Metastatic Tissue Classification

Vidhi Bhatt * vidhib@g.ucla.edu
Debapriya Tula * dtula@g.ucla.edu
Sujit Silas * sujitsilas@g.ucla.edu

*** Affiliation: University of California, Los Angeles (UCLA)**

Abstract

Histopathologists rely on microscopic imaging of lymph nodes to identify metastatic cancer, a process that is often subjective and varies between experts. This project aims to enhance the objectivity and accuracy of metastatic tissue classification using Machine Learning (ML) and Deep Learning (DL) techniques on the PatchCamelyon dataset, a publicly available collection of histopathologic scans. We explore traditional ML models such as k-Nearest Neighbors (KNN) and Support Vector Machines (SVM), alongside advanced DL architectures like ResNets, InceptionNet, EfficientNet, and U-Net. Additionally, we investigate post-CNN architectures, including Vision Transformers (ViTs), to assess their potential for improved classification performance. Given the importance of model interpretability in medical applications, we also employ Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize the regions of an image that contribute most to the model's predictions, enhancing explainability. By leveraging AI for cancer detection, our goal is to contribute to the development of objective, automated diagnostic tools that can aid pathologists in identifying metastatic cells more reliably and efficiently.

1 Introduction

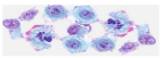
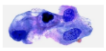
Cancer diagnosis heavily relies on the accurate examination of tissue samples, particularly through microscopic biopsy image analysis. Histopathologists assess cell structures to determine whether a tissue sample contains malignant cells. Key distinguishing factors include differences in cell nuclei color, shape, size, and their proportion to the cytoplasm Komura and Ishikawa [2018]. These morphological variations provide critical insights into disease progression, especially in metastatic cancer, where early detection plays a vital role in patient survival.

Among the various indicators of cancer, lymph nodes serve as essential markers. These small, bean-shaped structures are part of the immune system and often become swollen in response to infections or malignancies. The presence of metastatic cells in lymph nodes is a significant factor in staging cancer and determining treatment strategies Clinic [2023]. However, analyzing histopathologic images to detect cancerous cells is a complex and subjective process, with diagnoses varying between experts due to differences in training, experience, and interpretation. This variability highlights the need for automated, objective methods to assist pathologists in cancer detection.

Artificial intelligence (AI) and machine learning (ML) have emerged as powerful tools for medical image analysis, offering the potential to enhance diagnostic accuracy and reduce human bias. Recent advances in deep learning, particularly convolutional neural networks (CNNs), have demonstrated success in image classification tasks, including cancer detection. In this project, we aim to explore both traditional ML approaches, such as k-Nearest Neighbors (KNN) and Support Vector Machines (SVM), and deep learning architectures, including ResNets, InceptionNet, EfficientNet, and U-Net.

Code link:
https://github.com/VidhiBhatt01/Metastatic_Tissue_Classification-ECE_247_Final_Project_W25

Table 1: Differences between Normal and Cancerous Cells

Normal Cells	Cancerous Cells	Description of Cancerous Cells
		large and variably shaped nuclei
		many dividing cells and disorganized arrangements
		variation in size and shape of nuclei
		loss of normal feature (shape and morphology)

Additionally, we will investigate the performance of Vision Transformers (ViTs) as a post-CNN architecture to determine their effectiveness in classifying histopathologic images. To ensure model transparency, we also plan to apply Gradient-weighted Class Activation Mapping (Grad-CAM) to provide visual explanations of predictions.

By leveraging these AI-driven approaches, this project seeks to improve the objectivity and reliability of metastatic tissue classification. The integration of deep learning techniques with explainability methods could pave the way for more consistent and interpretable cancer diagnoses, ultimately supporting pathologists in clinical decision-making.

2 Dataset

For this study, we utilized the **PatchCamelyon (PCam) dataset** Veeling et al. [2018], a publicly available image classification dataset designed for metastatic tissue detection. The dataset consists of 327,680 color images, each measuring 96×96 pixels, which were extracted from histopathologic scans of lymph node sections. It is derived from the Camelyon16 Challenge dataset, which focuses on detecting lymph node metastases in whole-slide images.

2.1 Dataset Composition

The dataset serves as an ideal benchmark for evaluating various machine learning (ML) and deep learning (DL) models due to its well-balanced nature and large sample size. Given its widespread adoption in computational pathology research, PCam enables a standardized comparison of different classification algorithms.

- **Total Images:** 327,680
- **Positive Class (+) - Metastatic Tissue:** 163,862 images
- **Negative Class (-) - Normal Tissue:** 163,818 images
- **Class Balance:** Approximately 50-50 split between normal and metastatic tissue
- **Data Cleaning:** Four blank images (all-white) were identified in the negative class and removed to ensure only meaningful samples were analyzed.

2.2 Data Pre-Processing

To enhance image consistency across the dataset, we applied the Macenko normalization technique, a widely used stain normalization method in histopathology image processingMacenko et al. [2009]. Macenko normalization adjusts stain intensity distributions by decomposing an image into its stain components using singular value decomposition (SVD) and aligning them with a reference image. This ensures that color variations introduced by different staining protocols do not affect model performance. The normalization was implemented using the torchstain library, where we fitted a reference image and applied the transformation to all images.

In addition to Macenko normalization, we explored other key preprocessing techniques like Contrast Limited Adaptive Histogram Equalization (CLAHE) Pizer et al. [1990], Reinhard Normalization

Reinhard et al. [2001], and other image transformations (present in code repository). Images were resized to 224×224 pixels (299×299 pixels for Inception) to maintain consistency with pre-trained models. A sequence of transformations, including conversion to tensor format and normalization with ImageNet mean and standard deviation values, was applied to standardize pixel distributions and stabilize training. These preprocessing steps help mitigate domain shifts and improve model generalization.

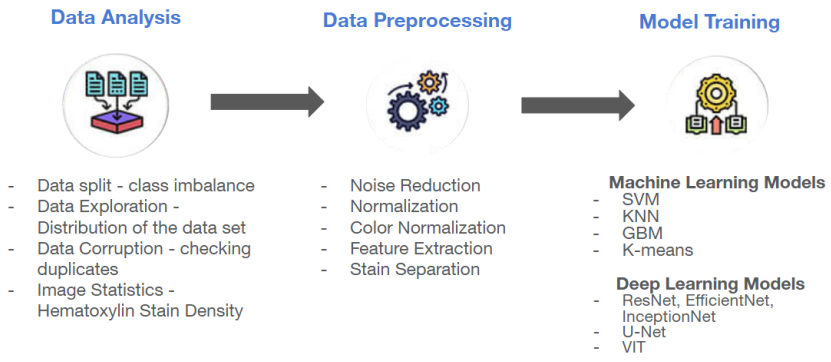


Figure 1: Procedural workflow

3 Methods

3.1 Machine Learning Models

Due to limited computational resources, we trained our machine learning models on a subset of the training data. Since the training dataset has an exact 50-50 split between normal and metastatic tissue, we randomly selected indices from the original dataset using a fixed random seed to ensure reproducibility.

We experimented with three supervised machine learning models: a gradient boosting classifier (a boosting machine learning algorithm which tries to reduce the errors by weighing samples based on their gradients), a k-nearest neighbors classifier, and a support vector machine classifier. Additionally, we tested each preprocessing method listed in the table above individually with the models and explored various combinations of preprocessing techniques.

In addition, we also performed mini-batch K-means clustering on the data, which given that it is an unsupervised algorithm, we cannot compare to the supervised algorithms, and instead compared Silhouette and K-means scores between different numbers of clusters.

3.2 Deep Learning Models

We primarily explored **5** vision architectures for the classification problem. For each of them, we train their base versions (with standard preprocessing) to start with. We also train them with each of the preprocessing techniques discussed above which have been found to be visually discernible.

- **ResNet 50** - The first architecture we explored was the ResNet-50 He et al. [2015], a 25.6 M parameter model which was introduced by Microsoft in 2015. It was the first CNN which allowed having a deep network with a large number of layers by introducing residual connections preventing the vanishing of gradients. For our experiments, we choose the original Resnet50 with ImageNet pre-trained weights. For training the model, we use a batch size of 128, and train it for 20 epochs with an AdamW(learning_rate=1e-3) and $L2$ regularization with $\lambda = 0.1$.
- **InceptionV3** - Using the same hyperparameter settings, we explored InceptionV3 Szegedy et al. [2016], a 23.9 M parameter model by Google (2016), which uses factorized convolutions and asymmetric kernels for efficiency while maintaining accuracy. A key innovation in this architecture is the use of auxiliary classifiers to mitigate the vanishing gradient problem

in deep networks by adding intermediate supervision. For our experiments, we use the InceptionV3 model with ImageNet pre-trained weights.

- **EfficientNet** - Next, we explored EfficientNet-B0 Tan and Le [2019], a 5.3 M parameter model that follows the compound scaling principle, where depth, width, and resolution are scaled simultaneously to maximize accuracy while maintaining efficiency. EfficientNet models utilize mobile inverted bottleneck convolution (MBConv) blocks and squeeze-and-excitation mechanisms to improve feature representation. The base version of EfficientNet-B0 is trained with ImageNet pre-trained weights, using a batch size of 128, an AdamW optimizer (learning rate = 1e-3, weight decay = 0.1), and for 20 epochs.

- **UNet** - UNet Ronneberger et al. [2015] is a 34.6 M parameter model popularly used for segmentation tasks. It consists of a contracting path which downsamples the image, and an expanding path which upsamples the downsampled features. Skip connections are introduced here as well to cater for vanishing gradients. We modify the UNet for our task of classification by adding a global average pooling (GAP) layer and classification head.

- **Vision Transformer** - To understand how transformer based models would perform for this dataset, we trained a Vision Transformer (VIT-Base), an 86 M parameter model with our dataset. Vision transformer Dosovitskiy et al. [2020] follows the same attention mechanism as the language based transformer model. Tokens in case of images are however formed by cropping the image into patches (eg 16x16) and passing them through an embedding layer. We use the extra CLS token affixed to the tokens for classification. The model is trained for 10 epochs with a learning rate of 1e-4 and weight_decay of 0.01.

4 Explainability

While our deep learning models may perform very well on the dataset, their lack of decomposability into individually intuitive components makes them hard to interpret. This necessitates the use of mechanisms to associate the decision of a model's output for a given input to the representations it builds out of it. For doing so, we use a popular method for interpretability of CNNs, called GRAD-CAM (Gradient-weighted Class Activation Mapping) Selvaraju et al. [2016]. It leverages the gradients of a target concept (e.g., 'dog' in a classification network or a word sequence in a captioning model) to generate a coarse localization map. This map highlights the key image regions that influence the model's prediction. For our models, we show how for a given tissue image, the salient regions in the image are highlighted in the GRAD-CAM maps.

We also extend GRAD-CAM for our VIT-Base model by reshaping the flattened outputs of the encoder layers into an image of size $(14, 14)$, as the $(224, 224)$ image is encoded into $(14, 14)$ patches each of size $(16, 16)$.

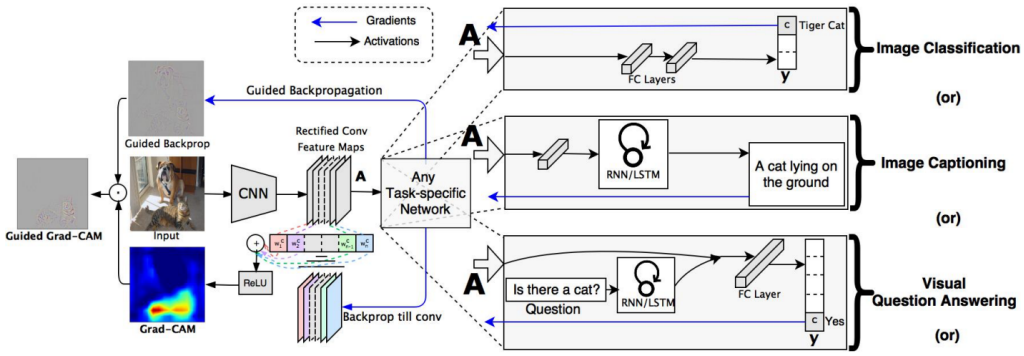


Figure 2: GradCAM

5 Results

5.1 Machine Learning Models

Experimentation was conducted on the K-nearest neighbors, support vector machine, and mini-batch K-means clustering algorithms in order to determine optimal parameters for the pre-processed data 3. For the gradient boosting classifier, we used RandomizedSearchCV from scikit-learn to find the more-optimal parameters. Then, these parameters were used to fit these algorithms to both pre-processed and raw validation data 4. The conclusion from these algorithms was that the gradient boosting classifier performed the best given the pre-processing steps, but was not run on the raw validation images, which makes it difficult to assess the benefits of the pre-processing steps for this particular model. K-means clustering was performed only on the training split of the data, but with all the corresponding pre-processing steps. Finally, it was seen that both the K-neighbors classifier and support vector machine classifier performed better with the raw validation images, suggesting that our subjective decisions of pre-processing techniques to apply to the images resulted in potential information loss and decrease in performances.

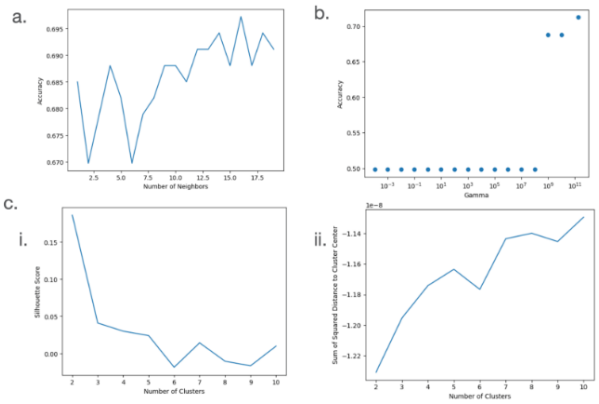


Figure 3: Experimentation on machine learning model parameters for preprocessed data: (a) K-nearest neighbors: Accuracy given number of neighbors. (b) Support vector machine: Accuracy given Gamma parameter value (influence of training sample's proximity). (c) Mini-batch K-means clustering optimization: (c.I) Silhouette score and (c.II) sum of distances to cluster centers given number of clusters.

Model	Variant	Validation Accuracy
GradientBoostingClassifier(subsample= 0.8, n_estimators=300, min_samples_split=2, min_samples_leaf=4, max_depth=7, learning_rate=0.1**)**	Reinhard, CLAHE, Pixel Normalization	81%
KNeighborsClassifier(n_neighbors=16**)**	CLAHE, Macenko, Grayscale, Opened	70% (same as raw)
MiniBatchKMeans(n_clusters = 2**)**	CLAHE, Macenko, Grayscale, Opened	N/A, but 2 clusters had best performance
svm.SVC(gamma='scale'**)**	CLAHE, Macenko, Grayscale, Opened	70% (raw 73%)

Figure 4: Machine learning model results

5.2 Deep Learning Models

To benchmark the performance of deep learning architectures on the dataset, we extensively experiment with the 5 models discussed above with various preprocessing schemes. Throughout our experiments, we use a Binary Cross-Entropy Loss with an AdamW or Adam optimizer to optimize our model weights. A batch size of 128 is used for all the experiments.

Our CNN models, viz, ResNet50, EfficientNet, InceptionNetV3 as well as UNet are trained with a learning rate of 1e-3 for 20 epochs. The results are compared with Veeling et al. [2018] which introduces a DenseNet architecture for the problem. For Resnet50, we observe that the base model performs the best with a test accuracy of 88.7. Other preprocessing techniques do not seem to have a major impact on the performance. We then explore the UNet with some of the preprocessing steps which resulted in better test accuracy over the others, and observe that the Reinhard normalization performs slightly better than the other methods with a test accuracy of 90.6. In addition to these, we explore two other CNNs, viz InceptionNetV3 and EfficientNet. But due to shortage of computational resources, we perform only one set of experiments over it with the Macenko Macenko et al. [2009] normalization. We used a lower learning rate=$1e-4$ for training these two models.

The post CNN architecture, Vision Transformer, performed the best among all the models. We trained it for 10 epochs with a learning rate of $1e-4$ and $\lambda = 0.01$ for the $L2$ regularization. We observe that the Macenko normalization provided the best accuracy among the other preprocessing techniques tried out.

The experiments were conducted on Nvidia V100 GPUs available on the Hoffman2 cluster at UCLA. The performance of these models was evaluated using standard classification metrics such as accuracy. The results are summarized in the following section.

Table 2: Deep Learning Model Results - ResNet 50

Model	**Variant**	**Train Acc (%)**	**Val Acc (%)**	**Test Acc (%)**
Baseline	PCam-DenseNet	-	-	89.8
ResNet 50	Base	91.74 ± 0.38	89.83 ± 0.78	$\mathbf{88.69 \pm 0.25}$
	RGB2HED	64.56 ± 0.4	65.66 ± 0.22	63.31 ± 0.52
	Wavelet Transform	91.36 ± 1.0	86.00 ± 1.5	86.53 ± 0.45
	CLAHE	92.51 ± 1.3	89.82 ± 0.8	88.49 ± 0.35
	Reinhard	91.46 ± 0.8	89.36 ± 0.22	88.20 ± 1.2
	Opening	90.53 ± 0.92	86.84 ± 0.25	86.82 ± 0.5
	Macenko	92.14 ± 0.86	89.75 ± 1.23	87.97 ± 0.7

Table 3: Deep Learning Model Results - UNet

Model	**Variant**	**Train Acc (%)**	**Val Acc (%)**	**Test Acc (%)**
Baseline	PCam-DenseNet	-	-	89.8
UNet	Base	95.29 ± 1.3	90.95 ± 0.8	90.48 ± 0.5
	CLAHE	94.58 ± 1.2	90.55 ± 0.52	89.83 ± 0.3
	Reinhard	92.95 ± 1.7	90.08 ± 0.33	$\mathbf{90.61 \pm 0.42}$
	Macenko	95.71 ± 0.95	90.41 ± 0.34	89.91 ± 0.72

Table 4: Deep Learning Model Results - ViT

Model	**Variant**	**Train Acc (%)**	**Val Acc (%)**	**Test Acc (%)**
Baseline	PCam-DenseNet	-	-	89.8
ViT	Base	92.02 ± 2.23	90.47 ± 0.86	90.86 ± 0.70
	CLAHE	94.38 ± 3.2	90.28 ± 0.77	89.99 ± 0.90
	Reinhard	92.11 ± 2.7	90.70 ± 0.52	90.97 ± 0.9
	Macenko	92.00 ± 2.5	91.19 ± 0.85	$\mathbf{92.25 \pm 1.15}$

5.3 Explainability

We use a pytorch library called *pytorch-grad-cam* Gildenblat and contributors [2021] for tracing the model decisions back to the features obtained from the intermediate layers. In the figures, 5 and 6

Table 5: Deep Learning Model Results - CNN models + Macenko Normalization

Model	Train Acc (%)	Val Acc (%)	Test Acc (%)
EfficientNet + Macenko	99.80 ± 0.02	87.63 ± 0.33	86.14 ± 0.4
InceptionV3 + Macenko	99.77 ± 0.02	86.85 ± 0.19	87.32 ± 0.52

we put together the maps for each of the models for some images from the positive class (with the metastatic tissue) and the negative class (the ones without).

For the ResNet50 model, we observe the class activation maps with respect to the 4th residual block - $layer4$. We observe that specific regions in some of the images are activated for this model. For the positive, most of the activation is restricted to a small region within the map as seen in 5. But for negative images, we see large regions being activated in the maps.

We see a similar pattern of activation for the UNet maps as well, which we obtain with respect to the last transposed convolution layer. But for the InceptionV3 maps, we see maps being activated more for the positive class, showing an inverse activation. We obtain these activation with respect to the $Mixed\_5b$ layer.

EfficientNet shows activations for portions which have higher intensity in both the positive and negative class cases. This is probably because we observe the maps with respect to the last convolutional layer in the model, $\_conv\_head$.

For the VIT model, we observe clearer and more diversely activated maps for the negative class, as well as small clear activations in the maps for the positive class. We obtain the maps with respect to the 4th encoder layer, $encoder\_layer\_4$.

Overall, we observe that the maps are primarily activated for the negative class, which helps reinforce the correct predictions.

6 Discussion

Our experiments highlights the importance of both explainability and automation of metastatic tissue classification. Among the traditional ML models, the gradient boosting classifier showed the best results on the pre-processed subset of the PCam dataset. Notably, however, the preprocessing steps did not consistently benefit other models (e.g., k-Nearest Neighbors and Support Vector Machines), likely indicating that the transformations, while visually intuitive, may sometimes reduce useful color and texture information. In contrast, DL architectures consistently demonstrated higher overall performance. Models such as ResNet-50, U-Net, and InceptionV3 all surpassed classical ML methods, albeit with some variability based on staining normalization and other preprocessing procedures. U-Net, traditionally favored for segmentation tasks, adapted well to classification when augmented with global average pooling, highlighting the adaptability of segmentation-centric designs to other vision problems.

ViTs emerged as particularly robust. Their attention-based architecture allowed them to surpass most CNN models when equipped with Macenko normalization, achieving a test accuracy of approximately **92.25 %**. This suggests that harnessing global context through self-attention may be especially beneficial for complex histopathological images, where subtle morphological features can be crucial for accurate classification.

Explainability, pivotal in medical imaging, was approached via Grad-CAM. Visual inspections of Grad-CAM outputs confirmed that well-performing models (CNNs and ViTs) attend primarily to regions rich in discriminative tissue morphology. This not only validates the learned representations but also fosters greater clinician trust in AI-assisted diagnosis. Future work could expand on these interpretability efforts, for instance, by integrating layer-wise relevance propagation or other forms of explainable AI to further elucidate the decision-making process.

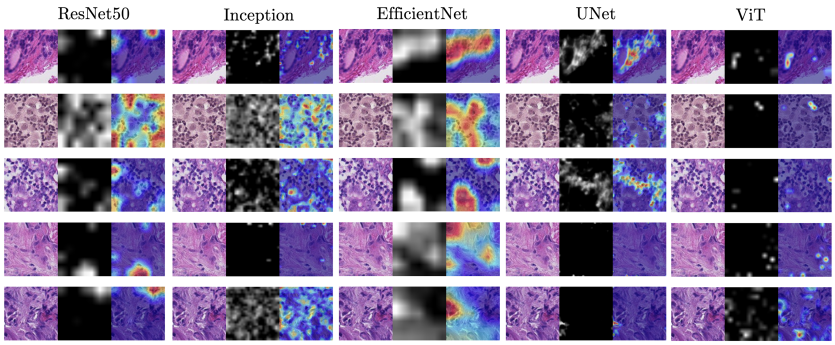


Figure 5: GradCAM Positive Class

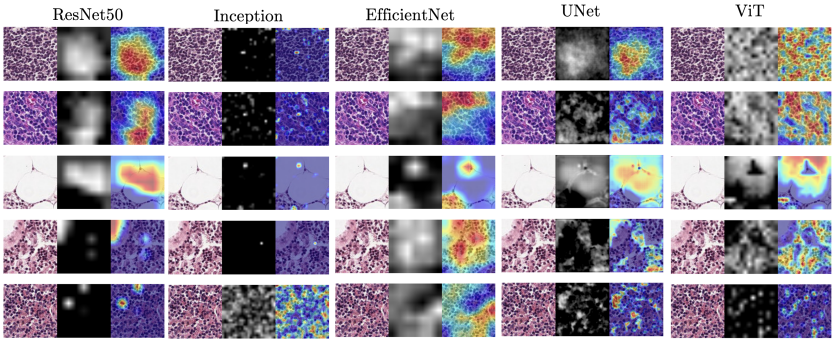


Figure 6: GradCAM Negative Class

7 Conclusion

Our study illustrates that deep learning architectures, particularly Vision Transformers and advanced CNNs, offer significant advantages over traditional ML methods for metastatic tissue classification on the PCam dataset. Stain normalization, while occasionally inconsistent in its effects, remains a valuable tool in mitigating color variations that arise from differing lab protocols. The best performing methods achieve competitive classification accuracy and, when coupled with explainability techniques like Grad-CAM, provide a foundation for more transparent clinical decision support. Looking ahead, further optimizations in hyperparameters, more extensive architecture search, and multi-scale image approaches may enhance classification performance. Additionally, incorporating specialized domain knowledge, such as explicit attention to tumor microenvironments, could yield further improvements in accuracy. Through continued development, these AI-driven systems hold the potential to expedite the decision-making process, and ultimately improve patient outcomes in oncology.

References

C. Clinic. Lymph nodes, 2023. URL `https://my.clevelandclinic.org/health/body/23131-lymph-nodes`.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL `https://arxiv.org/abs/2010.11929`.

J. Gildenblat and contributors. Pytorch library for cam methods. `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

D. Komura and S. Ishikawa. Machine learning methods for histopathological image analysis, 2018. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC4782618/.

M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, N. E. Thomas, and B. M. Beckmann. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1107–1110, 2009. doi: 10.1109/ISBI.2009.5193250.

S. Pizer, R. Johnston, J. Ericksen, B. Yankaskas, and K. Muller. Contrast-limited adaptive histogram equalization: speed and effectiveness. In *[1990] Proceedings of the First Conference on Visualization in Biomedical Computing*, pages 337–345, 1990. doi: 10.1109/VBC.1990.109340.

E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001. doi: 10.1109/38.946629.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016. URL http://arxiv.org/abs/1610.02391.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling. Rotation equivariant cnns for digital pathology. *CoRR*, abs/1806.03962, 2018. URL http://arxiv.org/abs/1806.03962.