Building the Best Pipeline for Histology Patch Classification

Abstract

The rapid development of neural networks has revolutionized artificial intelligence and enabled strong performance in image classification and pattern recognition, with applications in high-stakes domains such as healthcare. However, widespread application is hindered by persistent challenges such as inefficiencies in classification and vision tasks, where models struggle with noisy data, overfitting, and generalization. Despite large investments, many of these issues remain unsolved, and benchmarking of classification performance is not saturated. This study benchmarks multiple classifier architectures—attention pooling, K-Nearest Neighbors (KNN), and a linear Multi-Layer Perceptron (MLP)—on consistent histology patch embeddings from eight foundation models (e.g., Virchow2) based on the NCTCRCHE100K dataset. We evaluate performance using accuracy, F1-score, precision, recall, confusion matrices, and ROC curves, and analyze trade-offs between accuracy, complexity, and computational cost. Results show that attention pooling on patch tokens consistently achieves the best F1 scores, while MLP on CLS tokens offers a strong, efficient baseline. We discuss training dynamics, overfitting, and practical considerations for deployment, and outline directions for improving robustness and clinical relevance.

Keywords: Neural Network Architectures, Classification, Vision Transformers, CNNs, Medical Datasets, Performance Improvement

Table Of Contents

1. Introduction

The growth of AI in the past few years has been so rapid that the past constrictions of it are slowly forgotten. What began as rule-based systems in the mid-20th century, limited by computational power and simplistic algorithms, has evolved into sophisticated neural networks capable of classifying medical data with high accuracy and analyzing complex images for diagnostic purposes. This transformation accelerated dramatically with the transfer from machine learning to deep learning in the 2010s, encourage and fueled forwards by improvements in hardware (e.g., GPUs) and massive datasets. Creating milestones achievement since then, such as the convolutional neural networks (CNNs) for visual tasks (Krizhevsky et al., 2012). In addition, the introduction of Transformers in 2017 revolutionized sequence and image modeling, paving the way for vision transformers (ViTs) and beyond (Vaswani et al., 2023). Today, AI is vastly useful in healthcare—from predictive models aiding tumor detection to automated analysis of X-rays for disease identification—suggesting strides toward more reliable diagnostic tools. The scale is staggering, models now boast trillions of parameters, trained on datasets encompassing vast medical records, enabling applications in oncology, radiology, and personalized medicine.

However, this exponential growth conceals inherent weaknesses that compromise AI's reliability in clinical settings. Beneath the appeal of scaling—embodied in models such as GPT-4 variants with billions of parameters—lie foundational problems that implied bigger is not necessarily a worthy tradeoff (OpenAI et al., 2024). A case in point is inefficiencies in straightforward classification tasks, wherein baseline architectures overfit on tabular medical data, producing erroneous predictions (e.g., classifying benign tumors as malignant). Analogous problems afflict visual inspection tasks (e.g., object detection in X-rays), wherein CNNs break down under noisy conditions such as image quality variations, and attention-based alternatives such as ViTs shine (Dosovitskiy et al., 2021). These weaknesses are especially pronounced in histology, where classifying tissue patches for diseases such as cancer involves managing subtle morphologic differences, class imbalances, and high-dimensional features—tasks for which traditional architectures perform poorly in feature extraction, spurious correlations, and generalization to heterogeneous medical settings. Scaling parameters exacerbates rather than alleviates these defects, as supported by accuracy plateaus in models despite model size increases.

This essay contends that architectural innovations—such as integrating attention mechanisms, residual connections, or hybrid architectures—provide a more viable route to performance gains than brute-force scaling alone. By interrogating how targeted modifications in network architecture improve results across medical tasks, we reveal an uncomfortable truth: the future of AI in healthcare resides in smarter, not merely bigger, designs. Our chief objective is to benchmark classification performance on histology patch embeddings, contrasting architectures such as attention pooling, K-Nearest Neighbors (KNN), and Linear Multi-Layer Perceptron (MLP) to discern the most suitable for tissue/disease classification. This data-centric approach prioritizes hand-curated, balanced datasets (≥10,000 patches per class) and state-of-the-art embeddings (e.g., Virchow2) for strong generalization. Technical contributions comprise investigating classifier trade-offs and strict evaluations (e.g., confusion matrices, ROC curves), while disease-specific implications provide clinical relevance, e.g., automating tumor grading or metastasis detection to support pathologists.

To this end, the research questions are: (1) How do architecture changes (e.g., incorporating attention layers) enhance accuracy and robustness for basic classification tasks, through experiments comparing baseline feedforward networks to improved variants on the for tumor detection; (2) To evaluate classifiers on histology embeddings for disease classification, and examine implications for oncology. These questions extend previous research, including architectural evolution studies in medical AI and vision models, while making new contributions through analyses specific to healthcare datasets.

1.1. Neural Network Architectures

Deep learning, a subset of machine learning that originated from the study of artificial neural networks (ANNs), stands out for overcoming reliance on hand-designed features. Traditional machine learning techniques such as Support Vector Machines (SVM) and K-nearest neighbors (KNN) require explicit feature engineering and often struggle with visual image processing and human speech, while deep learning can learn from the data and make data-driven predictions even if not explicitly programmed.

In 2006, Geoffrey Hinton proposed a fast-learning algorithm for Deep Belief Networks (DBNs), a deep structured learning architecture often cited as marking the birth of modern deep learning. It addressed limitations in artificial neural networks, especially getting trapped in local optima caused by backpropagation training. Hinton’s layer-wise greedy learning offered a solution by performing unsupervised pre-training before supervised fine-tuning, improving initialization, convergence, and robustness to overfitting, especially when labeled data is scarce (Hinton et al., 2006). The success of DBNs paved the way for more efficient techniques, continuing to advance with self-supervised learning and Transformers.

1.2. Challenges in Histology AI

In the field of histology, the classification of images presents unique computational challenges for modern day classifiers. Addressing such a problem requires a strong processing approach, because as a high-stakes domain any blunder in output could propose serve ethical concerns. For a successful classification a model must face three main problems.

1.2.1. The “Gigapixel” Image Problem:

Standard histology images are data heavy. A single Whole-Slide Image (WSI) can be 100,000 × 100,000 pixels or larger, resulting in gigapixel-scale data. This requires substantial computational power to process, which is time and energy costly.

1.2.2. “Needle in a Haystack” Problem:

In many diagnostic scenarios, the desired feature of interest may be extremely small or could be exceptionally rare within the WSI. It be inefficient to process 99% of an image with unnecessary features, and the extreme imbalance may also create uncompetitive training because it might fail to learn the critical features but rather focus on the majority class.

1.2.3. Inconsistent variability problem:

Histology slide appearances are variable and inconsistent between datasets. Differences in institutional protocols create variation in color, luminosity, and contrast, stemming from reagents, staining, slide preparation, and scanner calibration. This variability challenges generalization and thus applicability across datasets.

Addressing these basic challenges—processing gigapixel-sized data, identifying rare diagnostic features, and generalizing across variable staining protocols—will be essential to creating clinically feasible AI systems that can improve pathological diagnostics without sacrificing accuracy or ethical standards. We aim to investigate the feasibility of applying vision transformers as a feature encoder to solve these tasks.

1.3. Related Works

1.3.1. Transformers

The original encoder-decoder transformer architecture was first introduced in 2017 in the “Attention is all you Need” paper. (Vaswani et al., 2023) An innovative approach towards sequence transduction problems, the Transformer model disregards the primary strategies of recurrence and convolution, instead relies solely on the attention mechanism. These attention weights are used to perform a weighted sum of the values. This self-attention computes the relationship between pairs of tokens in a single layer, as a result the entire context is captured within a single layer.

The Transformer structures utilize this attention function into the encoder and decoder. Within the encoder, all keys, values, and queries come from the output of a previous layer. Similarly, in the decoder each self-attention layer allows each position to attend to all position (inclusively) before its layer. In addition, the Transformer using two special algorithms, namely the Scaled Dot-Production Attention and Multi-Head Attention. In the scaled-dot-product mechanism, it computes a weighted sum of values based on the correlation between query and key. This dynamically filters and combines information globally, unlike CNNs’ fixed local kernels.

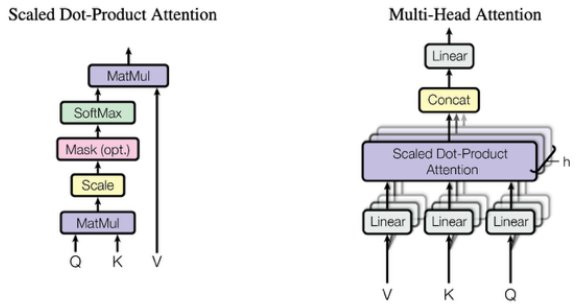

Figure 1. Standard transformer architecture.

The Transformer also uses embeddings (latent space); embeddings are learned representations used to convert input tokens into vectors of a specific dimension. Because the Transformer does not contain recurrence and convolution, positional encodings are added to the input embeddings at the bottom of the encoder and decoder

stacks. These positional encodings give information about the relative position of the tokens in a sequence, and then the Transformer can achieve a global understanding. In the end, the Transformer uses a learned linear transformation and a softmax function, which converts the decoder output to predicted next-token probabilities. Overall, what made the Transformer such a game changer was its ability for advanced parallelization during training, which recurrent models fail to excel at, drastically reducing training hours while achieving superior quality on machine translation tasks.

The attention mechanism's ability to model long-range dependencies without convolution is directly leveraged by vision backbones used to produce histology patch embeddings (e.g., Virchow2, UNI 2). In our study, these Transformer-based embeddings provide either CLS tokens (global) or patch tokens (spatial), on top of which we benchmark simple classifier heads (attention pooling, MLP, KNN).

1.3.2. Vision Transformer

In the original Vision Transformer paper, they detailed three main variants of the Vision Transformer: ViT-Base, ViT-Large, and ViT-Huge (Table 1). The notation for these models often includes the patch size; for example, "ViT-L/16" means the Large variant with 16×16 input patch size. It is important to note that the Transformer's sequence length is inversely proportional to the square of the patch size, meaning models with smaller patch sizes are computationally more expensive due to longer sequences. In addition, the paper indicates that larger ViT variants (L and H) fully benefit only when pre-trained on sufficiently large datasets like JFT-300M. When pre-trained on smaller datasets like ImageNet, they can underperform compared to smaller ViT models or even ResNets of comparable size, as they are more prone to overfitting due to their fewer inductive biases. However, on large datasets, larger ViT models (e.g., ViT-H/14) demonstrate superior performance and efficiency compared to CNNs. Scaling the depth of the Transformer results in the biggest performance improvements, while scaling the width results in the smallest changes (Dosovitskiy et al., 2021).

Specification	ViT-Base (ViT-B)	ViT-Large (ViT-L)	ViT-Huge (ViT-H)
Transformer Encoder Layers	12	24	32
Hidden Size (D)	768	1024	1280
MLP Size	3072	4096	5120
Attention Heads	12	16	16
Total Parameters	86M	307M	632M

Table 1. ViT architecture sizes.

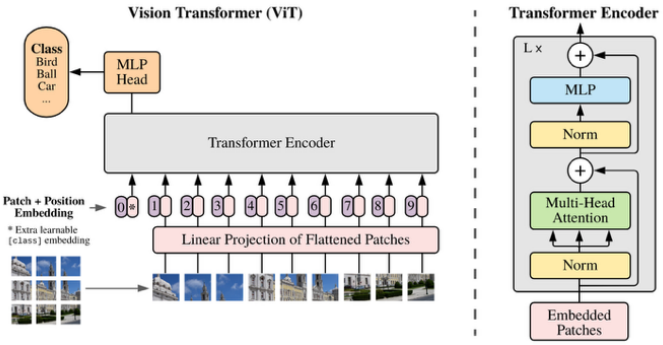

Figure 2. Vision transformer architecture.

1.3.3. Self-Supervised Training – DINO

DINO employs a self-supervised learning approach, an alternative to text-guided pretraining that can be interpreted as a form of knowledge distillation with no labels. DINO uniquely applies a student-teacher network architecture, two separate networks that processes two different views of an image. Both local (smaller resolution) and global views (bigger resolution) are passed through the student network, while only the global views are passed through the teacher network. The teacher's parameters are updated using an exponential moving (EMA) based on the student's parameters. Consequently, the teacher achieves a higher quality than the student earlier, allowing it to guide the student later towards higher-quality features. A stop-gradient operator is applied to the teacher's output, preventing both networks from simply learning to output a constant. Another crucial operation during DINO's training involves minimizing the cross-entropy loss between the teacher's output and the student's output; DINO employes centering and sharpening to the teacher's output to prevent the model from collapse. (Caron et al., 2021) A newer version of DINO, DINOv2 directly builds upon its predecessors DINO and iBot

(Zhou et al., 2022), focusing on scaling pretraining. Largely alike, by replacing the centering algorithm and adding a short, high resolution training phrase, DINOv2 improved its performance furthermore on pixel-level tasks. (Oquab et al., 2024) This self-supervised technique shows great potential in developing foundation models when combined with vision transformers, as it removes the need for extended human labelling. Many of the foundation models we evaluated are trained with DINO/DINOv2 and related SSL recipes.

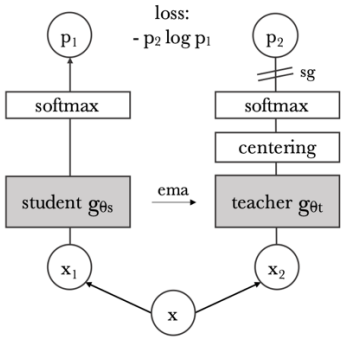

Figure 3. Dino self-supervised student-teacher training framework.

1.3.4. Foundation models

A foundation model is an AI neural network — trained on mountains of raw data, generally with unsupervised training— that can be adapted to accomplish a broad range of tasks. (Merritt, 2025) Regarded as the common basis for many task-specific model, examples include BERT, DALL-E, and GPT-3. The importance of foundation models can be summarized by two key concepts:

Emergence: This refers to behavior of a system that is implicitly induced rather than explicitly constructed. It is a source of both excitement and anxiety due to unanticipated consequences. For example, advanced functionalities like in-context learning in GPT-3 are emergent properties that were neither specifically trained for nor anticipated to arise.

Homogenization: This indicates the consolidation of methodologies for building machine learning systems across a wide range of applications. While it provides powerful leverage, meaning improvements in the foundation model can benefit many downstream applications, it also demands caution because defects or problematic biases in the foundation model are inherited by all adapted models downstream. This can lead to an "algorithmic monoculture" where consistent and arbitrary rejection or misclassification affects individuals.

Despite some challenges, the obvious benefits of foundation models make them crucial and extremely helpful for histology. Traditional AI methods often require large amounts of labeled data and are specialized for specific tasks. Foundation models, pre-trained on massive datasets, can be adapted to new tasks with less labeled data. Addressing cost issues and minimizing expert data that if often limited and expensive at the same time. In the context of histology, this could mean aiding pathologists in faster and more accurate diagnoses by leveraging vast amounts of image and patient data. In addition, due to its strong generative ability, clinics could potentially use foundation models as assistant to generate detailed tasks for a treatment or experiment. In connection to our research, we evaluate classifier heads on top of fixed embeddings from eight pathology foundation models. Our contribution is to benchmark, in a controlled setting, how simple yet representative heads exploit either CLS tokens (global) or patch tokens (spatial), quantifying trade-offs in accuracy versus computational/storage cost on a standardized histology dataset.

2. Embedding Models

In pathology, foundation model can generate embeddings that capture important features and information of a slide, enabling precision detections and predictions. Recently models such as Virchow and UNI demonstrated the potential of foundation models to reshaping pathology, where these models can act as a trustable collaborator in a diagnosis.

2.1. Virchow & Virchow2

The second generation of the Virchow family, Virchow2 is a foundation model for computational pathology. It

aims to generate data representations, known as embeddings, that can generalize well to a variety of downstream tasks. Virchow is also specifically designed to capture diverse patterns in histology images to aid cancer detection, cancer subtyping, biomarker quantification, mitotic event counting, and survival prediction. Virchow was trained on data from about 100 thousand patients acquired from Memorial Sloan Kettering Cancer Center (MSKCC), a total of 1.5 million hematoxylin and eosin-stained whole slide images. Including both cancerous and benign, 17 high-level tissue types collected via biopsy(63%) and resection(37%). Using this dataset, researchers used self-supervised learning (SSL), specifically the DINOv2 algorithm. The result is a 632 million parameter vision transformer model, achieving 0.949 overall specimen-level AUC across the 17 different cancer types. The more recent Virchow2 maintains the same parameter, but it comes with an enhanced training. Upscaling the dataset from 1.5million WSIs to 3.1million WSIs, dramatically expanding the diversity of the dataset covering global data and containing nearly 200 recorded tissue types compared to 17 for the original Virchow.

2.2. UNI & UNI 2

Based on the vision transformer (ViT-Large), UNI is another general-purpose self-supervised model for computational pathology. They identify current problems existing, such as limited size and diversity of pretraining data, as the TCGA is comprised of mostly primary cancer histology slides. Furthermore, limited evaluation of generalization performance across diverse tissue types, as many pan-cancer analyses and popular clinical tasks in CPath are also based on annotated histology region-of-interests. Similar to the Virchow models, UNI it is also trained on the DINOv2 algorithm, but on the Mass-100k dataset. Mass-100k is an extremely large histology slide collection consisting of over 100 million tissue patches sourced from Massachusetts General Hospital (MGH), Brigham & Women's Hospital (BWH), and Genotype-Tissue Expression (GTEx). consortium). This dataset also covers 20 major tissue types, including cancer tissue, normal tissue, and other pathologies. Resultingly, it outperforms REMEDIS and CTransPath on TCGA-evaluated data, with improvements on tasks with high diagnostic complexity. The improved version, UNI 2 is trained on a significantly larger and more diverse dataset. Combined with the fact it uses a Vit-Huge/14(Vit-h/14) as a backbone, UNI 2 showcases roughly 3-13% better performance than the original UNI model.

2.3. Hibou-B & Hibou-L

The Hibou family consists of two main Vision Transformer (ViT) variants: Hibou-B, based on the ViT-B/14 architecture, and Hibou-L, based on the ViT-L/14 architecture. Both models are pretrained using the DINOv2 framework. Specifically, Hibou-B was trained on 8 A100-80G GPUs for 500,000 iterations, while Hibou-L was trained on 32 A100-40G GPUs for 1.175 million iterations, both with a total batch size of 1024 and randomly initialized weights. Hibou was trained on a proprietary histopathology dataset that comprises over 1 million Whole Slide Images (WSIs), specifically 936,441 H&E and 202,464 non-H&E-stained slides, sourced from 306,400 unique cases. For training, WSIs are split into non-overlapping patches, with background patches filtered out. Hibou-L used 1.2 billion "clean" patches, while Hibou-B used 512 million, ensuring each unique patch is sampled only once per training.

2.4. Pikon & Pikon-v2

Phikon, is a ViT-Base (ViT-B/16) model pretrained using the iBOT algorithm on 43 million histology tiles from 6,000 TCGA whole slide images (WSIs). Building upon this, Phikon-v2 is an improved variant, utilizing a Vision Transformer Large (ViT-L/16) architecture and pretrained with the DINOv2 framework. Its training dataset, PANCAN-XL, is significantly larger and more diverse, comprising over 460 million histology tiles extracted from more than 55,000 publicly available slides. PANCAN-XL includes data from 132 public datasets and 4 internal datasets, covering more than 30 cancer sites and normal tissues, with main sources being TCGA, CPTAC for malignant tissue, and GTEx for normal tissue. Phikon-v2 surpasses the previously released Phikon model with an overall gain in AUC of almost 2 points.

3. Evaluation

3.1. Dataset

For this research we will primary be using the NCTCRCHE100K dataset as our source of patch data. NCTCRCHE100K is an up-to-date, open-source, large-scale dataset specifically designed for computational histology research. The diverse visual features these images provide by NCTCRCHE100K are crucial in classification in the field of histology for AI models to identify, classify, and segment various pathological features through training.

Many existing and previous datasets often lack such sufficient sample size and diversity, which will significantly limit the generalization and robustness of AI models. In addition, some of past datasets only focus primarily on a

certain diagnostic task and tissue, further limiting its applicability, for example: CAMELYON16, this dataset includes 400 WSIs of breast cancer sentinel lymph nodes. Built to benchmark algorithms specifically for detecting metastases in lymph node tissue. However, its limited and narrow focus on a single cancer type's diagnostic task demonstrate the kind of constraints that the larger, more diverse HISTAI dataset aimed to overcome.

On the other hand, NCTCRCHE100K successfully mitigates these gaps seen in past datasets, providing diverse collection of different cases, proving helpful in finding more robust AI solutions. With a set of 100,000 non-overlapping image patches from hematoxylin & eosin (H&E) stained histological images of human colorectal cancer (CRC) and normal tissue. All images are 224x224 pixels (px) at 0.5 microns per pixel and are color-normalized using Macenko's method. (Macenko et al., 2009) This data set include 9 class, separately these tissue classes are: Adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus (MUC), smooth muscle (MUS), normal colon mucosa (NORM), cancer-associated stroma (STR), colorectal adenocarcinoma epithelium (TUM). These images were manually cropped from N=86 H&E stained human cancer tissue slides of formalin-fixed paraffin-embedded (FFPE) samples from the NCT Biobank (National Center for Tumor Diseases, Heidelberg, Germany) and the UMM pathology archive (University Medical Center Mannheim, Mannheim, Germany). Tissue samples included CRC primary tumor slides and CRC liver metastasis tumor tissue; normal tissue classes were supplemented with non-tumorous areas from gastrectomy specimen to enhance variability. (*DykeF/NCTCRCHE100K · Datasets at Hugging Face*, n.d.)

We experiment with eight foundation models: Virchow, Virchow2, UNI, UNI2, Hibou-B, Hibou-L, Phikon, and Phikonv2. These models are used to embed medical image patches into fixed-dimensional vectors. Subsequently, supervised classifiers are trained on these embeddings to predict patch-level labels.

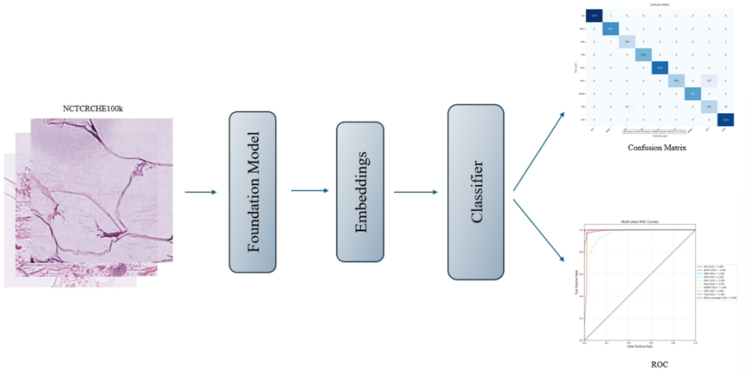

Figure 4. Patch classification pipeline. The foundation model and classifier model layers would be alternated.

To ensure a fair comparison, we fix the dataset, preprocessing, and train/validation splits across all experiments. This isolates the effect of the classifier head (attention pooling on patch tokens, MLP on CLS tokens, KNN with PCA) given identical foundation-model embeddings and evaluation settings.

3.2. Feature Extraction

We convert the NCT-CRC-HE-100K images and their validation counterparts into embedding features suitable for downstream classification and store them in an efficient numerical format. The procedure comprises four stages: (1) data loading and preprocessing, (2) model initialization, (3) feature extraction, and (4) feature storage. We extract both global [CLS] tokens and spatial patch tokens to enable a direct comparison between classifier heads that rely on global summaries versus spatial reweighting.

3.2.1. Data Loading and Preprocessing

We loaded the images locally with their labels inferred from standardized filenames. Each Foundation model's open sources preprocessing (normalization, resizing, color space handling) is applied to ensure the input distribution matches the model's pretraining regime. This alignment mitigates the risk of preprocessing induced performance variances and helps to maintain comparability across classifier heads.

3.2.2. Model Initialization

Each foundation model is instantiated with its publicly released architecture and pretrained weights, then set to inference mode. The encoders are frozen: no fine-tuning is performed during feature extraction. This design choice ensures that differences observed in downstream performance can be attributed to the classifier head rather than to updates in the backbone, mirroring controlled evaluations in prior work on transferable visual features.

3.2.3. Feature Extraction Process

This is the center of this section of the feature extraction pipeline, where model processes the images and extracts the features. Firstly, we perform a safety check on available disk space and calculating the estimated disk space for storing the features. Considering that the final files are considerably large reaching around 100GB depending on the model, this step is crucial for workflow efficiency. Afterwards, we feed the images to the model, processing the images in large batches to fully utilizes the GPU's parallel processing capabilities. For each input image, the visual encoder produces a token sequence from which we collect: The [CLS] token, a global representation of the entire image; the Patch tokens, per-patch embeddings capturing localized morphological cues.

3.2.4. Feature Storage

Embeddings and labels are stored in a hierarchical numerical format, separating into three different classes: cls_token, patch_token, and labels. This separation enables explicit measurement of the storage/computation trade-offs between CLS-only and patch-attention heads, which we later discuss in terms of efficiency and training dynamics. To optimize future usage, we create the datasets with the chunking enabled, storing the data in blocks for efficient file I/O. In addition. we write the extracted features batch by batch into this file.

3.3. Classification Head

This section of the paper will detail the design and process of training an effective classification model based on the pre-extracted features on the dataset by the foundation models. We mainly focused on three designs: an attention-based model, a K-Nearest Neighbors (KNN) model, and a multilayer perceptron (MLP) model. We evaluated each design on its ability to classify accurately, using metric such as accuracy, F1-score, precision, and recall.

3.3.1. Data Loading

This process is optimized for dealing with large datasets that will not fit within memory, our embeddings are accessed following a "lazy loading" strategy. The HDF5 file is not loaded into memory upon initialization of the dataset. For each sample, either patch tokens and labels (attention head) or CLS tokens and labels (KNN, MLP) are retrieved. Training data are shuffled at each epoch to enhance generalization. This regimen minimizes I/O bottlenecks and keeps throughput comparable across methods.

3.3.2. Model Architecture

Most of the differences within different classification model occur in the design of the model architecture. Different models have different complexities, inductive biases, and feature extraction capabilities, which influence their performance on specific tasks.

3.3.2.1. Attention Model

This model’s forwards pass can be divided into four main steps: projection, attention weight calculation, weight aggregation, and finally classification. During the projection stage, patch tokens derived from the foundation model are linearly mapped to a lower-dimensional space, enabling the network to learn compact, semantically meaningful representations. Lowering the dimension of the patch tokens to help the model learn a more meaningful, compact representation. These patch tokens are then processed by the attention mechanism, this key network will learn to assign a weight to every patch. Patches with meaningful and important information will receive a high weight, creating contrast with lower weighted background. The calculated attention weights are then multiplied with their corresponding projected patch tokens. This scales each patch's feature vector by its learned importance. These weighted features are then summed up to produce a single, aggregated feature vector. This is a form of attention pooling, resulting in a final vector with high relevance towards the focus of the image. In the final step, this feature vector is passed into a feed-forward classifier, producing final output logits of the 9 different classes. The logits are fed through a standard cross-entropy loss for loss calculation and gradient backward propagation.

3.3.2.2. KNN

The K-Nearest Neighbors (KNN) with PCA model pipeline employs a classical machine learning approach that pairs a KNN classifier with Principal Component Analysis (PCA) for dimensionality reduction. Instead of relying on deep learning, KNN is a non-parametric, instance-based learning method that memorizes the training data. Here, global feature vectors (CLS tokens) from the foundation model are flattened and projected into a reduced-dimensional representation using incremental PCA, which efficiently handles large datasets by retaining dominant variance while filtering out noise. At inference, classification of new images involves calculating distances (e.g., Euclidean) to stored vectors and selecting the k nearest neighbors (our model uses k=5). The final class prediction is obtained by a weighted vote where closer neighbors have more influence. This model balances powerful dimensionality reduction with a simple, instance-based classifier,

offering a computationally efficient baseline compared to deep neural networks.

3.3.2.3. MLP

The Multi-Layer Perceptron (MLP) classifier model is based on a lightweight feed-forward neural network, building on the global CLS token. The network architecture comprises several layers: an input linear layer reduces the dimensions of CLS token; followed by a Rectified Linear Unit (ReLU) activation introducing non-linearity; and a Dropout layer to reduce overfitting. A second hidden layer further reduces the feature size before the output layer maps the features to 9 logits, corresponding to the different tissue classes. These logits are used with cross-entropy loss, which internally applies softmax to compute probabilities during training. This MLP transforms the CLS token into a feature space that is linearly separable for the final classification, giving the final label.

3.3.3. Training Manager

The training manager handles the training of the models, incorporating the most suitable functions for model training and validation, and coordinates the core components, including the foundation model, the classification model, Adam optimizer for training, and corresponding loss functions used for multi-class classification. Training proceeds in epochs, for each batch it performs a forward pass in train mode, loss calculation, gradients backpropagation, and parameter updates. In the validation loop, the model is switched to eval mode (disabling dropout), and gradient calculations are turned off for efficiency. It runs the model on the validation set to compute the loss and accuracy on data it has not seen during training. To prevent overfitting, an EarlyStopping mechanism is used, it monitors the validation loss after each epoch and halts training if no meaningful improvement is observed within a defined patience interval. Importantly, the parameters from the epoch yielding the lowest validation loss are retained as the final model state. Training and validation statistics are recorded, enabling real-time monitor of the model's learning progress across epochs.

3.3.4. Evaluation and Visualization

Finally, we perform an evaluation and visualization to help us analyze the models' performances. Using the parameters preserved at the optimal epoch we collect the true labels, the predicted labels, and the class probability distributions for each class. Performance is first analyzed through a confusion matrix, which reveals the distribution of correct and incorrect predictions across all classes, accompanied by accuracy, precision, recall, and F1-score. This visual tool provides a clear breakdown of the model's performance, showing exactly which classes are being correctly identified and which are being confused for others. Furthermore, we plotted an individual ROC curve for each class using the one-vs-rest scheme and calculated the Area Under the Curve (AUC) metric for each curve. We also compute and plotted a macro-average ROC curve, which provides a single, aggregate measure of the model's separability performance across all classes.

4. Results

The following table shows our results obtained from our steps described in section 4. We benchmarked all these models based on their patch level classification accuracy, precision, recall, F1-score. We compared 8 different feature extraction model mentioned in section 2 against each other. Noticeably, to identify general performance trends, it is necessary to aggregate performance across multiple tasks which plotted as three distinctive areas: MLP, KNN, and attention pooling. We aim to spot whatever a given model shows superior performance comparably overall with respect to other models from an analytical standpoint. In total we spend around 50 GPU hour altogether to perform our feature extraction and classification model training; in addition, roughly over 600 gigabytes of data generated from models, dataset, and dataset feature extractions.

	KNN				**MLP**				**Attention Pooling**			
Model	*Acc*	*P*	*R*	F_1	*Acc*	*P*	*R*	F_1	*Acc*	*P*	*R*	F_1
Virchow	0.964	**0.953**	**0.960**	**0.952**	0.972	0.958	0.960	0.957	0.970	0.956	0.960	0.971
Virchow2	0.960	0.946	0.953	0.946	**0.973**	**0.962**	**0.963**	**0.959**	**0.973**	**0.960**	0.961	**0.973**
Hibou-B	0.930	0.903	0.915	0.907	0.963	0.947	0.953	0.950	0.967	0.955	0.956	0.952
Hibou-L	0.918	0.893	0.900	0.895	0.954	0.931	0.941	0.935	0.964	0.948	0.949	0.946
UNI	0.963	0.945	0.947	0.945	0.963	0.942	0.948	0.944	0.970	0.954	0.959	0.955
UNIv2	**0.967**	**0.953**	0.954	0.950	0.970	0.956	0.960	0.956	**0.973**	**0.960**	**0.962**	0.958
Phikon	0.962	0.945	0.942	0.942	0.960	0.939	0.943	0.940	0.959	0.938	0.944	0.941
Phikon-v2	0.962	0.944	0.943	0.942	0.964	0.945	0.951	0.947	0.967	0.950	0.955	0.951

Table 2. Metrics for different foundation model and classification model combinations. Evaluation dataset is detailed in 4.2, feature extraction detailed in 4.3, and evaluation procedures are detailed in 4.4. For each tasks the

best performance is presented as **bold**, and the second best is <u>underlined</u>.

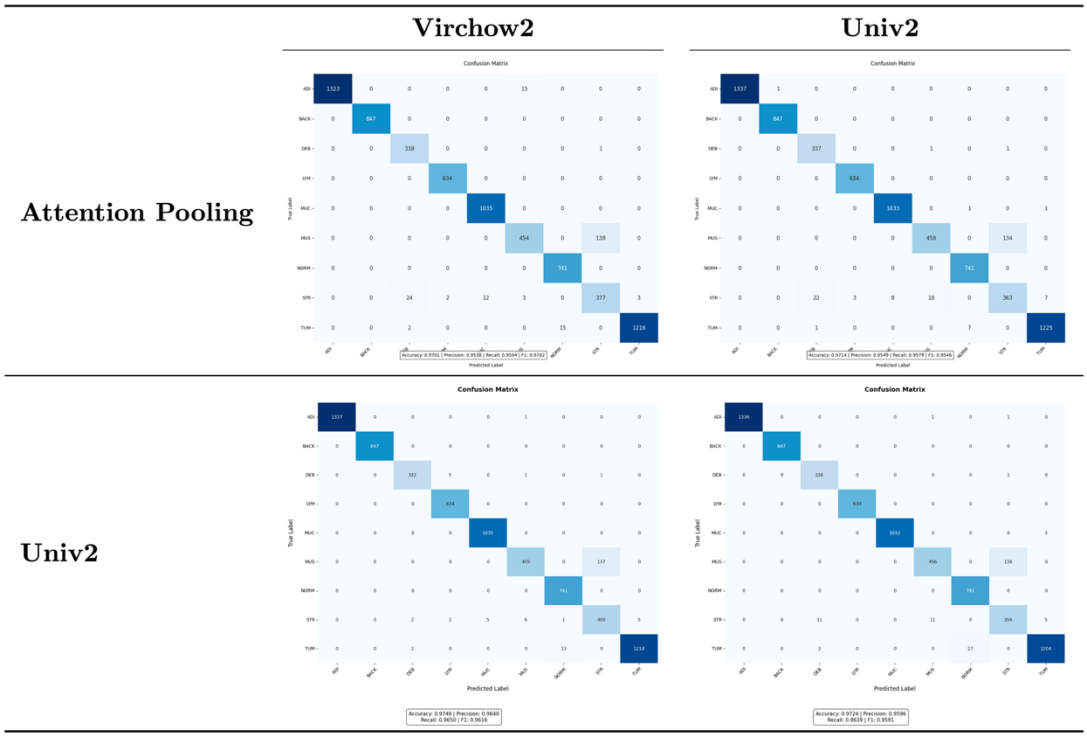

Figure 5. Confusion matrices for Virchow2 and Uni v2

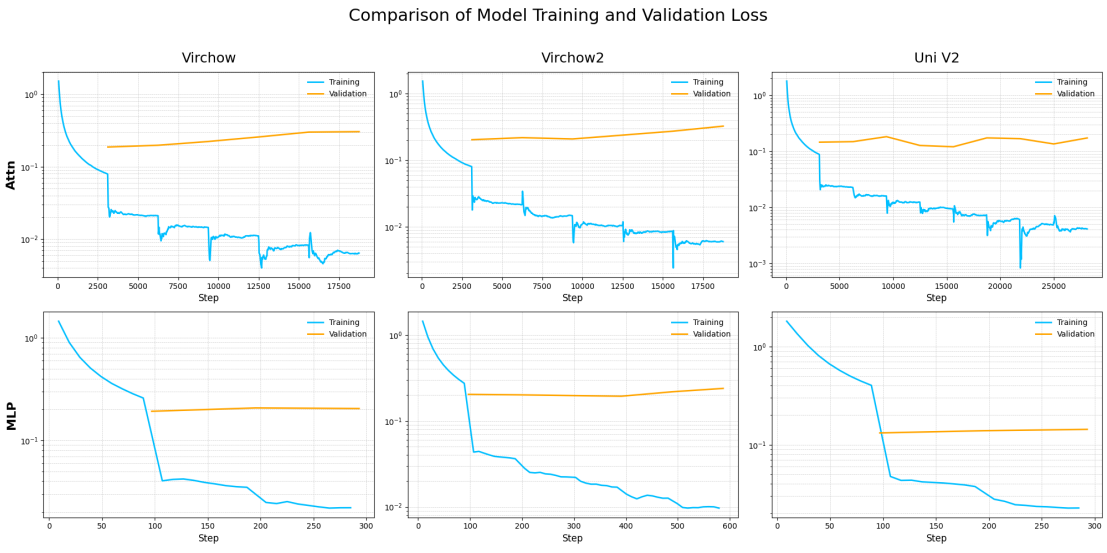

Figure 6. Tensor board logging of Model Training and Validation Loss for three models: Virchow, Virchow2, and Uni V2.

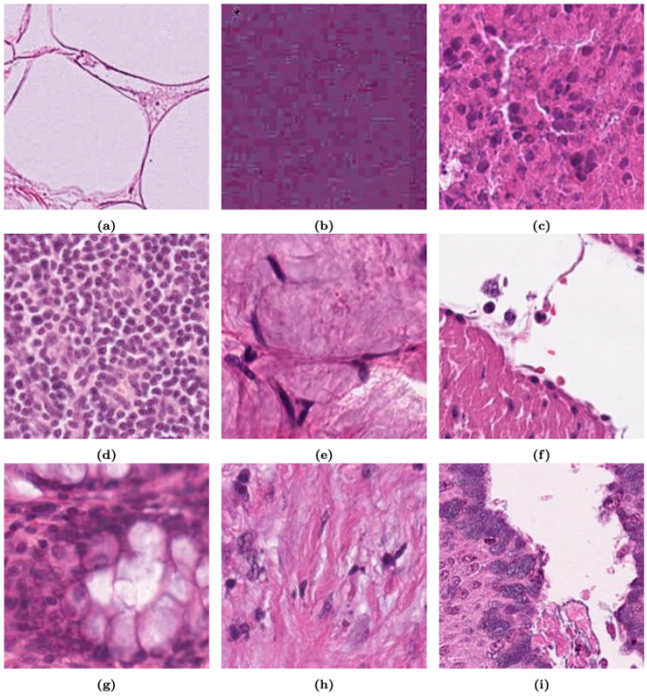

Table 3. Sample classification patches Virchow2 identified with 100% confidence: (a) Adipose (ADI), (b) background (BACK), (c) debris (DEB), (d) lymphocytes (LYM), (e) mucus (MUC), (f) smooth muscle (MUS), (g) normal colon mucosa (NORM), (h) cancer-associated stroma (STR), (i) colorectal adenocarcinoma epithelium (TUM)

4.1. High-level Findings

We can clearly see that across all classification models, attention pooling on patch tokens tend to achieve the highest F1 score. In second place and third place, MLP on [CLS] tokens and KNN. Among classification models, the most recent larger-scale models (Virchow2, UNI2, and Phikonv2) consistently outperformed their predecessors (except Hibou-L), suggesting that improving pretraining corpus diversity and scale can translate into stronger transferable patch representations. The strongest single combination was typically Virchow2 + Attention Pooling or UNI2 + Attention Pooling. For most embedding models the MLP narrowed the gap with models whose CLS token is particularly well-formed (e.g., Virchow2), but Attention always retained a margin. On the other hand, the KNN model is efficient to an extant with most models achieving a metric score of over 0.90; however, comparatively KNN encoder lagged behind other classification models by a substantial margin.

4.2. Training Dynamics

Using tensor board logging, we recorded optimization process and visualized the training loss and validation loss separately for three of the best performing models Virchow, Virchow2, and Uni v2, through the graphs we were able to notice five main observations:

Pervasive Overfitting: The most significant finding across all six experiments is the clear evidence of overfitting. In every case, the training loss (blue line) steadily decreases, while the validation loss (orange line) either remains flat or gradually increases. This indicates that while the models are effectively learning the training data, they fail to generalize this learning to new, unseen data after the initial training phase.

Rapid Convergence of MLP Heads: The MLP models (bottom row) converge extremely quickly, with the training loss dropping dramatically within the first 100-200 steps before plateauing. This suggests that the features extracted by the base models are easily learnable by a simple MLP, but further training quickly leads to overfitting.

Slower, Deeper Learning with Attention Heads: The Attention-based models (top row) are trained for a much larger number of steps. They achieve a lower final training loss compared to their MLP counterparts, suggesting a more thorough fit to the training data. However, like the MLP models, their validation loss does not improve after the initial phase.

Superiority of Virchow2 and Uni V2 Models: When comparing the base models, Virchow2 and Uni V2

consistently show better performance than the original Virchow model. They achieve lower training losses with both Attention and MLP heads, indicating that they produce higher-quality, more separable features.

Practical Implication: The results strongly suggest that the optimal model checkpoint for all configurations occurs very early in the training process. Continuing to train the models beyond this point is counterproductive, as it only improves performance on the training data while hurting the model's ability to generalize. Therefore, aggressive early stopping based on the validation loss is essential to achieve the best results.

4.3. Discussion

Overall, across all embedding models, we observe attention pooling as the highest achieving classification model. This is explainable, as we know the attention mechanism can learns to apply variable weighting over patch tokens, creating a focused view on salient micro-regions (e.g. lymphocytic aggregates) while also "ignoring" unimportant sections of down-weighted background. In addition, selfattention lets each patch "see" others when computing, so weighting reflects not just a patch's local features but its context in a global focus. This has proven to be effective at locating the sparse and localized class evidence in histology. Attention can reweight against staining/contrast shifts by focusing on morphology rather than color statistics, improving generalization under domain variance.

In comparison, we also notice that attention pooling outperforms MLP (using [CLS] token, trained via self-supervised methods), demonstrating that a supervised attention mechanism outperforms internal self-supervised learnable token. Because MLP on CLS is a shallow, linearnonlinear map on a single vector. It cannot reweight or reassemble spatial evidence lost during SSL pretraining, on contrast, attention pooling is a supervised recomposition of spatial evidence, effectively "recovering" discriminative structure that a global CLS average diluted. However, admittedly there is tradeoff in efficiency, computational requirements, and storage between MLP and attention pooling. The patch tokens for attention pooling are heavy in size. On average, saving all the patch tokens locally is around 10 to 20 times larger than saving one cls token; consequently, this inflates the need for storage and stretches the training duration by a substantial amount. For example, in our runs we can see that Attention Pooling's total training duration is 555 seconds longer than MLP's training time—roughly 3× longer—under identical hardware and batch settings.

Granted, we also noticed an interesting situation in our confusion matrix. The smooth muscle (MUS) class was commonly misidentified as cancer-associated stroma (STR) across both Virchow2 and Univ2 models and among both Attention Pooling and MLP. We can see that roughly 167 MUS were consistently misclassified as STR. However, this did not apply equally when classifying STR, which models performed unexpectedly well at. On average, only about 2 to 18 STR were misclassified as MUS, an abnormal situation that we cannot explain currently.

Table 4. Average Training Time for Attention Pooling and MLP (values may vary due to differences in external factors, e.g. computational power)

Method	Average Training Time (minutes/seconds)
Attention Pooling	15 min 05 sec
MLP	5 min 50 sec

4.3.1. Errors and Limitations

Just like any other studies, our work features many possible sources of error and limitation awaiting to be solved. Noticeably, we only evaluated based on a single dataset NCT-CRC-HE-100k, therefore our results may not be transferable to other scenarios with different organs, settings, whole slid images, and many more. This was mainly due to the fact that open-source dataset are rare in reality, and we simply could not find any other useable datasets. Looking foward, this situation can be mitigated with more open data exchange. Not to say this current dataset is perfect though, limitations also exist within this dataset as well, class imbalance for example is a distinctive trait we have observed. Through the confusion matrix we can identify two major classes, Adipose (ADI) and colorectal adenocarcinoma epithelium (TUM) that indicate a clear imbalance, even going four times more abundant than some of the minor classes (e.g. debris (DEB)) .

Furthermore, these results may introduce bias toward models whose pretraining aligns well with CRC histology. We also need to highlight that our training details and hyperparameter exploration are rather rough. The learning rates, dropout, and attention hidden sizes were keeping constant throughout all trails, so our results may not represent the best capabilities of each method. In addition, we did not strictly validate many possible factors that may influence results such as potential data leakage or preprocessing mismatches. As we showed, our training

curves show pervasive overfitting and very early optimal checkpoints, our metrics may be sensitive to validation noise without systematic early-stopping or cross-validation. An unavoidable limitation lies in the clinical translatability of our work; we did not assess model calibration, the cost of false positives/negatives, and our reported accuracy gains may not translate to safer clinical decisions. Finally, our study lacked statistical confidence measures like confidence intervals and did not include a structured interpretability and failure analysis, which hampers actionable insights for improving model safety and robustness.

4.3.2. Future Work

In the future, we aim to use a multi-head attention to extract deeper features from the patch tokens. This multi-head attention mechanism would take our work a step further, capturing more complex features from images. This would significantly improve model performance, with this new approach the model's accuracy, robustness, and ability to generalize to new, unseen data is expected to increase. Another goal we have is to map out the attention maps based on the attention weights, which can highlight which regions of the patch the model found is most influential towards its decision. This can increase the ethical reliability of histology models, because pathologists will be able to manually verify the thought process of the machine.

5. Conclusion

This study focused on some simple yet representative classifier heads—attention pooling over patch tokens, MLP over CLS tokens, and KNN—on embeddings from eight pathology foundation models using the NCT-CRC-HE-100K dataset. Across classification models, attention pooling consistently delivered the highest F1 scores, highlighting the value of supervised, task-specific reweighting of spatial evidence in histology. MLP heads on CLS tokens offered a strong, lightweight baseline with competitive performance when CLS representations were well-formed (e.g., Virchow2, UNI2), while KNN provided an efficient, parameter-free reference but lagged behind MLP and attention pooling. In conclusion, we would determine Virchow 2 with attention pooling as the state-of-the-art pipeline for patch level tissue classification.

In our results we observe that training dynamics revealed rapid early convergence and pervasive overfitting, underscoring the need for aggressive early stopping, better regularization, and validation. Practically, the principal trade-off observed was between accuracy and resource cost: attention pooling’s gains rely on patch-token storage and longer training, whereas CLS-only MLPs minimize I/O, disk, and wall-clock time.

Future work should focus on (1) deepen attention heads with multi-head pooling and gating, coupled with stronger regularization and calibration; (2) standardize preprocessing and conduct more rigorous methodology. These steps will move the benchmark toward more meaningful, reliable, and deployable histology AI systems.

6. Bibliography

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). *Emerging Properties in Self-Supervised Vision Transformers* (No. arXiv:2104.14294). arXiv. https://doi.org/10.48550/arXiv.2104.14294

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* (No. arXiv:2010.11929). arXiv. https://doi.org/10.48550/arXiv.2010.11929

DykeF/NCTCRCHE100K · Datasets at Hugging Face. (n.d.). Retrieved July 16, 2025, from https://huggingface.co/datasets/DykeF/NCTCRCHE100K

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, *18*(7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, *25*. https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

Macenko, M., Niethammer, M., Marron, J. S., Borland, D., Woosley, J. T., Xiaojun Guan, Schmitt, C., & Thomas, N. E. (2009). A method for normalizing histology slides for quantitative analysis. *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 1107–1110. https://doi.org/10.1109/isbi.2009.5193250

Merritt, R. (2025, February 11). What Are Foundation Models? *NVIDIA Blog*. https://blogs.nvidia.com/blog/what-are-foundation-models/

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2024). *GPT-4 Technical Report* (No. arXiv:2303.08774). arXiv. https://doi.org/10.48550/arXiv.2303.08774

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., … Bojanowski, P. (2024). *DINOv2: Learning Robust Visual Features without Supervision* (No. arXiv:2304.07193). arXiv. https://doi.org/10.48550/arXiv.2304.07193

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (No. arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., & Kong, T. (2022). *iBOT: Image BERT Pre-Training with Online Tokenizer* (No. arXiv:2111.07832). arXiv. https://doi.org/10.48550/arXiv.2111.07832

7. Appendix

Project Repository

- Title: foundation-model-histology
- URL: https://github.com/Cheezecats/foundation-model-histology.git
- Description: Benchmarks multiple classifier architectures—attention pooling, K-Nearest Neighbors (KNN), and a linear Multi-Layer Perceptron (MLP)—on consistent histology patch embeddings from eight foundation models using the NCT-CRC-HE-100K dataset.
- Contents:
    - Model archives: Virchow, Virchow2, UNI, UNI v2, HibouB, HibouL, Phikon, Phikonv2 (zip files)

Environment and Dependencies

- OS: Windows 11
- Python: 3.10.18
- Key libraries:
    - PyTorch, torchvision, timm, transformers
    - numpy, scikit-learn, h5py
    - matplotlib/seaborn (plots), tensorboard
- Hardware:
    - GPU: RTX 4080 (16 VRAM)

8. Acknowledgements

I would like to express my deepest gratitude to my guidance teacher, Mr. Shen, for his invaluable guidance, insightful feedback, and patience throughout the research and writing process. His expertise was instrumental in shaping this essay.

My sincere thanks also go to all of those who have supported me along the way including my family for their constant support and encouragement, which sustained me through this challenging endeavor.

8.1. Research Background

Personally, my interest in computer science originated during middle school with computer hardware, nowadays I have developed my interest with a specific focus on large language models and vision models; independently, I studied the core principles and recent advancements in this field. When I decided to participate in the丘成桐中学科学奖, I discussed my interest with my computer science teacher in 9th grade, Mr. Shen. Building on my self-directed learning, Mr. Shen broadly suggested I explore the application of AI models within specific domains and proposed possible directions of experiments.

After careful consideration of the current world and domains of research, I ultimately choose the field of digital pathology because of its unlimited expanding boundaries and its implication in real world medication, a rapidly evolving field that leverages artificial intelligence (AI) to derive quantitative insights from whole slide tissue images. Applications such as tumor detection, nucleus characterization, and survival prediction through tissue feature extraction have become increasingly common across multiple cancer types. Much of the progress in this field has been driven by deep learning models, traditionally relying on convolutional neural networks (CNNs) such as those with ResNet-50 backbones. However, traditional segmentation models have demonstrated limited generalizability. Training these networks is both time-consuming and costly, creating a substantial barrier for widespread adoption outside labs.

Recognizing these challenges, I developed the idea of examining how transformer-based models—specifically vision transformers (ViT)—are adapted to histopathology patch classification. While I am not responsible for training the foundation models themselves, I have selected and adapted publicly available vision transformer models for my experiment.

For this project, I utilized a publicly available, open-source patient dataset hosted on HuggingFace, which does not require authentication for access, namely the NCTCRCHE100K dataset.

8.2. Challenges Encountered and the Solutions Developed

The topic selection process is first difficulties I encountered. Not only did I needed to consider the realistic capabilities of a topic, but the topic also had to be innovative enough to be on the frontlines of current research. This was an incredibly challenging balance to find, because anything extremely expertise was way out of reach of any average high school student like me. Finally, I reached my current topic after careful consideration, weighing crucial factors such as time, resources, and knowledge requirements. Before even starting my research I had already roughly sketched out a plan for my entire work flow over the summer and confirmed my choice.

The main challenges I faced during my experiment process was probably fixing nuanced bugs and errors within my code that caused inconsistency and generated false results that did not seem reasonable. Moreover, this process of debugging was extremely time costing and fast solutions were not always plausible. In result, time was often sacrificed for better results and on a scale greatly extended the workflow.

The final essay construction also came to me as extremely difficult. Although this was not the first time I wrote a full-length research essay, the specific deadlines and high requirements brought heavy burdens. This becomes especially evident as the new school year started, and I struggled to find a balance between schoolwork and external competitions. In addition, the level of hours I would put into this project is enormous and became frustrating sometimes. The only feasible course of action was to adopt a stance of resilience and see the process through to its conclusion.

8.3. The mentoring relationship between the supervisor and the contest participant.

Mr. Shen was my 9th grade computer science teacher with his Bachelor's and Master's degrees in Computer Science from Columbia University; in addition, Mr. Shen is also my high school's external competition coordinator, providing completely free mentoring throughout the entire competition preparation duration. His advice mainly concentrated on essay structure and content, but as mentioned before also included directive advice on topic and methodology.