintain interpretability. The visualized embeddings were color-coded according to class labels by `matplotlib`, allowing for an intuitive assessment of the separability and clustering tendencies within the feature space.

The extracted features, accompanied by their corresponding class labels, were stored as NumPy arrays for downstream processing. This representation served as the foundation for subsequent tasks, including the classification of features using classical machine learning algorithms such as Support Vector Machines (SVMs).

### 2.2.4 Classification

The classification step evaluated the utility of extracted features from ResNet18 encoders for downstream tasks using Support Vector Machines (SVMs). Two CNN encoders were utilized: (1) the Task 1 ResNet18 model, fine-tuned on colorectal cancer data, and (2) the ImageNet-pretrained ResNet18 model; Classification was done on the features

extracted from both encoders on Datasets 2 (prostate cancer classification) and 3 (animal face classification).

SVM classifiers were implemented using the scikit-learn library. The radial basis function (RBF) kernel was selected to accommodate potential non-linearities in the feature space. A grid search was conducted for hyperparameter tuning, exploring the regularization parameter ( $C$ ) and kernel coefficient ( $\gamma$ ), with values  $C \in \{0.1, 1, 10\}$  and  $\gamma \in \{\text{scale}, 0.1, 0.01\}$ . Three-fold cross-validation was used to identify the optimal hyperparameters, ensuring robust performance during training. The best-performing model from the grid search was then evaluated on the independent test set.

The classification workflow was performed across four scenarios: (1) features from the Task 1 ResNet18 model applied to Dataset 2, (2) features from the ImageNet-pretrained ResNet18 model applied to Dataset 2, (3) features from the Task 1 ResNet18 model applied to Dataset 3, and (4) features from the ImageNet-pretrained ResNet18 model applied to Dataset 3. The datasets were divided into training, cross-validation, and test splits, ensuring consistency with earlier preprocessing steps.

For each scenario, the extracted features were used as input to the SVM classifiers, and the performance was assessed using precision, recall, F1-score, and overall accuracy. These metrics were computed using the `classification_report` function in scikit-learn. The scikit-learn implementation also provided tools for parameter optimization, classifier training, and performance validation, ensuring reproducibility.

## 2.2.5 Libaries and Tools

The implementation of this project relied on a variety of libraries and tools for preprocessing, model training, feature extraction, dimensionality reduction, and classification. The deep learning framework PyTorch was the cornerstone of this project, providing tools for constructing and training the ResNet18 convolutional neural network (CNN) model. Torchvision, an extension of PyTorch, was used for dataset-specific preprocessing, including resizing, cropping, normalization, and data augmentation. These preprocessing steps were tailored to align the dataset format with the input requirements of ResNet18.

For loading and batching data during model training and evaluation, the `torch.utils.data` module, including the `DataLoader` class, was employed. The `DataLoader` facilitated efficient parallelized loading of images onto the GPU, optimizing the training pipeline. The images were stored on Google Drive and accessed through `google.colab`, allowing seamless integration into the Google Colab environment for computation on cloud GPUs.

Dimensionality reduction was performed using scikit-

learn's `TSNE` class, which enabled visualizing high-dimensional feature embeddings in two dimensions. This library was chosen for its implementation efficiency and flexibility in hyperparameter tuning, such as perplexity, learning rate, and maximum iterations. For the classification task, scikit-learn's `SVC` (Support Vector Classifier) was employed, along with `GridSearchCV` for hyperparameter optimization. These tools enabled the evaluation of various SVM configurations, such as the regularization parameter  $C$  and kernel coefficient  $\gamma$ , using cross-validation.

Visualization of model performance and data exploration was carried out using `matplotlib.pyplot`. This library allowed plotting of training and validation accuracy, loss trends across epochs, and t-SNE visualizations. Additionally, NumPy provided matrix operations and numerical computations essential for data manipulation throughout the project.

The development environment was Google Colab, leveraging its integration with Google Drive for dataset management and its GPU support for computational efficiency. This environment facilitated seamless experimentation with ResNet18 architectures, enabling rapid prototyping and evaluation. The combined use of these libraries and tools provided a robust and scalable framework for conducting deep learning experiments in image classification and transfer learning.

## 2.3. Improvements and Innovations

This project introduced several advancements to optimize model training, improve generalization, and enhance interpretability across diverse datasets. A key improvement was the fine-tuning of hyperparameters, including the learning rate, weight decay, and gradient clipping, to ensure stable convergence during training. The dynamic adjustment of the learning rate using a scheduler allowed the model to efficiently escape local minima, leading to improved overall performance.

The integration of transfer learning played a pivotal role in extending the applicability of the ResNet18 architecture. By leveraging feature extraction from both a custom-trained CNN and a pre-trained ImageNet model, the project demonstrated the versatility of CNN-based encoders in classifying datasets from distinct domains, such as pathology images and animal faces. This approach showcased a hybrid method where features extracted by deep learning were classified using a Support Vector Machine (SVM), combining the strengths of both paradigms.

Dimensionality reduction using t-SNE provided a novel mechanism to analyze and visualize high-dimensional feature embeddings. By tailoring t-SNE parameters, such as perplexity and learning rate, to suit each dataset, the visualization highlighted distinct clustering patterns that informed model refinement. This technique improved interpretability,

offering deeper insights into the learned feature representations.

The project further contributed to efficient resource utilization by implementing parallelized data loading through PyTorch’s `DataLoader` and leveraging cloud GPUs for computationally intensive tasks. These strategies reduced training time and enabled processing of large datasets with minimal overhead.

By addressing the challenge of generalizing across domains, this work highlighted the adaptability of deep learning models trained on highly specialized datasets. The application of classical machine learning techniques, such as SVM with hyperparameter optimization, demonstrated the effectiveness of hybrid methods in achieving robust classification performance. These innovations collectively advanced the methodology for image classification tasks, emphasizing scalability, adaptability, and interpretability in machine learning workflows.

## 3. Results

### 3.1. Experiment Setup

The experiments were conducted in the Google Colab environment, which provides GPU acceleration for computational efficiency. The datasets were stored on Google Drive and accessed via the `google.colab` module, ensuring seamless integration and efficient data retrieval. The experimental pipeline was implemented using Python, leveraging PyTorch for model training and feature extraction, scikit-learn for dimensionality reduction and classification, and matplotlib for visualizations.

A single NVIDIA Tesla T4 GPU was used for training the ResNet18 architecture and for feature extraction. To ensure consistency across experiments, the random seed was fixed at 42 across all frameworks, including PyTorch, NumPy, and scikit-learn. The datasets were preprocessed to standardize input dimensions, normalize pixel values, and split into training, cross-validation, and testing subsets, following the methodology outlined in Section 2.

Training of the ResNet18 model was performed using the Adam optimizer with a learning rate of  $\eta = 0.0001$  and a weight decay of  $\lambda = 1 \times 10^{-4}$ . The learning rate scheduler reduced the learning rate by a factor of 0.1 if the cross-validation loss plateaued for more than two epochs. The model was trained over 10 epochs, with gradient clipping applied to stabilize optimization. Feature extraction involved removing the fully connected layer from the trained ResNet18 model and extracting 512-dimensional embeddings for all datasets.

Dimensionality reduction was carried out using t-SNE with parameters tuned for optimal visualization: perplexity of 30, a learning rate of 200, and a maximum of 1,000 iterations. These embeddings were visualized in two di-

mensions to assess separability among classes. The reduced features were subsequently classified using Support Vector Machines (SVMs) with an RBF kernel, optimized through grid search over hyperparameters  $C$  and  $\gamma$ . Three-fold cross-validation ensured robust hyperparameter tuning, and final evaluations were performed on the independent test sets.

All metrics, including precision, recall, F1-score, and accuracy, were computed using scikit-learn’s evaluation functions. This setup ensured reproducibility, consistent comparisons between models, and reliable performance evaluations across all datasets and scenarios.

### 3.2. Main Results

#### 3.2.1 Training and Evaluation on ResNet18 Model

The first phase of the pipeline involved training the ResNet18 model, initialized with random weights, on Dataset 1, which focuses on colorectal cancer classification. This step was critical for creating a domain-specific encoder capable of extracting relevant features for downstream tasks on Datasets 2 and 3.

The model was trained over 10 epochs, using the Adam optimizer with a learning rate of  $\eta = 0.0001$  and weight decay of  $\lambda = 1 \times 10^{-4}$ . Cross-entropy loss was used as the objective function. Training and cross-validation datasets were derived using an 80-20 stratified split, ensuring class balance across splits.

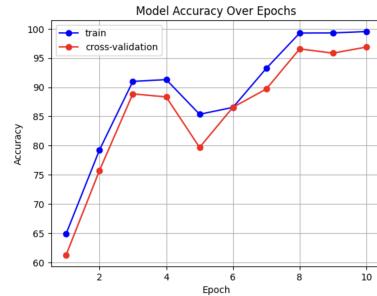


Figure 3. Training (blue) and cross-validation (red) set accuracies over 10 epochs during training of ResNet18 model on colorectal cancer dataset.

Figure 3 and Figure 4 show the training and cross-validation accuracy and loss curves over the course of training, respectively. The loss curves indicate a relatively smooth convergence for both training and cross-validation datasets, with minimal overfitting observed. The accuracy curves show consistent improvement, demonstrating the model’s capacity to learn domain-specific features effectively.

At the final epoch of training, the model achieved a training loss of 0.0217 and a cross-validation loss of 0.0845. The corresponding accuracies were 99.53% for the training

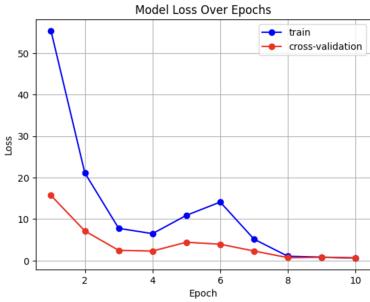


Figure 4. Training (blue) and cross-validation (red) set losses over 10 epochs during training of ResNet18 model on colorectal cancer dataset.

set and 96.88% for the cross-validation set. These results highlight the model’s robust learning capability and its generalization performance, particularly in capturing domain-specific features critical for the classification of colorectal cancer tissue.

The classification performance of the ResNet18 model trained on the colorectal cancer dataset (Dataset 1) is summarized through its performance on the train, cross-validation, and test sets. Table 1 highlights the train set results, demonstrating near-perfect precision, recall, and F1-scores across all three classes: smooth muscle (class 0), normal colon mucosa (class 1), and cancer-associated stroma (class 2). The model achieved an overall training accuracy of 100%, indicating its ability to effectively learn domain-specific features with minimal errors on the training data.

The cross-validation results, summarized in Table 2, reflect the model’s strong generalization capabilities. An accuracy of 97% was achieved, with high precision, recall, and F1-scores across all classes. Notably, class 1 exhibited the highest performance, achieving an F1-score of 0.99. Classes 0 and 2 also performed well, with F1-scores of 0.96 and 0.95, respectively. These results suggest that the model effectively captured the nuances of the dataset while maintaining robust generalization.

Class	Precision	Recall	F1-Score	Support
0	0.99	0.99	0.99	1263
1	1.00	1.00	1.00	1261
2	0.99	0.99	0.99	1316
Accuracy	1.00 (3840)			
Macro Avg	1.00	1.00	1.00	3840
Weighted Avg	1.00	1.00	1.00	3840

Table 1. Classification Report for the Training Set. The results demonstrate high precision, recall, and F1-scores across all classes, with an overall accuracy of 100%.

Finally, Table 3 presents the model’s performance on the independent test set, achieving an accuracy of 97%. The

Class	Precision	Recall	F1-Score	Support
0	0.98	0.94	0.96	320
1	1.00	0.98	0.99	349
2	0.92	0.98	0.95	291
Accuracy	0.97 (960)			
Macro Avg	0.97	0.97	0.97	960
Weighted Avg	0.97	0.97	0.97	960

Table 2. Classification Report for the Cross-Validation Set. The results indicate high precision, recall, and F1-scores across all classes, with an overall accuracy of 97%.

precision, recall, and F1-scores for all classes remain consistent with the cross-validation set, reinforcing the model’s ability to generalize effectively to unseen data. Class 1 again exhibits the highest performance, with an F1-score of 0.99, while classes 0 and 2 achieve F1-scores of 0.96. The slight variation in recall, particularly for class 0, suggests that some samples in the test set may present overlapping features, but these discrepancies are minimal.

Across all datasets, the ResNet18 model demonstrated strong and consistent performance. The training accuracy of 100% and the near-equal test and cross-validation accuracies of 97% illustrate a balanced model capable of learning robust features without significant overfitting. These results underscore the effectiveness of the ResNet18 architecture, the preprocessing pipeline, and the training strategy in addressing complex classification tasks in computational pathology.

Class	Precision	Recall	F1-Score	Support
0	0.97	0.96	0.96	417
1	0.99	0.99	0.99	390
2	0.95	0.97	0.96	393
Accuracy	0.97 (1200)			
Macro Avg	0.97	0.97	0.97	1200
Weighted Avg	0.97	0.97	0.97	1200

Table 3. Classification Report for the Test Set. The results indicate consistently high precision, recall, and F1-scores across all classes, with an overall accuracy of 97%.

### 3.2.2 Feature Extraction and t-SNE Dimensionality Reduction Visualization

t-SNE visualizations in Figure 5 and Figure 6 illustrate feature extraction. The training set (Figure 5) shows clear class clusters, reflecting effective learning. The cross-validation set (Figure 6) displays less distinct clusters, indicating generalization with minimal overfitting, as class separation remains evident.

The t-SNE visualizations in Figures 7 and 8 illustrate the feature separability achieved through dimensionality re-

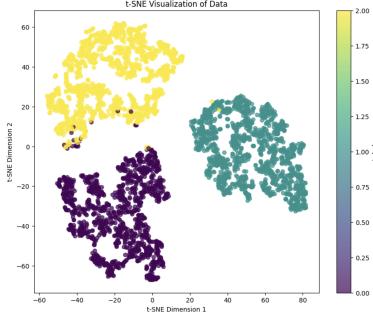


Figure 5. t-SNE dimensionality reduction visualization of three distinct classes in the colorectal cancer dataset extracted from the custom-trained ResNet18 model.

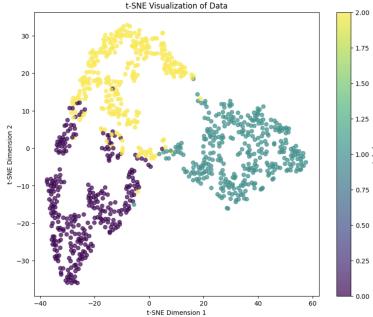


Figure 6. t-SNE dimensionality reduction visualization of three distinct classes in the colorectal cancer dataset extracted from the pre-trained ImageNet-1k ResNet18 model.

duction for Dataset 2 and Dataset 3. These visualizations compare the performance of the custom-trained ResNet18 encoder with the ImageNet-pretrained ResNet18 encoder, highlighting their capacity to generalize across domains.

For Dataset 2, focused on prostate cancer classification, the custom-trained ResNet18 encoder (Figure 7, left) shows discernible but overlapping clusters, indicating moderate class separability. In contrast, the ImageNet-pretrained encoder (Figure 7, right) produces compact, well-separated clusters, demonstrating superior generalization due to diverse pretraining.

For Dataset 3, which involves animal face classification, the difference is even more pronounced. The custom-trained encoder (Figure 8, left) exhibits significant overlap, underscoring the challenge of transferring features from pathology data to unrelated domains. Meanwhile, the ImageNet-pretrained encoder (Figure 8, right) generates distinct, well-separated clusters, reflecting its robustness across tasks.

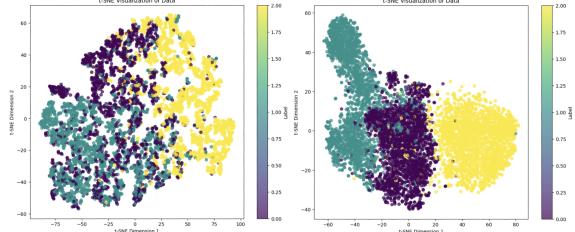


Figure 7. t-SNE dimensionality reduction visualization comparison of custom-trained ResNet18 model (left) and pre-trained ImageNet-1 ResNet18 model (right) on prostate cancer images (Dataset 2).

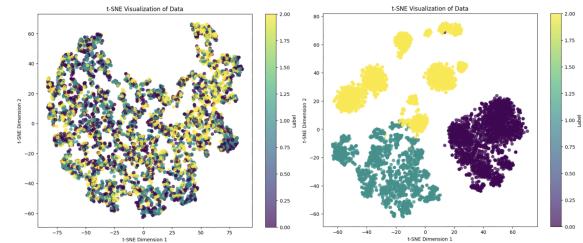


Figure 8. t-SNE dimensionality reduction visualization comparison of custom-trained ResNet18 model (left) and pre-trained ImageNet-1 ResNet18 model (right) on animal faces images (Dataset 3).

### 3.2.3 SVM Classification from Extracted Features

The performance of the Support Vector Machines (SVMs), applied to classify features extracted from the ResNet18 encoders, revealed significant insights into the generalization capabilities of both the custom-trained ResNet18 encoder and the ImageNet-pretrained ResNet18 encoder across Datasets 2 and 3. The SVM parameters for all scenarios, optimized through grid search, consistently selected the radial basis function (RBF) kernel with a regularization parameter  $C = 10$ . For Dataset 2, the optimal gamma parameter varied between  $\gamma = 0.01$  for the custom-trained encoder and  $\gamma = \text{scale}$  for the ImageNet-pretrained encoder. Similarly, for Dataset 3, the same gamma parameters were chosen for their respective encoders.

For Dataset 2, the SVM classification results seen in Table 4 showed that features extracted by the custom-trained ResNet18 encoder achieved an accuracy of 92%, with strong performance across all three classes. The precision, recall, and f1-scores for individual classes indicated balanced predictions, with a slight edge in Class 2, where the SVM reached a near-perfect f1-score of 0.98. However, the ImageNet-pretrained encoder outperformed the custom-trained encoder, achieving an overall accuracy of 98% and is represented in Table 5. This superior performance was reflected across all metrics, with f1-scores exceeding 0.97 for all classes. The enhanced separability of the features

extracted by the pretrained encoder underscores the advantages of leveraging large-scale datasets such as ImageNet for general-purpose feature extraction.

Class	Precision	Recall	F1-Score	Support
0	0.89	0.87	0.88	417
1	0.89	0.90	0.89	390
2	0.98	0.98	0.98	393
Accuracy	0.92 (1200)			
Macro Avg	0.92	0.92	0.92	1200
Weighted Avg	0.92	0.92	0.92	1200

Table 4. SVM Test Classification Report for Dataset 2. The SVM model, using extracted features from the ResNet18 model with parameters  $C = 10$ ,  $\gamma = 0.01$ , and RBF kernel, achieved an overall accuracy of 92%.

Class	Precision	Recall	F1-Score	Support
0	0.96	0.98	0.97	417
1	0.98	0.97	0.98	390
2	0.99	0.99	0.99	393
Accuracy	0.98 (1200)			
Macro Avg	0.98	0.98	0.98	1200
Weighted Avg	0.98	0.98	0.98	1200

Table 5. SVM Test Classification Report for Dataset 2. The SVM model, using extracted features from the ImageNet-1k ResNet18 model with parameters  $C = 10$ ,  $\gamma = \text{scale}$ , and RBF kernel, achieved an overall accuracy of 98%.

In Dataset 3, the results highlighted a stark contrast between the two encoders and is apparent when comparing Table 6 and Table 7. The custom-trained ResNet18 encoder achieved an overall accuracy of 66% (Table 6), with moderate performance on precision, recall, and f1-scores across all classes. The SVM demonstrated challenges in distinguishing between Class 0 and Class 1, with the lowest f1-score observed in Class 0 at 0.61. Conversely, the ImageNet-pretrained ResNet18 encoder achieved perfect classification performance, with 100% accuracy (Table 7) and f1-scores across all classes. This remarkable performance illustrates the generalization strength of the pretrained encoder, particularly in a domain as visually distinct as animal faces, which the custom-trained encoder struggled to capture effectively.

The results reaffirm the versatility and robustness of ImageNet-pretrained features in diverse domains, as well as the importance of domain-specific training when sufficient data is available. The comparative performance of the SVM classifiers demonstrates the complementary roles of transfer learning and classical machine learning in building scalable and adaptable image classification pipelines.

Class	Precision	Recall	F1-Score	Support
0	0.61	0.61	0.61	404
1	0.67	0.63	0.65	389
2	0.70	0.74	0.72	409
Accuracy	0.66 (1202)			
Macro Avg	0.66	0.66	0.66	1202
Weighted Avg	0.66	0.66	0.66	1202

Table 6. SVM Test Classification Report for Dataset 3. The SVM model, using extracted features from the Task 1 ResNet18 model with parameters  $C = 10$ ,  $\gamma = 0.01$ , and RBF kernel, achieved an overall accuracy of 66%.

Class	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	404
1	1.00	1.00	1.00	389
2	1.00	1.00	1.00	409
Accuracy	1.00 (1202)			
Macro Avg	1.00	1.00	1.00	1202
Weighted Avg	1.00	1.00	1.00	1202

Table 7. SVM Test Classification Report for Dataset 3. The SVM model, using extracted features from the ImageNet-1k ResNet18 model with parameters  $C = 10$ ,  $\gamma = \text{scale}$ , and RBF kernel, achieved an overall accuracy of 100%.

### 3.3. Discussion

The ImageNet-1k ResNet18 model with a non-linear Support Vector Machine (SVM) as a classification head performed better on prostate cancer (Dataset 2) and animal faces (Dataset 3) images than the ResNet18 model pretrained on colorectal cancer images (Dataset 1) and with a SVM classification head; this is further visualized through t-SNE dimensionality reduction.

Theoretically, the ImageNet ResNet18 model should perform better in general when applied to new random datasets on average, then the ResNet18 model trained specifically on prostate cancer tissue sample images, where the lack of variation, when compared to the complex world we live in with different textures, shapes, macro-structures (ie. tires, human faces, numbers) in the data, accounts for the results of this project.

Nevertheless, these findings do not diminish the versatility and effectiveness of Convolutional Neural Networks (CNNs) in transferring learned features across domains. The ability of the task-specific ResNet18 model to extract meaningful features from the prostate cancer and animal faces datasets and achieve notable classification accuracy underscores the robustness of CNN-based transfer learning approaches. This study highlights the potential of leveraging feature extraction techniques, even from specialized datasets, to address classification tasks in unrelated domains with commendable performance.

## References

- [1] Chenjie Gu and Minkyu Lee. Deep transfer learning using real-world image features for medical image classification, with a case study on pneumonia x-ray images. *Bioengineering*, 11(4):406, 2024. [2](#)
- [2] S. Binta Islam and et al. Animal species recognition with deep convolutional neural networks from ecological camera trap images. *Animals*, 13(9):1526, 2023. [2](#)
- [3] Dorota Kwiatkowska, Patrycja Kluska, and Adam Reich. Convolutional neural networks for the detection of malignant melanoma in dermoscopy images. *Postepy dermatologii i alergologii*, 38(3):412–420, 2021. [2](#)