Lung and Colon Histopathological Image Classification

Atharva Bhagwat (acb9244), Kostis Paschalakis (kp2405), Sayantan Mukhopadhyay (sm9752)

Link to the repository
Tandon School of Engineering, New York University

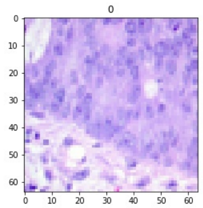
Overview

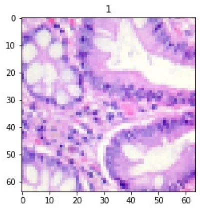
With the evolution of Artificial Intelligence and Machine Learning, we have seen how both of them have countless use cases in our lives. One of the most important areas where Machine Learning can bring the biggest impact and affect humanity as a whole is in Medicine. Again within medicine, there can be numerous use cases, however, one of the most significant ones is the early detection/ prevention of medical diseases. In order to see how ML can play a significant role in medicine and improve human lives overall we found the following data set(Borkowski et al.) which contains Lung and Colon Cancer Histopathological Images. Lung and colon cancer are causes of a significant number of deaths. Early detection will help medical professionals to act faster in treating the disease. Cancer detection using AI/ML is an interesting problem with a lot of ongoing research. In this paper, we intend to use different models such as CNN, ResNet, and Vision Transformers (ViT)(Dosovitskiy et al. 2020) in order to see how each performs in detecting cancer using the images from the data set. We also intend to compare each model to each other and determine the advantages and disadvantages of each.

Dataset

The data set consists of two types of histopathological images: colon and lung.
Classes in colon images:

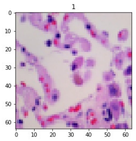
- Colon Benign Tissue
- Colon Adenocarcinoma

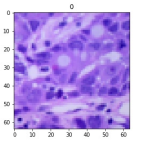


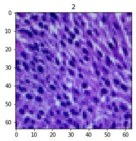


Classes in lung images:

- Lung Benign Tissue
- Lung Adenocarcinoma
- Lung Squamous Cell Carcinoma







Data pre-processing: Data set is divided into train-validation-test sets using 80%-10%-10%. Then the following transformations are applied on the training set:

- Random horizontal flip
- Random rotation of 20 degrees
- Resize image to 64*64
- Normalize image
- Convert to tensor

The following transformations are applied on the validation set and the test set:

- Resize image to 64*64
- Normalize image
- Convert to tensor

A larger image size can be used, but will require better computational systems.

Literature Survey

We referred to three main papers:

Lung and colon cancer classification using medical imaging: a feature engineering approach.

This(Hage Chehade A) paper manually applies pre-processing and feature extraction techniques like un-sharp masking, stain normalization, obtaining features from first-order statistis, gray level co-occurrence matrix for texture analysis, etc. then applying recursive feature elimination for feature selection, followed by training a model using XG-Boost, SVM, Random Forest, LDA, MLP, and LightGBM and comparing the results. The paper concludes with, XG-Boost model giving the best performance with an accuracy of 99% and a F1-score of 98.8%

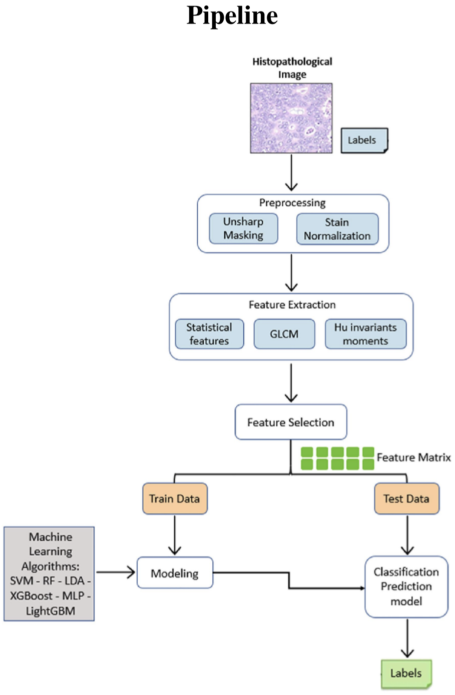


Figure 1: Pipeline for paper 1(Hage Chehade A)(fig 2)

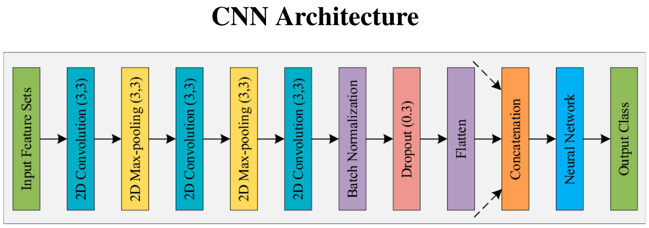


Figure 2: CNN architecture used in paper 2(Masud et al. 2021)(fig 7)

A Machine Learning Approach to Diagnosing Lung and Colon Cancer Using a Deep Learning-Based Classification Framework.

This(Masud et al. 2021) paper uses a CNN for classification. The paper applies the following pre-processing techniques, image sharpening using un-sharp masking, extraction of 2D fourier features, and extraction of 2D wavelet features. The paper concludes with, the model achieving 96.33% accuracy. While this papers uses a CNN, it uses a generated set of features instead of an image.

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

This (Dosovitskiy et al. 2020) paper introduces Vision Transformers. Transformers, used mainly for Natural Language Processing applications, can be used for image classification, over the traditional use of traditional Convolutional Neural Networks, which uses attention (Vaswani et al. 2017) mostly in conjugation or as small component replacements.

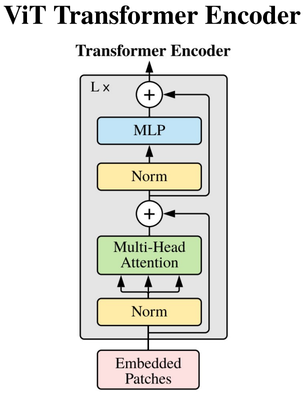


Figure 3: Structure of a ViT transformer encoder presented in paper 3(Dosovitskiy et al. 2020)

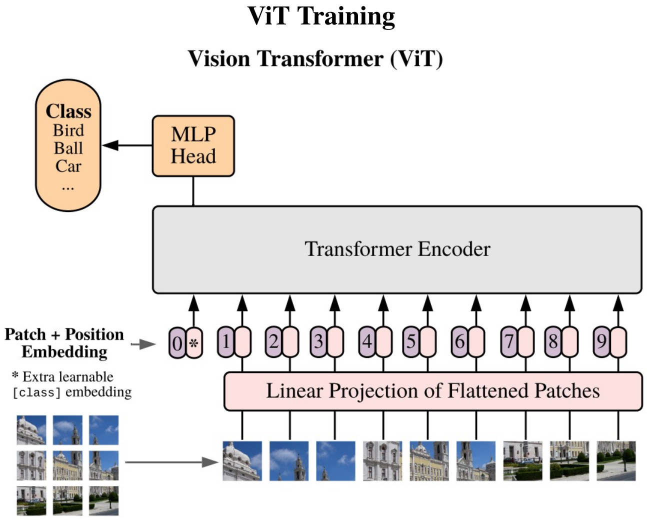
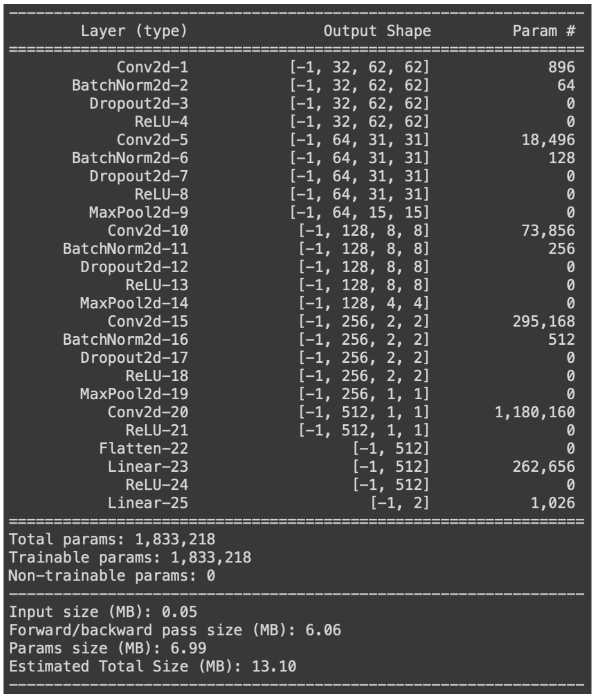

ViT Training
Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 4: ViT training pipeline(Dosovitskiy et al. 2020)

The authors have presented that Transformers can be applied directly on an image, by splitting the image fixed-size patches, embedding each of them linearly, adding position embeddings, and feeding the resulting vector to the transformer. ViTs can perform almost similar, if not better than state-of-the-art CNNs, at much-reduced expenses. The architecture of a ViT model is as follows (Bosech)-

- An image is split into patches of fixed sizes
- The patches are then flattened
- Lower-dimensional linear embeddings are created from these flattened image patches
- Positional embeddings are included
- The sequence is fed as an input to a transformer encoder
- The ViT model is pre-trained with image labels, which is then fully supervised on a big dataset
- The model is then Fine-tune downstream on a dataset for image classification

CNN Parameters



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1           [-1, 32, 62, 62]             896
       BatchNorm2d-2           [-1, 32, 62, 62]              64
         Dropout2d-3           [-1, 32, 62, 62]               0
              ReLU-4           [-1, 32, 62, 62]               0
            Conv2d-5           [-1, 64, 31, 31]          18,496
       BatchNorm2d-6           [-1, 64, 31, 31]             128
         Dropout2d-7           [-1, 64, 31, 31]               0
              ReLU-8           [-1, 64, 31, 31]               0
         MaxPool2d-9           [-1, 64, 15, 15]               0
           Conv2d-10            [-1, 128, 8, 8]          73,856
      BatchNorm2d-11            [-1, 128, 8, 8]             256
        Dropout2d-12            [-1, 128, 8, 8]               0
             ReLU-13            [-1, 128, 8, 8]               0
        MaxPool2d-14            [-1, 128, 4, 4]               0
           Conv2d-15            [-1, 256, 2, 2]         295,168
      BatchNorm2d-16            [-1, 256, 2, 2]             512
        Dropout2d-17            [-1, 256, 2, 2]               0
             ReLU-18            [-1, 256, 2, 2]               0
        MaxPool2d-19            [-1, 256, 1, 1]               0
           Conv2d-20            [-1, 512, 1, 1]       1,180,160
             ReLU-21            [-1, 512, 1, 1]               0
          Flatten-22                  [-1, 512]               0
           Linear-23                  [-1, 512]         262,656
             ReLU-24                  [-1, 512]               0
           Linear-25                    [-1, 2]           1,026
================================================================
Total params: 1,833,218
Trainable params: 1,833,218
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.05
Forward/backward pass size (MB): 6.06
Params size (MB): 6.99
Estimated Total Size (MB): 13.10
----------------------------------------------------------------
```

Figure 5: Number of parameters in CNN

Models

CNNs

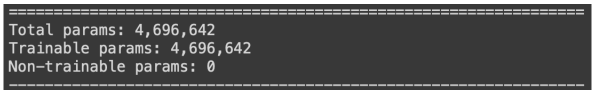
The CNN consists of a ‘conv block‘ and a ‘dense block‘. The ‘conv block‘ consists of 4 conv2D layers with 32, 64, 128, 256, and 512 output kernels with batch normalization and a dropout layer of with 20% probability, expect for the last conv2D layer. There is a MaxPool2D layer after the 2^{nd}, 3^{rd}, and 4^{th} conv2D layer. Each conv2D layer has a ReLU activation.

The ‘dense block‘ consists of 1 fully connected layer with 512 output units.

CNN architecture for colon data ends with a 2 unit fully connected layer. While, CNN for lung data ends with a 3 unit fully connected layer.

Fig 5 shows the number of trainable parameters.

ResNet Parameters



```
================================================================
Total params: 4,696,642
Trainable params: 4,696,642
Non-trainable params: 0
----------------------------------------------------------------
```

Figure 6: Number of parameters in ResNet

Training parameters for colon dataset:

- Batch size: 64
- Optimizer: Adam
- Learning rate: 0.001
- Loss function: Cross Entropy Loss
- Epochs: 10

Training parameters for lung dataset:

- Batch size: 64
- Optimizer: Adam
- Learning rate: 0.0001
- Loss function: Cross Entropy Loss
- Epochs: 20

ResNet

Training a large ResNet model, like ResNet-50, from scratch is time consuming and might not be the best fit for the size of our data. So, our intuition was that we should decrease the number of layers as this would significantly drop the number of parameters and work with data of our size. With a 3 layer network, intuitively we designed the number of blocks to decrease with network depth.

Fig 6 shows the number of trainable parameters.

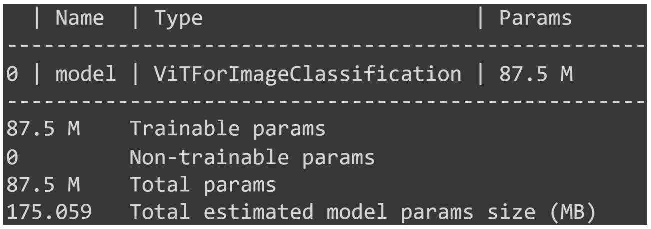
Training parameters for colon dataset:

- Batch size: 64
- Optimizer: Adam
- Learning rate: 0.001
- Loss function: Cross Entropy Loss
- Epochs: 5

Training parameters for lung dataset:

- Batch size: 64
- Optimizer: Adam
- Learning rate: 0.0001
- Loss function: Cross Entropy Loss
- Epochs: 10

ViT Parameters



```
  | Name  | Type                        | Params
-----------------------------------------------------
0 | model | ViTForImageClassification | 87.5 M
-----------------------------------------------------
87.5 M    Trainable params
0         Non-trainable params
87.5 M    Total params
175.059   Total estimated model params size (MB)
```

Figure 7: Number of parameters in ViT

Vision Transformers (ViTs)

The main advantage of ViTs is that it outperforms CNN-based models by nearly four times (Paul and Chen 2021). However, training transformers itself is resource intensive. High-performance transformer models are usually trained using multiple performance-quality GPUs in parallel for days or weeks. Such pre-trained transformers can be then fine-tuned on the basis of the dataset, when a classification layer is added that calculates the probability of a class, and provides an economic and environmentally sustainable way for general users with access high performance transformer models.

The transformer used for this experiment is the vit-base-patch32-384 by Google Research (Research 2020). It is a Vison Transformer, which like BERT (Devlin et al. 2018), uses multiple self-attention layers instead of CNN and RNN, which has proven to be a key element for vision networks to achieve higher robustness. Similar to BERT's token, a learnable classification token embedding is prepended to the sequence of image tokens' patch embeddings, whose state at the output of the transformer encoder serves as the image representation. The transformer encoder has Multi-Head Self Attention Layer (MSP), Multi-Layer Perceptrons (MLP), and Layer Norm (LN). This thereby helps improve the training time and overall performance. The classification head is implemented by the simple MLP layer. One hidden layer at pre-training and one linear layer at fine-tuning is used by ViTs for image classification.

The transformer vit-base-patch32-384 has been trained on ImageNet-21k (Deng et al. 2009), a large collection of images, at a resolution of 224x224 pixels, in a supervised fashion.The model was then fine-tuned by training on on ImageNet dataset, containing 1 million 384x384 images and 1,000 classes. The base model has 12 layers and 86 million parameters. Huggingface's transformer library provide functions to implement this model. This includes ViTFeatureExtractor and ViTForImageClassification (Huggingface). ViTForImageClassification is used use to instantiate the vit-base-patch32-384 model, while ViTFeatureExtractor is used to prepare the images.

Fig 7 shows the number of trainable parameters.

Training parameters for colon dataset:

- Batch size: 64
- Optimizer: Adam
- Learning rate: 0.00005
- Weight Decay: 0.0025
- Loss function: Cross Entropy Loss
- Epochs: 3

Architecture	Test Accuracy(%)	F1-Score
CNN	99.8	0.997
ResNet	98.5	0.981
ViT	99.87	0.998

Table 1: Results on colon data

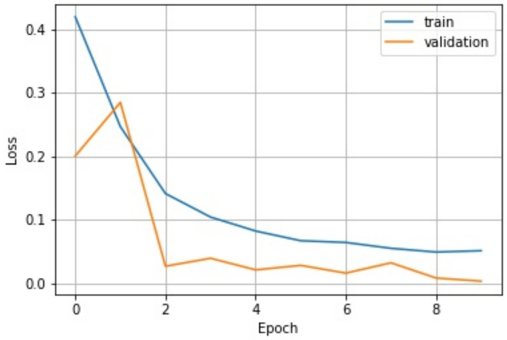


Figure 8: CNN Loss Plot(Colon)

Training parameters for lung dataset:

- Batch size: 64
- Optimizer: Adam
- Learning rate: 0.00005
- Weight Decay: 0.0025
- Loss function: Cross Entropy Loss
- Epochs: 3

Results

Colon Data

Refer table 1 for model comparison.

We observe that ViT performs slightly better than CNN and ResNet, with a F1-score of 0.998 and test accuracy of 99.87%.

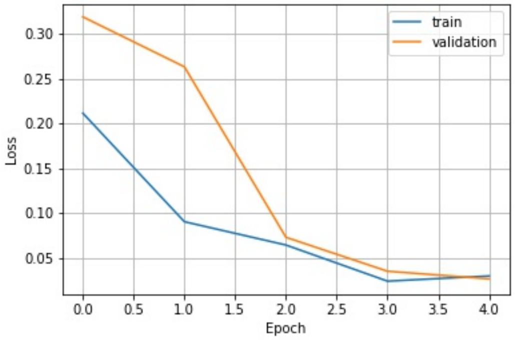


Figure 9: ResNet Loss Plot(Colon)

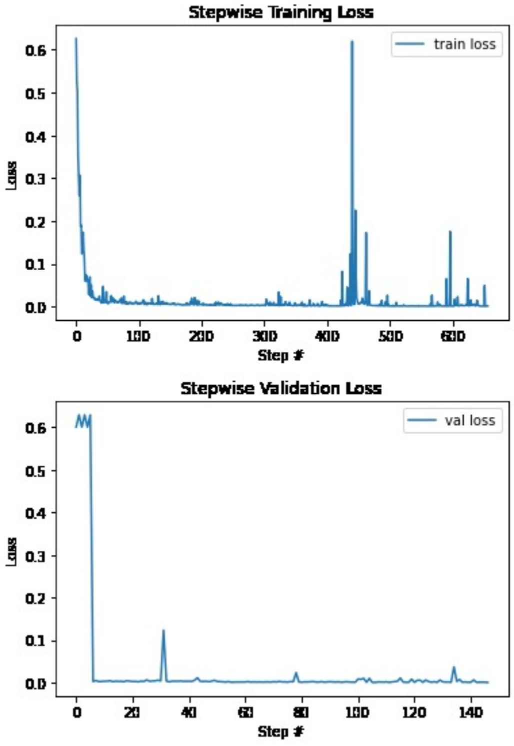


Figure 10: ViT Loss Plot(Colon)

Architecture	Test Accuracy(%)	F1-Score
CNN	98.8	0.987
ResNet	98.93	0.988
ViT	99.91	0.99

Table 2: Results on lung data

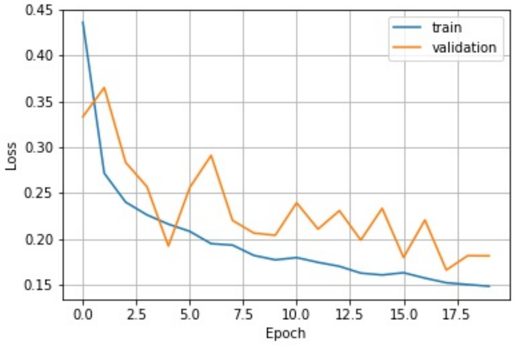


Figure 11: CNN Loss Plot(Lung)

Lung Data

Refer table 2 for model comparison.

We observe that ViT performs slightly better than CNN and ResNet, with a F1-score of 0.99 and test accuracy of 99.91%.

The ViT model not only performs better in all the metrics we have considered, it also takes less time to train as the majority chunk of training can be done upstream.

We notice, in case of colon data, the CNN has validation loss less than the training loss. This could be an indication of under-fitting. This can be because the colon data has two class labels. Still, the CNN has a high accuracy on the test set.

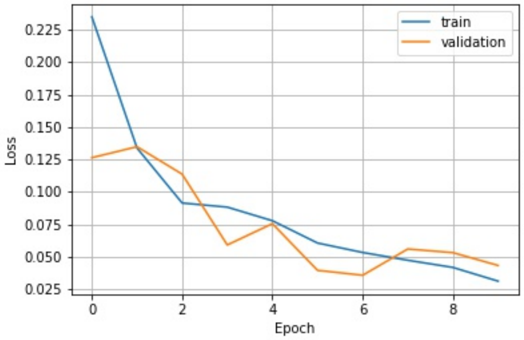


Figure 12: ResNet Loss Plot(Lung)

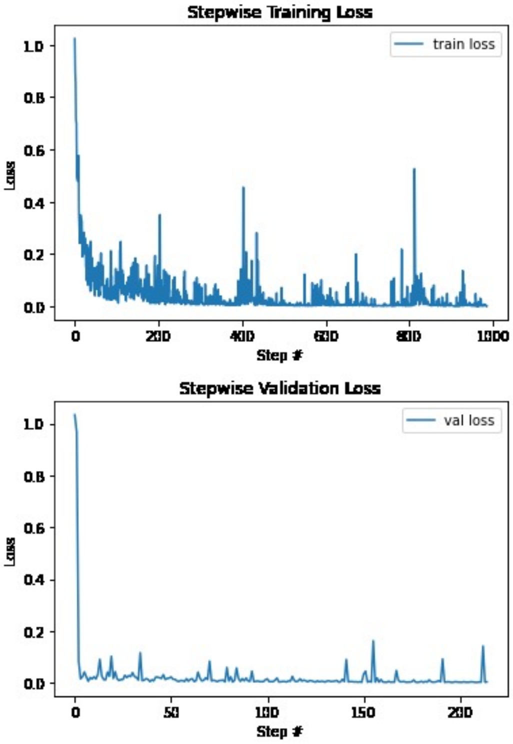


Figure 13: ViT Loss Plot(Lung)

Conclusion

With these deep learning architectures, we can create classifiers which perform at extremely high accuracies. We also observe that Vision Transformers can be considered a viable alternative to even state-of-the art CNNs and ResNet models for image classification. Various ViTs are available that have been pre-trained by large organizations like Google, Microsoft, Facebook, Apple, etc using large data sets, by leveraging their resources. Individual users can choose such a model that best suits their requirements and fine-tune it further as per their specific use-case, like done in this experiment for Histopathological Image Classification. The APIs from Huggingface allows to end-users to use such models seamlessly.

While immediate use of these kinds of systems might not be feasible, we can clearly see improvements in techniques and architectures. With a generalised data set and with improvements in interpretability of such systems, using these systems in a human-in-loop manner are the next steps.

References

Borkowski, A. A.; Bui, M. M.; Thomas, L. B.; Wilson, C. P.; DeLand, L. A.; and Mastorides, S. M. ???? LC25000 Lung and colon histopathological image dataset.

Bosech, G. ???? Vision Transformers (ViT) in Image Recognition.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

Hage Chehade A, M. J. O. M. C. P., Abdallah N. ???? Lung and colon cancer classification using medical imaging: a feature engineering approach.

Huggingface. ???? Vision Transformer (ViT).

Masud, M.; Sikder, N.; Nahid, A.-A.; Bairagi, A. K.; and AlZain, M. A. 2021. A Machine Learning Approach to Diagnosing Lung and Colon Cancer Using a Deep Learning-Based Classification Framework. *Sensors*, 21(3).

Paul, S.; and Chen, P.-Y. 2021. Vision Transformers are Robust Learners.

Research, G. 2020. Transformers for Image Recognition at Scale.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need.