Robust Convolutional Vision Transformer Hybrid for Multi-Depth Magnification Pigmented Skin Lesions Diagnosis

Jonathan Gong[†, *]
[†] University of Waterloo

Abstract

Purpose

This paper proposes a robust convolutional vision transformer (ViT) hybrid model for diagnosing multi-depth magnification pigmented skin lesions. The objective is to leverage the strengths of both convolutional neural networks (CNNs) and vision transformers to improve diagnostic accuracy.

Models and Data

The model combines DenseNet201 for feature extraction and a vision transformer for capturing global contextual information. The proposed hybrid model was trained and evaluated on the Mpox Skin Lesion Dataset (MLSD), consisting of multi-class skin lesion images.

Results

The hybrid CNN+ViT model demonstrated superior performance compared to standalone CNN models and the model with the highest accuracy from the original paper. The results showed higher test accuracy, precision, recall, and f1-scores across various skin lesion classes, highlighting the efficacy of integrating self-attention mechanisms in ViTs.

Conclusion

Our hybrid model presents a novel approach to enhancing the accuracy of skin lesion diagnosis, offering significant potential for application in medical image analysis. This model can assist dermatologists and medical professionals in making more accurate diagnoses, ultimately improving patient outcomes. Future work will explore further optimization and application to other medical imaging datasets.

Keywords: Artificial Intelligence, Deep Learning, Machine Learning, Radiography, Data Mining, Image Processing, Biomedical Engineering

1. Introduction

The skin, often heralded as the body's largest organ, serves as protection from the external environment and foreign substances. However, being the barrier between our internal system and external world, it's prone to a myriad of diseases, reflecting not just dermatological disorders but sometimes even underlying systemic diseases. Central to understanding skin diseases are skin lesions, which are abnormalities in the skin's texture, appearance, or composition. Understanding these lesions is crucial to diagnosing various skin disorders. In May 2022, an outbreak of a viral disease called mpox, popularly referred to as "Monkeypox", was officially confirmed. As of August 2023, the virus had accumulated 89,385 cases, spanning across the world [1]. Monkeypox, refers to a rare but potentially severe viral disease that has garnered attention in recent years due to its zoonotic nature. Originating in remote parts of Central and West Africa, Monkeypox has, over the years, expanded its geographical footprint, causing concern among global health organizations. However, in May 2023, the World Health Organization (WHO) declared an end to the global health emergency that was declared in response to the worldwide outbreak of the Monkeypox virus [2]. With this declaration, the WHO still emphasized the need for "long-term partnerships to mobilize the needed financial and technical support for sustaining surveillance, control measures and research for the long-term elimination of human-to-human transmission" [2].

AI has made its appearance in society as a way of stimulating computers to think like humans. Tasks that are human-like such as learning and problem-solving, have been the goal that AI tries to mimic. The main selling point of AI is an attempt to rationalize decisions, eliminate human errors, and take the best actions to achieve a goal. The field of Machine learning is a common subdivision of AI aimed at developing models to be trained with data (called training data) to perform a task. Without explicit instruction, this model is utilized to make predictions or decisions. According to a survey done by Deloitte, machine learning accounts for 63% of company applications using AI [3]. A truly broad technique that can be applied to almost any industry and an industry that is projected to rise from USD 387.45 billion in 2022 to USD 1394.30 billion in 2029 [4].

[*] j56gong@uwaterloo.ca

One way of diagnosing skin disease patients is through a simple body image or dermatoscopic images of a featured skin

lesion. Whereas body images can be captured using any variety of digital camera, dermatoscopic images require specialized microscopes. Moreover, the current well-established skin lesion dataset, HAM10000 [5], focuses on closely magnified images of pigmented skin lesions, which raises the question if working models would perform similarly to multi-depth magnification images of the same lesions. A deep learning model can predict if a patient has certain skin diseases or not by feeding one's body image through the model (this is called a classification model). Current SOTA models for HAM10000 and related datasets revolve around models with CNN as their model architecture such as [6] with a 96.49% accuracy where transformer or Visual Transformer models perform less accurately as shown in [7] with a 94.3%. However, the HAM10000 dataset does not show the strength of ViTs as with this dataset the attention layer could only compare with the same lesion, when compared to multi-depth magnification images ViTs can show their advantage with self-attention between each lesion in the image. With the Mpox Skin Lesion Dataset (MLSD) [8, 9], we can find the potential of ViTs for diagnosing multi-depth magnification skin diseases.

We prove here the efficacy of the combination of CNNs and ViTs for skin disease diagnosis in this captured medium. The hope is for radiologists and other medical professionals in pathology to use this procedure for the benefit of the patient's diagnosis and well-being.

2. Classification Model

2.1. CNN+ViT Introduction

With the advancement of AI and machine learning models, they have found their place in the field of medical imaging. In the area of medical image recognition, the convolutional neural network is widely accepted and achieves SOTA accuracy on many medical imaging datasets. The deep structure of a convolutional neural network (CNN) is employed to extract features from numerous small objects in an image, but it struggles to pinpoint the truly significant areas. The transformer model coined by Vaswani et al. [10] is the leading model in the field of natural language processing. This model uses a technique called self-attention which subsequently inspired the Visual Transformer (ViT) [11]. The ViT serves to use the transformer architecture for image classification applications. In these models, the training approach involves segmenting the input image into patches and considering each patch as an equivalent to a word in natural language processing. Self-attention mechanisms are used to understand the relationships between these segmented patches. Lately, the Vision Transformer (ViT) has demonstrated exceptional performance in standard classification tasks. The self-attention component in the transformer serves to enhance important features while minimizing the impact of noisy ones.

By combining both CNNs and ViTs, we can overcome some limitations of each of the models to outperform existing architectures. Where CNNs fail due of their strong inductive bias in some cases, ViTs feature minimal inductive bias which makes ViTs hard to train on smaller datasets. By leveraging both architectures, we can more easily find the balance of this bias to sculpt a strong model.

2.2. Methodology

To better understand the process in which we trained the model, the approach used is displayed in Figure 1.

- 1) The acquisition of the skin lesion image dataset
- 2) We apply normalization and our own data augmentation techniques such as resizing, reflection, rotation, and scaling upon the original data for improved performance.
- 3) The feature extraction is conducted by our hybrid Conv + ViT model through image division into patches.
- 4) The model is trained on a train and test dataset with a 90-10 train-test split. We use the same identical test dataset the original paper uses.
- 5) We evaluate the model using metrics such as accuracy, loss, precision, recall, and f1-score.

2.3. Model Architecture

2.3.1. DenseNet201 branch

The input image is first fed through the DenseNet [12] architecture before being concatenated with the output of the transformer encoder. To be more precise, the model uses DenseNet201, which is in the middle of the DenseNet family (DenseNet121-DenseNet264). DenseNet has twenty million two hundred thousand parameters and is 201 layers deep hence the name DenseNet201. DenseNet uses element-wise addition, similar to ResNet [13]. By viewing each module with the state of the last module, each module receives extra data from the past modules culminating in a hopefully more effective model. However, DenseNet differs from ResNet as DenseNet receives not only the state of the last module but all the modules before it. Concatenating the inputs received from all preceding layers, DenseNet passes on the feature map of the preceding layers to the next layer. Each layer collects the collective knowledge from each preceding layer. By using a pretrained DenseNet model on the imagenet dataset, we are implementing transfer learning to our model.

The DenseNet model is shown in Figure 2.

Figure 1. **Our Methodology**. This diagram explains our five-step process.

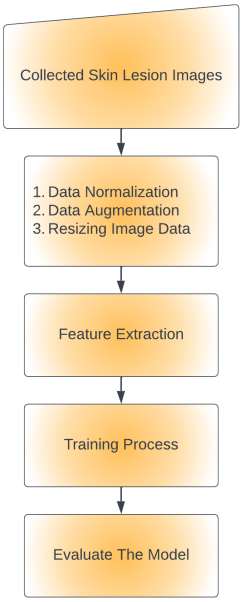
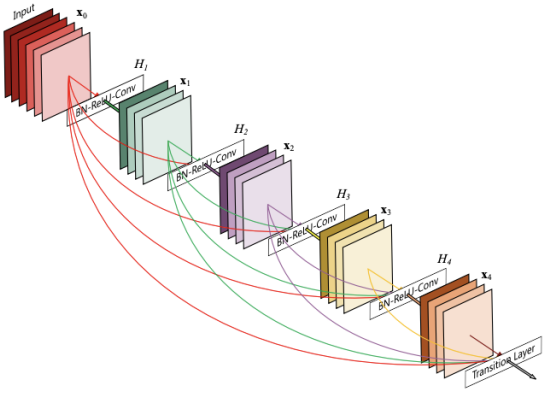


Figure 2. **DenseNet Model Diagram**. This model shows how DenseNet processes data. This is proposed in [12].



2.3.2. **Patch Embedding and Position Encoding**

After data processing and augmentation, the image data enters the traditional ViT structure as presented in Dosovitskiy et al.[11]. The complete architecture of the ViT is shown here Figure 3.

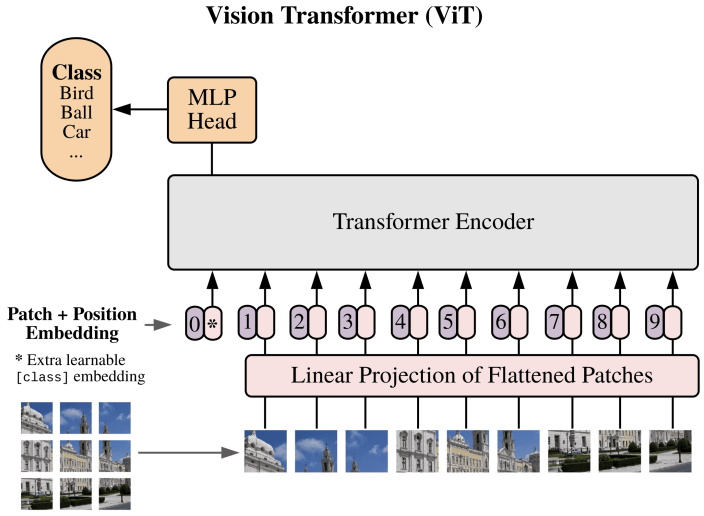
As shown in the figure the patch and embedding are performed on the image data first, which split the image into patches, equally sized 8 by 8 pixels. If the input image is shaped $H \times W \times C$ where H is the height, W is the width, and C is three channels for the RGB encoding, we can see how each patch is represented in equation 2.1. The patch's top left index is defined as [r, c] and the patch's size is determined as $P \times P$ which when used as parameters outputs the rows one patch represents, $r \times P : (r+1) \times P$, and the columns, $c \times P : (c+1) \times P$, with the predetermined RGB channels represented by the ending colon.

$$Patch(r, c) = Image[r \times P : (r+1) \times P, c \times P : (c+1) \times P, :] \tag{2.1}$$

The patches are passed through an embedding layer which first flattens the patches into 1D array. If the number of patches the have been segmented from the input image is equal to N, and the shape of a patch is $P \times P \times C$, the resulting 1D flattened array would result in a dimension of $N \times (P \times P \times C)$.

This 1D array would then be fit into a linear embedding layer with a predetermined projection dimension, which this proposed ViT model has been tuned to 32. Each patch is encoded through this linear layer with their respective weights and biases.

Figure 3. **ViT model architecture**. This diagram is from Dosovitskiy et al.[11]



To initialize the position encoding, each patch is passed through a position encoding layer that assigns a position to each patch. This process is illustrated in the equation 2.2 which is repeated for each patch from the range $[0, N)$, where N is the number of patches.

$$FinalInput = PatchEmbedding + PositionalEncoding \tag{2.2}$$

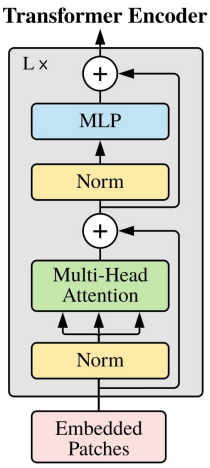
The patches of one image is visualized in Figure 4.

Figure 4. **Patch Visualisation** Patches visualized on an sample datset image.

2.3.3. **Transformer Encoder**

The embedded and encoded patches are then sent through the transformer encoder, the architecture of which is shown in Figure 5.

Figure 5. **Transformer Encoder Diagram**. This diagram explains how a transformer encoder processes data. This is proposed in [11].



First passing through a layer normalization layer proposed in Ba et al. [14], which works by simulating the data as a Gaussian distribution as presented in Equation 2.3. Layer normalization normalizes the previous output, and in our case, this is the output of the position encoded layer. The output is made by dividing the mean value (μ_{layer}) from the layer (x) and dividing it by the standard deviation (σ^2_{layer}) and ϵ for a legal division. With the γ and β being learnable scale and shift parameters, respectively.

$$\text{LN}(x) = \gamma \left(\frac{x - \mu_{\text{layer}}}{\sqrt{\sigma^2_{\text{layer}} + \epsilon}} \right) + \beta \tag{2.3}$$

Using the Query (Q), Key (K), and Value (V) matrices from the output of the layer normalized patches, we calculate the attention scores and input them into each head of the multi-head-attention (MHA) layer which we set the three heads. The full process of the MHA layer is shown in equation 2.4 which is introduced in Vaswani et al. [10].

$$\begin{aligned} \text{Attention Scores} &= \frac{QK^T}{\sqrt{d_k}} \\ \text{Output of one Head} &= \text{softmax}(\text{Attention Scores}) \times V \\ \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \end{aligned} \tag{2.4}$$

The output of the MHA layer is then concatenated with the output of the original position encoded patches which is then fed into another layer normalization layer. This is then passed through a MLP layer which is made of two dense layers with 64 and 32 nodes respectively. With our model we have set the amount of layers of transformer encoder blocks to 8. The output of the MLP layer is then concatenated with the previously concatenated output of the MHA layer and encoded patches. This structure gives an output that has feature maps of three processed versions of the same data, one through raw patch encoding, MHA, and MLP.

2.3.4. **MLP head**

After receiving the output from the transformer encoder, it is then normalized, flattened, and concatenated with the output of the DenseNet branch. This concatenated output is then sent into a MLP head with three dense layers containing 4096, 1072, and 256 nodes, respectively. Each dense leayer is paired with a dropout layer with factors of 0.3, 0.2, 0.15. Each dense layer, excluding the output layer, uses the ReLU activation function [15]. The ReLU (rectified linear unit) is one of the most popular activation functions for Deep Neural Networks.

ReLU acts like a linear function as a piecewise function while still providing the neural networks to learn complex mapping functions. Thus effectively eliminating the downside of nonlinear activation functions, which prevent learning algorithms from learning efficiently in deep networks. ReLU works by returning the input if it is above zero; otherwise, it will return zero. This can be expressed in Equation 2.5.

$$f(x) = max(0, x) \tag{2.5}$$

The last layer of the MLP head uses the softmax activation function [16]. The output layer outputs six units corresponding to the labels of skin diseases in our dataset. Softmax, also known as a normalized exponential function, converts a vector of inputs into a probability distribution of the possible outcomes of possible inputs.

Softmax works by normalizing the inputs into a probability distribution over the output classes. Inputs that have a range of negative numbers, zero, and positive numbers, and after applying softmax, these inputs will be in the interval [0, 1] where the components add up to 100%, thus acting like a categorical distribution of the number of outputs. In this case, we have six outputs. The mathematical representation of softmax is shown in Equation 2.6 where each logit's exponential (z_i) of vector $\vec{z}$ is divided by the sum of all logits' exponential in vector $\vec{z}$ which ensures the output will always be in the interval [0, 1].

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.6}$$

The model uses the Adam optimizer algorithm [17]. The Adam optimizer algorithm is an extension of stochastic gradient descent that has been widely used in deep learning applications. The task for optimizers such as Adam is to grant smoothness to the objective function. Adam's popularity is attributed to its speed in achieving good results. Adam works by adapting the learning rate based not on the average first moment but on the average of the first and second moments of the gradients. Adam is expressed mathematically in Equations 2.7 2.8 2.9 [17]. m_t and v_t represent the initial estimates of the first and second moments where the bias-corrected equations for the two moments are shown in $\hat{m}_t$ and $\hat{v}_t$. These are then used to update parameters which can be expressed as the Adam update rule (θ_{t+1}).

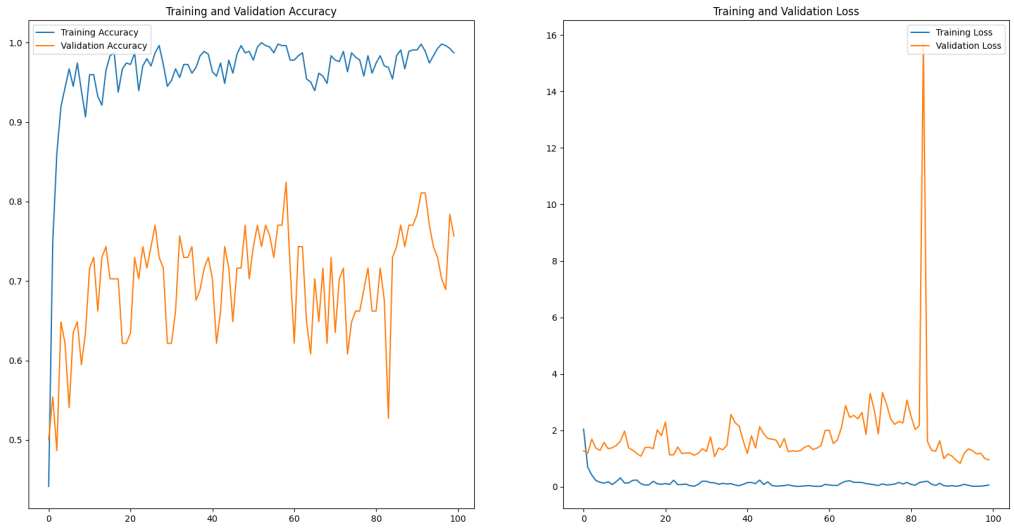
$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \tag{2.7}$$

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \end{aligned} \tag{2.8}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \tag{2.9}$$

The accuracy and loss graphs can be seen in Figure 6

Figure 6. **Accuracy and Loss.**



The loss function used for the model output is categorical-crossentropy. Categorical-crossentropy is used for the model for its specialty in producing loss from one-hot encoded labels. It is also widely used for multi-class classification, which suits our purpose. Categorical-crossentropy works by calculating the loss from the negative sum of the log of outputs multiplied by their respective target values. Where $\hat{y}_i$ is the ith scalar output of the model and y_i is the ith target value.

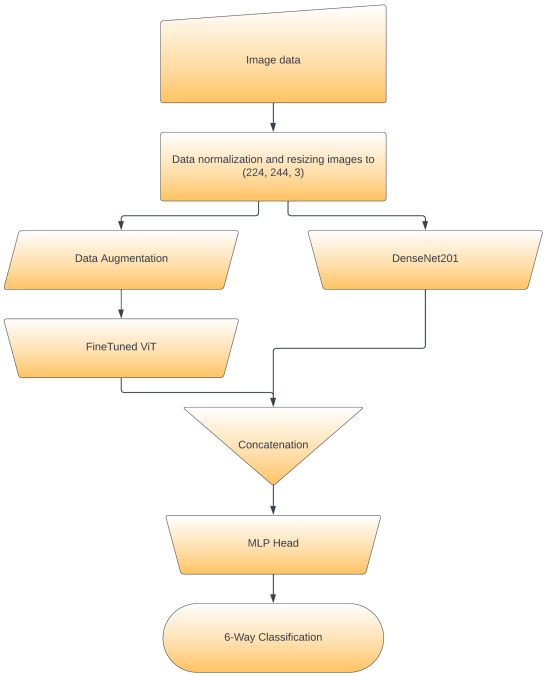
The categorical-crossentropy loss function is shown mathematically in Equation 2.10.

$$\text{Loss} = -\sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i \tag{2.10}$$

The learning rate is a hyperparameter that governs how much the model tunes when encountered with the estimated error each time the model weights are updated. The learning rate of the model proposed is 0.0001 and the weight decay being 0.0001 as well. This rate differs from the default rate of Adam, being 0.001. The default learning rate causes the model to converge too quickly to a result. Therefore, we change the rate to find a more optimal convergence.

Our model system is as follows in this Figure 7.

Figure 7. **General Model Architecture**.



3. Application

3.1. Introduce Data

For our study, we sourced the data from the second version of the Mpox Skin Lesion Dataset (MLSD) [8, 9]. The race distribution of the data is shown in Figure 8. The image data contains web-scraped images belonging to 6 classes: mpox (284 images), chickenpox (75 images), measles (55 images), cowpox (66 images), hand-foot-mouth disease or HFMD (161 images), and healthy (114 images) for multi-class classification. Most of the images are shaped (224, 224, 3) with the three channels belonging to the RGB values. However, some images in the dataset were not resized to this shape thus we reshaped those images to (224, 244, 3) as well.

For our purposes, we used the first folder given for the training data, and subsequent validation and test data.

The dataset is also found to have hundreds of duplicates included in the given augmented data. For that reason, our proposed model is not trained on any augmented image data and instead the original images only. Thus the comparison with our results and the results from the original paper are not directly correlated as our models were trained on a lack of augmented imaging. However, the original paper had results with their non-augmented images which we draw a comparison to.

The different classes of images are listed in Figure 9

3.2. Results and Ablation Study

Here the results of the inference of the first folders test data are listed in Table 1.

Figure 8. **This Figure is from [9].**

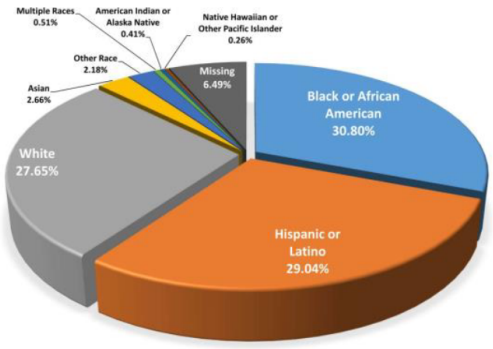


Figure 9. **Images from [9].**

Table 1. **Classification results**. The table shows results of the ablation study conducted comparing the CNN based model with the CNN+ViT based model and the highest accuracy (DenseNet) model from the original study

Model Type	Train Accuracy	Train Loss	Test Accuracy	Test Loss
CNN	0.9816	0.0623	0.7432	1.0582
CNN+ViT	**1.0**	**0.0124**	**0.8243**	**0.8234**
DenseNet121[9]	N/A	N/A	0.8170	N/A

Table 2. **Classification Report**. The report of the classification model

Labels	precision	recall	f1-score	support
Monkeypox	0.72	0.96	0.83	27
Cowpox	1.00	0.29	0.44	7
Healthy	0.88	0.78	0.82	9
HFMD	0.94	0.94	0.94	16
Measles	0.83	0.83	0.83	6
Chickenpox	1.00	0.67	0.80	9
accuracy	N/A	N/A	0.82	74
macro avg	0.89	0.74	0.78	74
weighted avg	0.86	0.82	0.81	74

4. Conclusion

In this study, we proposed a robust convolutional vision transformer (ViT) hybrid model for the diagnosis of multi-depth magnification pigmented skin lesions. Our approach leveraged the strengths of both convolutional neural networks (CNNs) and vision transformers to create a model capable of effectively handling the complexity and variability of skin lesion images.

Through the integration of CNNs for feature extraction and ViTs for capturing global contextual information, we achieved a significant improvement in diagnostic accuracy compared to using CNNs alone.

Our methodology involved preprocessing steps such as normalization and data augmentation, followed by the application of a hybrid model architecture. The DenseNet201 branch and ViT components worked in tandem to process images and extract meaningful features, which were then combined to make final classifications. The model was trained and evaluated on the Mpox Skin Lesion Dataset (MLSD), demonstrating superior performance metrics, including higher test accuracy and improved precision, recall, and f1-scores across various skin lesion classes.

The results from our ablation study highlighted the efficacy of our hybrid approach, with the CNN+ViT model outperforming the standalone CNN model. This indicates that the integration of self-attention mechanisms in ViTs provides a complementary advantage in understanding complex skin lesion patterns, especially when dealing with images of varying magnifications.

Our work contributes to the field of medical image analysis by presenting a novel approach that combines state-of-the-art deep learning techniques to enhance the accuracy of skin lesion diagnosis. This hybrid model has the potential to assist dermatologists and medical professionals in making more accurate diagnoses, ultimately improving patient outcomes. Future work could explore the application of this model to other medical imaging datasets and further optimization of the model architecture to enhance its generalizability and performance.

5. Code availability

The code is available at https://github.com/jogong2718/MpoxViT

References

[1] C. for Disease Control and Prevention. *2022 Mpox Outbreak Global Map*. URL: https://www.cdc.gov/poxvirus/mpox/response/2022/world-map.html. (accessed: 08.26.2023).

[2] WHO. *Fifth Meeting of the International Health Regulations (2005) (IHR) Emergency Committee on the Multi-Country Outbreak of mpox (monkeypox)*. URL: https://www.who.int/news/item/11-05-2023-fifth-meeting-of-the-international-health-regulations-(2005)-(ihr)-emergency-committee-on-the-multi-country-outbreak-of-monkeypox-(mpox). (accessed: 08.26.2023).

[3] J. Loucks. *State of AI in the Enterprise, 2nd Edition*. URL: https://www2.deloitte.com/us/en/insights/focus/cognitive-technologies/state-of-ai-and-intelligent-automation-in-business-survey-2018.html. (accessed: 09.20.2022).

[4] X. M. Size. "Share&Covid-19 Impact Analysis". In: *By Application (Imaging and Lighting, Medical, satellite, Electronics&Semiconductors, and others (including R&D), and Regional Forecast, 2020–2027. Available online: https://www. fortunebusinessinsights. com/xenon-market-101965 (accessed on 9 February 2021)* (2021).

[5] P. Tschandl, C. Rosendahl, and H. Kittler. "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions". In: *Scientific data* 5.1 (2018), pp. 1–9.

[6] Z. Lan, S. Cai, X. He, and X. Wen. "Fixcaps: An improved capsules network for diagnosis of skin cancer". In: *IEEE Access* 10 (2022), pp. 76261–76267.

[7] C. Xin, Z. Liu, K. Zhao, L. Miao, Y. Ma, X. Zhu, Q. Zhou, S. Wang, L. Li, F. Yang, et al. "An improved transformer network for skin cancer classification". In: *Computers in Biology and Medicine* 149 (2022), p. 105939.

[8] S. N. Ali, M. T. Ahmed, J. Paul, T. Jahan, S. M. S. Sani, N. Noor, and T. Hasan. "Monkeypox Skin Lesion Detection Using Deep Learning Models: A Preliminary Feasibility Study". In: *arXiv preprint arXiv:2207.03342* (2022).

[9] S. N. Ali, M. T. Ahmed, T. Jahan, J. Paul, S. M. S. Sani, N. Noor, A. N. Asma, and T. Hasan. "A Web-based Mpox Skin Lesion Detection System Using State-of-the-art Deep Learning Models Considering Racial Diversity". In: *arXiv preprint arXiv:2306.14169* (2023).

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[13] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[14] J. L. Ba, J. R. Kiros, and G. E. Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[15] V. Nair and G. E. Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Icml*. 2010.

[16] J. S. Bridle. "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition". In: *Neurocomputing*. Springer, 1990, pp. 227–236.

[17] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).