Date of publication: Unpublished, date of current version: April 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Classification of Cervical Cancer Cells Through Visual Transformers

THOMAS FIELLO II[1], MARIAH MARIN[1], WAIRIMU MWANGI[1], MARCELO RUIZ LEON, SR.[1], AND ALEXIS SMITH.[2]

[1]Department of Mathematics, Embry-Riddle Aeronautical University, Daytona Beach, FL 32117 USA
[2]Department of Civil Engineering, Embry-Riddle Aeronautical University, Daytona Beach, FL 32117 USA

Corresponding author: Thomas Fiello II (e-mail: FielloT@my.ERAU.edu).

This work was supported in part by Dr. Prashant Shekhar and the MSDS Program of ERAU.

ABSTRACT The primary purpose of this project is to design and implement a custom visual transformer (ViT) for medical imaging applications. Since their debut to machine learning in 2017 with the paper "Attention is All You Need", Transformers have quickly become the dominant deep learning architecture in data science [1]. They saw initial success as unparalleled text generators and eventually were redeveloped to work with computer vision in 2021 with a new paper "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale" [2]. Now all of the top state-of-the-art models for the benchmark computer vision dataset ImageNet are either ViTs or Transformer-Hybrids [3]. The goal of this research is to utilize the developed visual transformer to classify images of cervical cells to classify and assist in diagnosing of cervical cancer. The data set consists of images that have been previously categorized into medically significant classifications. Using these provided classifications, the developed transformer will be able to sort these images into their respective categories. This project will have implications on the usability of machine learning in a medical environment, specifically in aiding diagnosis of cell screenings in the medical environment Specifically, the results of this project aim to provide an outlook on how transformers can be applied to revolutionize cell screening and imaging in the medical field with a specific concentration on cervical cancer cells.

INDEX TERMS Cervical cancer, computer vision, image classification, Herlev, machine Learning, SipakMed, Visual Transformers.

I. PROBLEM OVERVIEW

Cervical cancer is one of the most common cancers found in women. Despite being mostly treatable with fewer than 5000 women killed annually in the United States, the statistics are much different for the rest of the world where it kills over 300,000 women annually [4] [5]. This large disparity is primarily caused by a lack of effective screening hence leading to poor detection of the cancerous cells which reduces the survival rate for women who develop the disease. The current global digital and technological advances are invaluable as they are allowing humans to create solutions to real life problems that negatively impact quality of life. The development of advanced machine learning models has provided solutions with significant impact in multiple industries such as the medical field. Some of these solutions include providing healthcare workers with the ability to effectively perform critical tasks such as screening for cancerous cells in blood samples.

Developing a classification model that can accurately identify the different categories of cells within a pap smear sample could increase the accuracy of cervical cancer screening results exponentially. This would correspond to the increasing the number of women whose lives are saved from better detection methods. This project aims at being able to create and provide a model that achieves such results. To do so, current state-of-the-art model architecture for computer vision, the visual transformer will be utilized.

II. BACKGROUND AND LITERATURE

According to the World Health Organization (WHO), an estimated 570,000 women are diagnosed with cervical cancer around the world, and in 2018 alone, over half of the diagnosed women lost their lives to this disease. The vast majority of these cases occur in poor and developing countries due to the lack of screening and treatment in addition to a multitude of other conditions, one of which is long term infection with

human papillomavirus (HPV) which is one of the leading causes of cervical cancer.

Research has demonstrated that the most effective method to detect cervical malignancy is pap smears. Cytologists, scientists with an expertise in the study of cells, analyze cells collected from the squamocolumnar terminal of the cervix under a light microscope. While this method is very popular in today's age, it does pose challenges and constraints in it being time-consuming, expensive, and overall vulnerable to substantial errors due to the multitude of orientations a single slide could contain. Each slide of a pap smear contains millions of cells, which allows for a vast range of orientation and overlapping possibilities that cause difficulties in the accuracy of the current pap smear technique. These constraints prevent the idea of population-wise screening for cervical cancer, which in turn contributed to the large number of cases and deaths of women as a result of cervical cancer.

To address these challenges, modern technology has been used to create numerous Computer-Aided Diagnosis (CAD) systems which have become viable alternatives to detect cell malignancy. Researchers have been in the process of creating CAD systems using various techniques to achieve the goal of detecting cervical cancer with advanced speed, sensitivity, and accuracy. One of the overarching goals of the new CAD systems is to be able to handle the overlapping of cell images, as it is a very large concern and challenge in cervical cell malignancy detection.

A. RELATED WORK: ENSEMBLE-BASED MODELS

A journal article titled *A fuzzy rank-based ensemble of CNN models for classification of cervical cytology*, explains the research conducted in creating an ensemble-based classification model using three Convolutional Neural Network (CNN) architectures (Mana et al., 2021). Ensemble learning involves the fusion of decision scores from multiple classifiers to predict the final class label of an input sample. The proposed scheme uses a fuzzy rank-based fusion of classifiers by considering two non-linear functions on the decision scores generated by the base learners. This technique differs from simple fusion schemes that currently exist, because it considers the confidence the predictions of the base classifiers when making the final predictions of a cell's malignancy status. The method was pre-trained on ImageNet datasets, and then tested on two publicly available datasets known as the SIPakMeD dataset, which is one of the priority datasets for this research project, and the Mendeley Liquid Based Cytology (LBC) dataset. To evaluate the performance of this technique, metrics including precision, recall, accuracy and F1 score were used, which allowed for a better understanding and comparison of the model with other research-created models. Ultimately, the fuzzy rank-based ensemble produced a classification accuracy of 98.55% on the SIPakMeD dataset with a two-class setting, and a 95.43% accuracy with a five-class setting. When applied to the LBC dataset, the model achieved a classification accuracy of 99.23%. Future work and improvements this research article highlighted include the consideration of ensembles created from other base learners, the exploration of different rank generation functions on which to perform the ensemble, and the need to pre-process the pap smear cell images to further improve the model's performance.

Another team of researchers published a paper on a deep learning-based framework that was created to classify cervical cells using hybrid deep feature fusion techniques (HDFF) (Rahaman et al., 2021). The ensemble scheme proposed in this research, utilizes pre-trained deep learning models trained on ImageNet datasets. This research highlighted that deep learning provides poor performance on multi-class classification environments when there is an uneven distribution of data, which is prevalent in the cervical cell dataset. As a result, the proposed method uses a variety of deep learning models to capture more potential information and thus enhance the performance of classification. The model compares the performance itself with a deep learning-based method against a late fusion based method, using the same evaluation metrics as the research cited previously. Ultimately, the HDFF model obtained a classification accuracy of 99.85% on the SIPakMeD dataset for a two-class setting and a binary class accuracy of 98.32% on the Herlev dataset.

Research has been conducted solely on the SIPakMeD dataset, and researchers at the University of Ioannina, Greece, tested several classification schemes on the dataset in order to evaluate each performance on the discrimination of the different cell types (Plissiti et al., 2018). The schemes outlined in this paper include a Support Vector Machine (SVM) and a multi-layer perceptron (MLP) on the cell features, and a CNN and deep feature applications on the image features. The results of this particular research provide a reference point for the evaluation of alternative and improved techniques for cell image classification. The same evaluation metrics as mentioned in the previous research papers were used for this model, thus allowing for a more uniform method of comparing performance across the models. In addition to this model being used for the classification of cervical cell malignancy, it can also be used for the evaluation of image segmentation techniques or overlapping cells, which are the toughest challenges cervical cell classification continues to face. These research papers showcase only a portion of the variety of models that have been created to assist in the classification of cervical cells specifically. The majority use deep learning techniques as well as CNNs and some of the more effective and high performing models utilize a combination of features and techniques. These papers provide us with substantial background on what methods have already been attempted, the accuracy of certain features and techniques compared to others, and future work or improvements each research team conveyed that could potentially further improve the model's performance.

While conducting research on approaches taken on the same dataset used for this research project is crucial, it is also beneficial to take a more generic lens and conduct research on other classification methods used on different datasets and/or

different fields. A research paper delves into the detection for pap smear images using moving K-means clustering and a seed based growing algorithm (Isa, 2005). The objective of this research was to detect the edges of certain regions of interest in digital images. Isa proposed a combination of two techniques in order to detect the cytoplasm and nucleus edges of the cervical cells as a way of addressing the image quality and overlapping challenge this type of data has continuously experienced. A modified seed-based region growing (MS-BRG) algorithm is merged with a moving K-means clustering technique in order to automatically find threshold values, detect the edges of regions of interest, and ultimately apply the procedure to cervical cells. The achievements of this research involve the technique being more stable with respect to noise, and it being successful in differentiating both the cytoplasm and nucleus regions.

B. RELATED WORK - TRANSFORMERS

Transformers have self-attention based architectures, which has allowed them to become the standard algorithms for natural language processing tasks. Their applications in computer vision, however, remain limited which poses a challenge and an opportunity for research and investigation.

Transformers are a type of neural network model based solely on attention mechanisms. The model's architecture forgoes recurrence, a dominant characteristic in alternative neural networks, and relies instead entirely on attention mechanisms to draw global dependencies between the input and output (Vawani et al., 2017). The majority of transformer applications have been either in conjunction with CNNs, or utilized to replace certain components of CNNS while maintaining overall structure. As a result, researchers have developed variations of transformers to analyze the model's performance in a range of fields compared to the performance of a common CNN. A detailed diagram of a traditional transformer's architecture can be seen below:

One research team's application of a transformer delved into translation tasks between English-to-German and English-to-French, in order to compare the model's performance. Here, the transformer utilized the standard encoder-decoder structure, but with stacked self-attention and point-wise, fully connected layers (Vawani et al., 2017). Their variation in the algorithm was in the replacement of the recurrent layer of the standard architecture with multi-headed self-attention. Their research paper, *Attention is All You Need*, also emphasized the main motivators for the use of self-attention in the transformer which encompassed the total computational complexity per layer, the amount of computation that can be executed in parallel, and the path length between long-range dependencies in the network. Their research concluded that the transformer algorithm is a model that requires significantly less training time and is more parallelization, and is thus superior in quality compared to a CNN in this language translation environment.

The application of a transformer to a different field can be seen in the research paper *An Image is Worth 16x16*

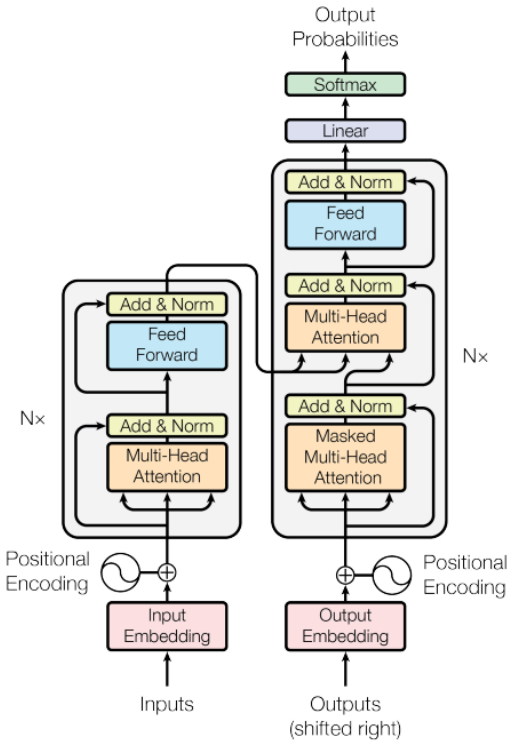

FIGURE 1: The left side of Figure 1 denotes the structure of the encoder, and the right half of the figure showcases the structure of the decoder for a transformer [1].

Words: Transformers for Image Recognition at Scale. In this paper, the direct application of a transformer alone to image patches sequences is showcased, along with its successful performance in image classification tasks. Dosovitskiy et al., utilized a Vision Transformer (ViT) for this application, as it required the analysis of images to conduct the classification.

The architecture of a ViT begins as the splitting of an image into patches and then flattening those patches. The flattened patches are used to produce lower-dimensional linear embeddings which are combined with positional embeddings and this sequence is feed as input to the transformer. The model is pre-trained with image labels from an extensive data set and the downstream data set is then fine-tuned for image classification. Based on the model's architecture, in order for a ViT to be successful, it needs to be trained on data sets with more than 14M images and the data needs to be converted into sequences for the input to be processed (Adaloglou, 2021). A diagram of a ViT's architecture can be seen in the figure below:

Thus, the ViT created by Dosovitskiy et al., was pre-trained on extensive amounts of data and then transferred to multiple image recognition benchmarks to ensure the transformer had sufficient training. The images were also split into patches and then provided as a sequence of linear embeddings as input to the transformer in order to avoid the introduction of image-specific inductive bias that prior

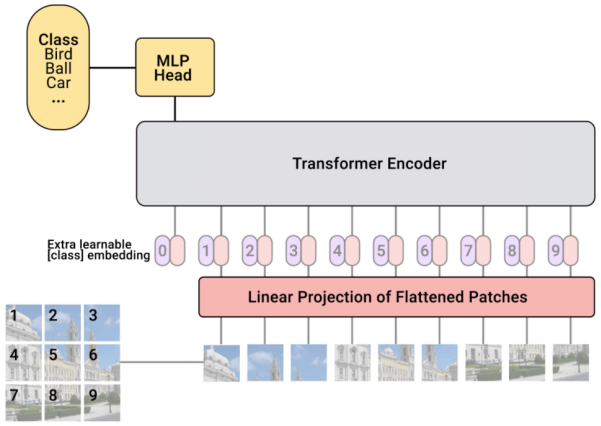

FIGURE 2: The diagram of a visual transformer's architecture. It treats an input image as a sequence of patches, similar to the natural language processing transformer [12].

models do. The results showcased that the ViT attained excellent results in the classification tasks compared to a CNN and it also required substantially less computational resources to train.

III. DATA SOURCES

A. SIPAKMED DATASET

The SIPaKMeD dataset is publicly available for scientific and experimental research. It is currently the largest dataset of labeled cervical cancer cell images. The dataset consists of 4,049 isolated cell images which were manually cropped from 966 cluster cell images taken from pap smear slides. The images were acquired using a Charge Couple Device (CCD) camera connected to a standard optical microscope. Cells can be considered normal, abnormal, or benign and from there can be further divided into 5 specific classes: Superficial-Intermediate, Parabasal, Koilocytotic, Dysketarotic, and Metaplastic [6] [7] [8]. Because this is the largest available dataset, it has either been the sole or just primary dataset used for similar academic papers.

1) The original research paper

Marina E. Plissiti, Panagiotis Dimitrakopoulos, Giorgos Sfikas, Christophoros Nikou, Olga Krikoni, Antonia Charchanti, SIPAKMED: A new dataset for feature and image based classification of normal and pathological cervical cells in Pap smear images, IEEE International Conference on Image Processing (ICIP) 2018, Athens, Greece, 7-10 October 2018.

2) Official database website

https://www.cs.uoi.gr/ marina/sipakmed.html

3) Kaggle page

https://www.kaggle.com/prahladmehandiratta/cervical-cancer-largest-dataset-sipakmed

B. HERLEV DATASET

The Technical Universtiy of Denmar (DTU)/Herlev pap smear dataset is actually a combination of two different datasets. An original from the 1990s and a newer one from 2005. They were collected by DR. MD. Beth Bjerregaard at the Herlev University Hospital in Denmark. The total dataset size is 917 single cell images collected using a digital camera microscope. These cells are all annotated into one of seven different classes: Superficial squamous epithelia, Intermediate squamous epithelia, Columnar epithelial, Mild squamous non-keratinizing dysplasia, Moderate squamous non-keratinizing dysplasia, Severe squamous non-keratinizing dysplasia, and Squamous cell carcinoma [9] [10].

1) Official database website

http://mde-lab.aegean.gr/index.php/downloads

2) Current best model paper

Bhatt AR, Ganatra A, Kotecha K. Cervical cancer detection in pap smear whole slide images using convNet with transfer learning and progressive resizing. PeerJ Comput Sci. 2021;7:e348. Published 2021 Feb 18. doi:10.7717/peerj-cs.348

IV. PROPOSED APPROACH

Visual transformers have shown the current highest accuracy on image recognition and classification problems thus earning them the title of state-of-the-art algorithms from benchmark databases such as ImageNet [3]. The proposed approach for this project is the development of a custom visual transformer trained on the two largest cervical cancer datasets aforementioned.

The intended outcome for this approach is that the transformer's famed attention mechanisms which provide unparalleled long-term memory may enable the creation of a model that will ultimately be more accurate and less computationally expensive than previously applied models. This provides the motivation to pursue the application of transformers in this manner with the goal of saving more lives by reducing the rate of women who die from cervical cancer.

At this benchmark stage, a compact transformer has been used from an existing model within Pytorch with slight modifications. The bulk of the progress achieved at this stage is in cleansing both of the datasets and modifying them in a way that allowed for a smooth concatenation of a single large image dataset. Now that the dataset has been transformed into a feasible input for a transformer, the next step is to begin the creation of a visual transformer.

In order to create and implement the ViT, a specific Python library known as Pytorch, had to be installed. Pytorch is an open source machine learning framework that is utilized for a range of applications, but most commonly for computer vision and natural language processing. Its overarching benefit is that it accelerates the path from research prototyping to actual deployment of the prototype/product. Thus, it is the

ideal library to install and incorporate for the application of a ViT.

Regarding the data cleansing process, the project has experienced some internal challenges so more information as to how both of the datasets were cleansed will be explained in this section of the report in the near future.

Regarding the data sources, a csv map of the SIPaKMeD images was created because the original files did not contain a compact csv of the images and their results. The Herlev dataset already contained csv files, but the dataset was still checked to ensure that both datasets would function well with one another as combined inputs for the ViT. A slight modification was made to both of the datasets in which all images in each were split into only three classes of cells: benign, normal, and abnormal. While both of the datasets originally contained the three overarching classes, they vary in which the SIPaKMeD dataset then splits these classes further into five sub-classes, whereas the Herlev datasets splits into seven sub-classes. Thus, both datasets were initiated at the three-class level in order to maintain consistency in the application of the ViT. The goal is that once the visual transformer is trained enough on the datasets, the deeper class splits will be explored for each dataset individually.

As explained, both datasets were split into three classes: benign, having the label 0, normal, having the label 1, and abnormal, having the label 2. The pre-trained models that can be loaded into Pytorch are setup to use datasets in a specific format which is why a custom image dataset class was created in the process.

Once the SIPaKMeD and Herlev datasets were cleansed, mapped into csv files, and loaded into dataframes, they were concatenated so the ViT could receive one large input of images for ideally better results. For the concatenation process, the individual cell images of the SIPaKMeD dataset were isolated and all attributes were dropped with exception of the 'LocalPath' and 'Class' attributes so the refined data would match up cleanly with the Herlev data. The concatenated image dataset was then split into training and testing sets and the visual transformer was implemented. For this initial run-through of the transformer, 15% of the dataset was designated as the testing data and the remaining 85% was designated as training data.

For both the training and testing sets, the transformer was set up to alter random training images for better results. In other words, the images were horizontally flipped and rotated during each iteration so the ViT would learn the characteristics of the images better and not rely solely on the placement of the cells as the classifying factor.

Finally, a separate function was created so that the ViT would show a single batch of images in a grid format along with their class labels to create a more compact and user-friendly view of the results.

V. METHODOLOGY

A. IMPLEMENTATION RESEARCH

Due to the complex nature of working with transformers, the team has been dedicated to expanding knowledge regarding the subject to ensure a wholesome understanding and most effective implementation. In the first weeks the team focused primarily on research. Additional help was provided by Dr. Prashant Shekhar of ERAU through lectures on the architecture and function of the original transformer model. These lectures helped to cement a foundation of understanding for encoder and decoder block functions, artificial neural networks, and machine translation.

B. DATA CLEANSING

Because transformers are high-variance models, they require very large amounts of data to prevent over fitting by means of generalization. While this often calls for datasets in the millions of images, the best we could do was concatenate the two datasets together by grouping them into shared three categories: benign, normal, and abnormal. This gave us a single dataset containing 4966 images, with an almost even distribution of the three classes.

C. CUSTOM DATASET

While the transformer was still in development, we also found out that PyTorch models require a special dataset class to properly create the tensor inputs. While PyTorch's built-in datasets feature an easy to call function from the library, other datasets must be created by hand. So, the dataset was the first part of the code which was finished and tested.

D. INITIAL TRANSFORMER MODEL

Eventually, a replica of the original vision transformer from "a picture is worth 16x16 words" was developed by following the papers methodology, and a lot of coding by Dr Shekhar. The transformer was built layer by layer to allow for maximum flexibility and modality, with a primary transformer constructor as the primary connector. This component acts essentially as the hub of the transformer because it ensures that each part of the transformer is called in the correct order and allows for passing on local variables. One notable change from the original code is that the sinusoidal positional encoding layer was replaced with a trainable matrix of zeroes. This modification allowed for adding another dimension to the data which the model could then use to develop the ideal parameters in relation to the position. This is referred to as "learnable" positional encoding.

Now that we had a working ViT, we could begin to prepare for the final model and testing the data.

E. FINAL TRANSFORMER MODEL

The next step we wanted to achieve was making the transformer compact by replacing the class embedding and patching layer with a Sequence Pooling layer and a convolutional layer, respectively. By using sequence pooling in place of class embedding, the class information can now be pooled

with the rest of the data. This provides some needed bias and allows the model to learn more accurate parameters in relation to the specific classes. The patching layer was replaced because, despite it being able to take more samples of the input data, an unfortunate side-effect would be the creation of, what were essentially, bordered segments. This made it harder for the model to learn the relationship between inputs and thus the bigger picture, without larger amounts of data. The implemented convolutional layer aids in eliminating this issue by utilizing convolutional filters to create multiple overlapping segments.

Once the CCT was completed, it was then trained on the concatenated SIPaKMeD and Herlev dataset. The previously designed custom dataset and dataloaders allowed the transformer to take the inputs as batches of images. The final step initialized the transformer and selected the desired loss function and optimizer and thus the results were obtained.

F. COMPUTATIONAL CHALLENGES

The size of the images themselves, posed a problem during the model's training process. In the first iteration, the images had been resized to 32x32 pixels, as these dimensions provided compactness and speed in being processed. The downside of this resize was that some of the finer details of the images would be lost. While this may not be a problem for the classification of more macro objects, like a car versus a dog, these fine details are essential to the classification of microscopic images. To try and mitigate this issue, the images were resized to as close to their original dimensions as possible, but in turn, this process more than doubled the amount of storage space that was required to hold the images in memory. This could be mitigated by reducing the batch size, or the number of images which would be taken as a single input by the model, but this could be even worse for the model's performance as the comparison between images is what allows it to establish the important parameters. Thus, the first hyper-parameter requiring optimization was a balance between the dimensions of the images and the batch size. The final balance was resizing every image to 64x64 pixels with a batch size of 128.

G. INITIAL TESTING AND OPTIMIZATION

To get an accurate comparison of the CCT model versus the ViT model, the first step was to optimize the hyper-parameters and lock in the RNG values so that a good baseline could be established. The loss function which was selected was Cross Entropy, as it is widely accepted as one of the best for multi class classification. To measure accuracy, we simply compared the total number of correct samples versus the total predictions made. Optimization was performed by running smaller epochs, in the interest of saving time, and changing the parameters one at a time to see if the overall accuracy was improved or not. This resulted in the following selections: patch size: 4, embedding dimensions: 128, encoder layers: 2, attention heads: 2, and an mlp ratio of 1. The Numpy random seed was set to 1, the PyTorch random seed was set to 17, and the initial number of epochs used for testing was set to 100.

VI. RESULTS

A. TESTING

The CCT produced visual results in terms of showcasing the labels of individual cells as either Normal, Abnormal, or Benign. In addition, the training loss and accuracy, as well as the validation loss and accuracy, were components of the output after every epoch the model conducted. Figure 3 showcases seven cell images along with their corresponding labels, as dictated in the original dataset:

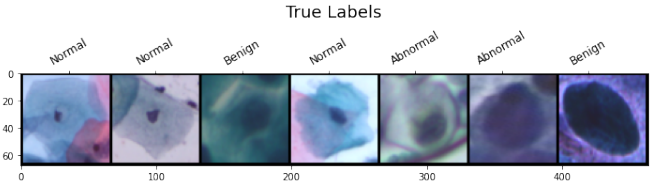

FIGURE 3: A sample batch of seven cell images from the original dataset that were part of input data fed to the CCT model.

In other words, the figure above shows seven random cell images from the original dataset with their true labels. This is a sample of the types of cell images that were fed into the CCT as input data.

Figure 4, showcases seven random cell images with their true labels listed on top, and the transformer's predicted labels on the bottom. Despite this figure being only a fractional sample of the dataset, it provides some insight on how the model performed in terms of its performance. Within the code, this component can be altered to showcase a much larger view of the comparison between the cell images' true labels and their predicted labels, if desired.

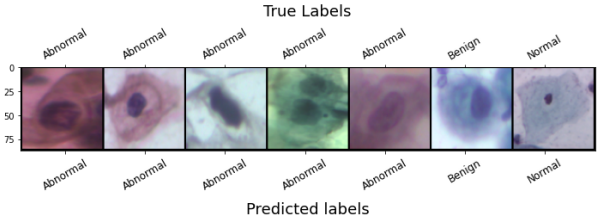

FIGURE 4: Seven cell images from the dataset with their true labels versus the transformer's predicted labels.

A loss is a number indicating how bad a model's prediction was on a single example. The loss function is used the optimize a ML algorithm. Thus, the goal of training a model is to identify the set of weights and biases that produce the lowest loss, on average, across all samples.

For this model, the very first epoch resulted in a test loss of 0.6831 and a test accuracy of 72.349. The final epoch resulted in a train loss of 0.0078, a train accuracy of 99.668, a test loss of 0.2234 and a test accuracy of 96.10. This substantial

improvement in both accuracy and loss demonstrates the model's improvement as the number of epochs increased, which is usually a desired trend. However, something to note is that the test loss shown in the graph is not for the particular epoch, but for the overall average loss. Figure 5 shows reflects these results of the standard ViT's model performance.

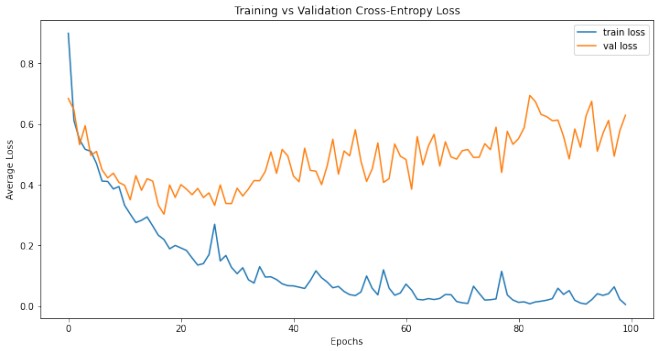

FIGURE 5: A graph showing the trends in train and test loss based on the standard ViT model's performance.

Figure 6 highlights the trends in train and test loss produced by the built CCT model. In this case, it had an average test accuracy of 88.47% and an average loss of 0.4516 after running 100 epochs.

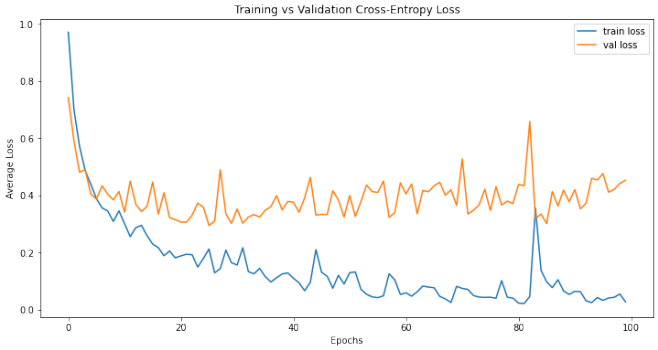

FIGURE 6: A graph showing the trends in train and test loss based on the CCT model's performance.

To further improve the results of the CCT model, in comparison to the standard model, the convolutional layer was normalized, thus resulting in the graph shown in Figure 7. The normalized model is identified as CCT2 and had an average test accuracy of 92.26% , which was roughly 2% higher than the base model and 4% higher than the non-normalized CCT.

B. AUGMENTATION APPLICATION

In order to improve the results of the CCT model that was built in comparison to the standard ViT, augmentations such as rotations, horizontal flips, brightness, and hue changes were applied. The table in Figure 8 below shows the augmentations that were applied to CCT2. These transformations

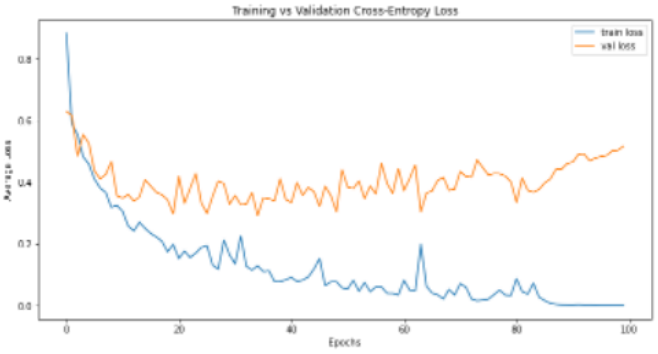

FIGURE 7: A graph showing the trends in train and test loss based on the CCT model's performance after being normalized.

were applied to the training set with every call of the test dataloader. The hope is that, by randomizing unimportant features like location, color, etc, it would force the model to focus on more important features, such as the structure of the cells themselves. This, ideally, aids the model in learning the input data better by reducing the tendency of trend or image pixel memorization.

Measurements	Base Model	CCT-2	+ Random Horizontal Flips (75%)	+ Random Rotation (60°)	+ ColorJitter (Brightness and Hue: 20%)
Avg Accuracy	90.82	92.26	93.14	94.37	94.66
Avg Loss	0.5089	0.5067	0.3304	0.1755	0.1597
Avg Epoch Duration (s)	7.68	9.06	9.94	10.50	11.79
Total Train Time (HH:MM:SS)	00:12:48	00:15:06	00:16:34	00:17:31	00:19:40
GPU Memory (GB)	3.1	3.4	3.5	3.5	3.6

FIGURE 8: CCT2 accuracy results after applying augmentations

Figure 9 shows the results of the CCT2 model after applying all the augmentations listed in Figure 8. It yielded an average testing accuracy of 96.22% . Looking at the graph, this version of the model produced the best results as it is evident that is was more stable and did not overfit as much as the other tested models.

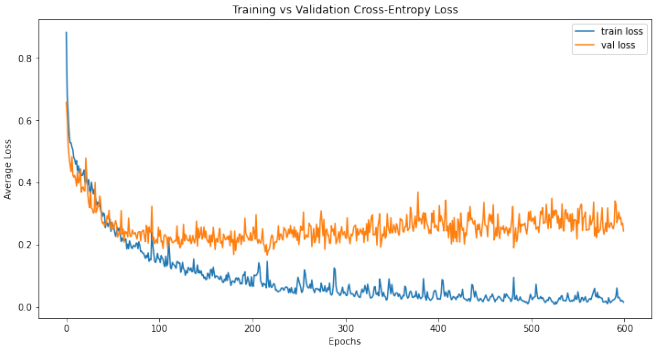

FIGURE 9: Results of CCT2 after applying all three augmentations and running the model for 600 epochs

VII. FUTURE WORK

Going forward, there are a variety of ideas that could be implemented or modified to this CCT with the hopes of further improving it's performance on the corresponding datasets.

For instance: exploring a difference in accuracy from preprocessing data augmentation versus the in-place we performed, using GANs to artificially inflate the dataset, and of course the implementation of any new improvements which may have occurred.

REFERENCES

[1] Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia, "Attention is all you need" , 2017.

[2] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).

[3] "Papers with Code - ImageNet Benchmark (Image Classification)," Paperswithcode.com, 2021. https://paperswithcode.com/sota/image-classification-on-imagenet (accessed Feb. 10, 2022).

[4] World, "Cervical cancer," Who.int, Dec. 02, 2019. https://www.who.int/health-topics/cervical-cancer (accessed Feb. 13, 2022).

[5] "Cervical Cancer Statistics | Key Facts About Cervical Cancer," Cancer.org, 2022. https://www.cancer.org/cancer/cervical-cancer/about/key-statistics.html (accessed Feb. 13, 2022).

[6] Marina E. Plissiti, Panagiotis Dimitrakopoulos, Gior-gos Sfikas, Christophoros Nikou, Olga Krikoni, AntoniaCharchanti, SIPAKMED: A new dataset for feature and im-age based classification of normal and pathological cervicalcells in Pap smear images, IEEE International Conference onImage Processing (ICIP) 2018, Athens, Greece, 7-10 October2018.

[7] "SIPaKMeD," Cs.uoi.gr, 2022. https://www.cs.uoi.gr/ marina/sipakmed.html (accessed Feb. 13, 2022).

[8] Cv2, "Cervical Cancer largest dataset (SipakMed)," Kaggle.com, 2018. https://www.kaggle.com/prahladmehandiratta/cervical-cancer-largest-dataset-sipakmed (accessed Feb. 13, 2022).

[9] "Downloads," Aegean.gr, Jul. 23, 2008. http://mde-lab.aegean.gr/index.php/downloads (accessed Feb. 13, 2022).

[10] Bhatt AR, Ganatra A, Kotecha K. Cervical cancer detec-tion in pap smear whole slide images using convNet withtransfer learning and progressive resizing. PeerJ ComputSci. 2021;7:e348. Published 2021 Feb 18. doi:10.7717/peerj-cs.348.

[11] Manna, A., Kundu, R., Kaplun, D. et al. A fuzzy rank-based ensemble of CNN models for classification of cervical cytology. Sci Rep 11, 14538 (2021). https://doi.org/10.1038/s41598-021-93783-8.

[12] Adaloglou, N. (2021, January 28). How the vision transformer (VIT) works in 10 minutes: An image is worth 16x16 words. AI Summer. Retrieved from https://theaisummer.com/vision-transformer/

[13] Pytorch. PyTorch. (n.d.). Retrieved from https://pytorch.org/

VIII. TEAM 5 INTRODUCTION

IF you were to ask a stranger in the street about the members of this illustrious team, they would most likely experience such a powerful combination of fear and respect that they would only be able to respond "who?"

THOMAS FIELLO II I'm from Western PA, though I've been living in Florida for the past ten years. I received a bachelor's in IT from Daytona State University but I was originally a Biology major. Aside from currently working as a GTA in the math department,I also work full-time as a freelance IT technician for local security companies. In my free time I can most likely be found playing video games, working out, and harassing my senior pug, Bane.

MARIAH MARIN I'm from Orlando, FL and I've lived in Florida my whole life. I received a bachelor's in Computational Mathematics with a minor in Humanities from Embry-Riddle Aeronautical University. I am currently a graduate teaching assistant in the mathematics department and a student assistant for the Humanities department. I've taken part in research projects that predicted flight delay propagation for commercial airlines, and another that utilized dynamic pricing based on private jet flight data to predict flight costs within a year. In my free time I like to read, visit the beach and spend time with my two cats.

WAIRIMU MWANGI I was born and raised in Nairobi, Kenya. I attained my bachelor's in Aerospace Engineering with a minor in Aviation Safety from Embry-Riddle Aeronautical University (ERAU). I am currently pursuing my master's in Data Science while working part-time as a Graduate Teaching Assistant (GTA) for the Engineering Fundamentals department at ERAU.In my free time I love to travel, participate in any outdoor activities, and try out different restaurants.

MARCELO RUIZ LEON, . I was born and raised in Ecuador, South America. I received my bachelor's in Finance and an MBA from the University of Edinburgh in the United Kingdom. I have previous experience in financial, telecommunications, and fin-tech industries. I am currently studying a master's in Data Science at Embry- Riddle University and working as GTA. In my free time, I love to travel, read, swim, and have off-road adventures.

ALEXIS SMITH Born and raised in southwest Florida, I moved to Embry-Riddle Aeronautical University to obtain my bachelors degree in Civil Engineering. In 2021 I began working on my masters degree in the civil department in the Environmental and Sustainable engineering track.I work as Lab Assistant in the Sustainable and Environmental Engineering Lab, and my research includes characterization of microplastics in the atmosphere. My free time is spent on the water, or with my cat Freddie.

IX. ROADMAP

A. PROJECT PLAN

What we plan to achieve with our project in the coming weeks: The team is hoping to continue the external learning.

Between the following deadlines, the team can be expected to dedicating multiple hours each week just to communication and external learning.

1) Feb 15 - Mar 11 (HW2)

~~We plan to finalize our datasets for training as well as start the initial implementation of the ViT and possibly GAN. Dataset is finalized but no ViT implementation yet.~~

2) Mar 12 - April 11 (HW3)

~~Initial model should be finished and we can begin the all of the modifications and refinement. Should be providing estimates/projections of the expected final results of the project.~~

3) April 12 - April 28 (Project Due)

Finalize training and record our best results. Finish the report and possibly publish findings.

B. INDIVIDUAL CONTRIBUTIONS

The team consists of five people. Potentially, two of these members will be hyper-focused on the understanding of the working of the complex components, where as three will be dedicated to the implementation of this project. The following contributions will be expanded upon as the team further develops the direction of the project.

- Thomas Fiello II:
  02/14/22: Group lead and in charge of Transformer research as well as implementation. Going forward, will continue to perform independent study followed by tutoring of other team members.
  03/11/22: Performed further research on transformers. Learned how to build, and successfully implemented, a custom Pytorch Dataset for working with our images going further. Then walked other team members and students through the code implementation. Next step is beginning the construction of the ViT.
  04/11/22: Finalized the ViT, CVT, and the CCT model. There are still some possible improvements to explore and then the optimization and testing can begin.
  04/28/22: The work is done, the paper is almost completely finished and now I may sleep.

- Mariah Marin:
  02/14/22: Conducted research on related techniques and models created to classify cervical cell data both in general and the SIPakMeD dataset specifically. Tasked with continuously updating the report and further developing the report's background and literature section.
  03/11/22: Wrote proposed approach section and incorporated more research papers on visual transformers in background and literature section. Update sections of the report based on expectations listed for next submission. For the next benchmark, will continue to update the report as results are produced and methodology is refined as well as attempt to work with the transformer code created by the data modelers and vary the parameters to determine which combination provides the best accuracy in image classification.
  04/11/22: Ran through Thomas' code to understand transformer, how it was performing with the data and make connections between the literature reviews to actual implementation of the model. Updated the report with the final results of the transformer as well as some details on the final CCT model itself and future work related to this project.

- Wairimu Mwangi:
  02/14/22: Tasked with providing research support and implementation of transformers and neural networks needed to complete the project algorithm which is aimed at providing improved cell imaging in the medical field.
  03/11/22: Provided additional research information on the implementation of transformers. Supported other members in understanding the dataset cleansing process and application. Will be working on further building the code to meet attain the application of transformers on image recognition and prediction of cervical cancer cells.
  04/11/22: Created a PowerPoint presentation outline of the tasks carried out during the project and results achieved. Expounded on the methodology used to create the ViT and its application in predicting and classing the classes of cervical cancer cells in the dataset that was used.
  04/28/22: Updated the report and included information on the methodology, results, and applied augmentations of the tested models.

- Marcelo Ruiz Leon:
  02/14/22: Researching for applications, contribution for current research and findings. Looking previous applications of transformers in similar projects.
  03/11/22: Performed data cleansing after researching subject matter.
  04/11/22: Marcelo, write something here.
  04/28/22: Marcelo, write something here.

- Alexis Smith:
  02/14/22: Continuing external and independent learning to solidify a base knowledge regarding transformers and machine learning.
  03/11/22: Got sick and almost died.
  4/11/22: Continued learning performance mechanisms of multiple types of transformers, explored the relationships between Artificial Intelligence, Machine Learning, and Visual Computer Learning
  4/28/22: Contributed Background Information regard-

ing machine learning, as well as biological information in delivering of presentation.

APPENDIX. CODES

The coding language used for this project was Python, more specifically Jupyter Notebook. The following is an example of the code used for the custom dataset.

```
#Importing packages & setting up environment
import os
from PIL import Image
import torch
import torchvision.transforms as transforms
import torchvision

from torch.utils.data import DataLoader,Dataset

import cv2
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

# CUDA for PyTorch
use_cuda = torch.cuda.is_available()
device = torch.device("cuda:0" if use_cuda else "cpu")
torch.backends.cudnn.benchmark = True

#Create a CSV map of the SIPaKMeD images
def sipakmed_prep():
    root = "./CervicalCancer/Sipakmed/"

    all_img = []
    for path, subdirs, files in os.walk(root):
        for name in files:
            curr_img = os.path.join(path,name)
            all_img.append(curr_img)

    sipakmed_df = pd.DataFrame()
    sipakmed_df['LocalPath'] = all_img
    sipakmed_df['ID'] = sipakmed_df['LocalPath'].apply(lambda x: str(x.split("\\")[-1]))
    sipakmed_df['Slide'] = sipakmed_df['LocalPath'].apply(lambda x: 'False' if
        (str(x.split("\\")[1])) == 'CROPPED' else 'True')
    sipakmed_df['Specific_Status'] = sipakmed_df['LocalPath'].apply(lambda x:
        str((x.split("im_")[1]).split("\\")[0]))
    sipakmed_df['Status'] = sipakmed_df['Specific_Status'].apply(lambda x: 'Benign' if x ==
        'Metaplastic' else
    ('Normal' if x == 'Superficial-Intermediate' or x == 'Parabasal' else 'Abnormal'))

    sp_class = {'Metaplastic':4, 'Superficial-Intermediate':0, 'Parabasal':1,
        'Koilocytotic':2, 'Dyskeratotic':3}
    sipakmed_df['Specific_Class'] = sipakmed_df['Specific_Status'].map(sp_class)
    gen_class = {'Benign':0, 'Normal':1, 'Abnormal':2}
    sipakmed_df['Class'] = sipakmed_df['Status'].map(gen_class)

    sipakmed_df =
        sipakmed_df.loc[sipakmed_df['ID'].str.contains('bmp')].reset_index(drop=True)

    return sipakmed_df

#Prepare the Herlev Dataset to work with the SIPaKMeD one
def herlev_prep():
    root = './CervicalCancer/Herlev/smear2005/results.csv'
    img_folder = './CervicalCancer/Herlev/smear2005/Images/'

    herlev_df=pd.read_csv(root)
    herlev_df = herlev_df.rename(columns={'Class':'Specific_Class'})

    gen_class = {1:1, 2:1, 3:1, 4:2, 5:2, 6:2, 7:2}
    herlev_df['Class'] = herlev_df['Specific_Class'].map(gen_class)
```

```
    herlev_df['LocalPath'] = herlev_df['ID'].apply(lambda x: str(img_folder + x))
    return herlev_df

#Now both datasets have classes: 0 (Benign), 1 (Normal), and 2 (Abnormal)

#Create the custom image dataset class for pytorch
class DatasetSample(Dataset):
    def __init__(self, df, transform):
        # Initialization
        self.transform = transform

        self.image_names = df['LocalPath']
        self.img_class = np.array(df['Class'])

    def __len__(self):
        # Denotes the total number of samples
        return len(self.image_names)

    def __getitem__(self, index):
        # Retrieves the data
        image=cv2.imread(self.image_names.iloc[index])
        image=cv2.cvtColor(image,cv2.COLOR_BGR2RGB)

        image = self.transform(image)
        label = self.img_class[index]

        x, y = image, label

        # Enable to output as single variable instead
        #sample = {'image': image,'labels':label}

        return x, y

#Function to prepare the dataset
def concat_data():
    # Load the dataframes
    sipakmed_df = sipakmed_prep()
    herlev_df = herlev_prep()

    # Final adjustments to make sure they're attributes match
    sipakmed_df = sipakmed_df.loc[sipakmed_df['Slide'] == 'False']
    herlev_df = herlev_df.loc[:,['LocalPath', 'Class']]
    sipakmed_df = sipakmed_df.loc[:,['LocalPath', 'Class']]

    concat_df = pd.concat([sipakmed_df,herlev_df], ignore_index=True)

    # Split them into training and testing sets
    train_set,test_set = train_test_split(concat_df,test_size=0.15)

    # Set it up to alter random training images for better results
    train_transform = transforms.Compose([
                    transforms.ToPILImage(),
                    transforms.Resize((32, 32)),
                    transforms.RandomHorizontalFlip(p=0.75),
                    transforms.RandomRotation(degrees=60),
                    transforms.ToTensor()])

    test_transform = transforms.Compose([
                    transforms.ToPILImage(),
                    transforms.Resize((32, 32)),
                    transforms.ToTensor()])

    train_dataset = DatasetSample(train_set, train_transform)
    test_dataset = DatasetSample(test_set, test_transform)

    Dtr = DataLoader(
        train_dataset,
```

```
        batch_size=32,
        shuffle=True
    )

    Dte = DataLoader(
        test_dataset,
        batch_size=4,
        shuffle=True
    )

    return(Dtr, Dte)

#Function to show a single batch of results
def imshow(images, labels, predicted_labels=None):

    # Using torchvision to make a grid of the images
    img = torchvision.utils.make_grid(images)

    # Inverting the normalization
    img = img.numpy().transpose((1, 2, 0))
    img = np.clip(img, 0, 1)

    # Plotting the grid
    fig, ax = plt.subplots(figsize=(6, 24))
    plt.imshow(img)

    if predicted_labels is not None:
        # Outputing the predicted labels
        ax.set_xlabel('Predicted labels', fontsize=18, labelpad=12)
        ax.set_xticks(torch.arange(len(images)) * 35 + 20)
        ax.set_xticklabels([classes[predicted_labels[j]]
                            for j in range(len(images))], fontsize=14)

    # Outputing the Real truth labels
    gax = ax.secondary_xaxis('top')
    gax.set_xlabel('Real truth', fontsize=18, labelpad=12)
    gax.set_xticks(torch.arange(len(images)) * 35 + 20)
    gax.set_xticklabels([classes[labels[j]]
                         for j in range(len(images))], fontsize=14)
    plt.show()

#Output:
# Sets classes for the imshow function
classes = ('Benign', 'Normal', 'Abnormal')
Dtr, Dte = concat_data()

# Retrieves a batch of samples
images, labels = next(iter(Dte))
imshow(images, labels)
```