Image Classification using Vision Transformers

Authors
Viswajith Menon (vm623)
Zeeshan Ahsan (za224)
Kevin Paul (kc1368)
Pranoy Sarath (ps1279)
Reuben Samuel Varghese (rsv39)

Abstract

CNN architecture-based models have been state-of-the-art vision models for a long time. The success of large-scale training of transformers in the field of NLP encouraged researchers to adopt the attention mechanism for vision models, and it has been shown that vision models with pure transformer architecture can perform very well on Image classification. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to the existing state-of-the-art CNNs while requiring sub- substantially fewer computational resources to train. Many advancements have been made to the vanilla ViT that improves the performance and accuracy.

The primary aim of this project is to enhance skin cancer detection accuracy (Ham10000 dataset) by employing modern deep learning architectures such as Vision Transformer models, fine-tuning them and comparing them against the CNN-based models (ResNet50, VGG16, DenseNet121, VGG19, and ResNet152) which would act as the benchmark.

The transformer models that we have explored are

- **Vision Transformer (ViT)**
- **Convolutional Vision Transformer (CvT)**
- **BERT Pre-Training of Image Transformers (BEiT)**
- **Hierarchical Vision Transformer using Shifted Windows (Swin Transformer)**

1 Data Preprocessing

1.1 Dataset

The dataset used is the publicly available benchmark dataset known as HAM10000(Human Against Machine with 10000 training images). This consists of dermatoscopic images from different populations, acquired and stored by different modalities. The final dataset consists of 10015 dermatoscopic images which can serve as a training set for deep learning purposes.

The images are a representative collection of all important diagnostic categories in the realm of pigmented lesions. These are classified into 7 types of skin lesion namely • Melanocytic nevus (nv) • Actinic keratosis (akiec) • Basal cell carcinoma (bcc) • Dermatofibroma (df) • Vascular lesion (vasc) • Malignant melanoma (mel) • Benign keratosis (bkl).

The images have been standardized to a 224x224x3 input size RGB image.

1.2 Dataset Augmentation

The dataset is highly imbalanced with Melanocytic nevus having the most images. In order to balance the dataset, several data augmentation techniques like flipping, rotating and tilting have been implemented.

Dataset before Augmentation -

Total Images	akiec	bcc	bkl	df	mel	nv	vas
10015	297	479	1067	107	1011	5822	129

Dataset After Augmentation –

Total Images	akiec	bcc	bkl	df	mel	nv	vas
35967	4455	4790	5055	5350	5335	5822	5160

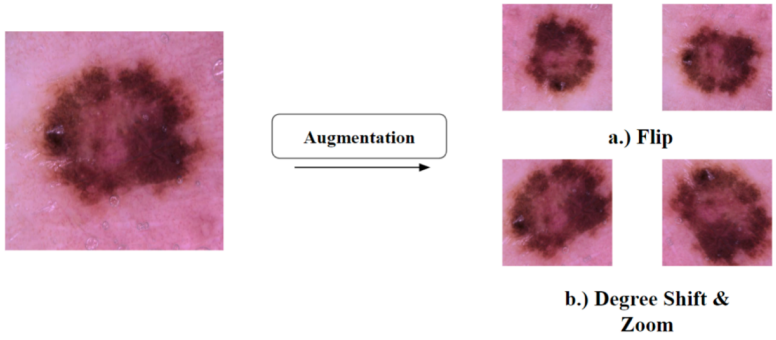


Figure 1: Augmentation

Flip (Mirroring) as seen in Fig 1(a), Flip top and bottom: In this operation, the image will be flipped along the horizontal axis. Probability of 0.5 is used which signifies 50% of the samples will be flipped randomly. Flip left and right: In this operation, the image will be flipped along the vertical axis. Here also probabilty of 0.5 is used for flipping the samples of dataset.

Degree Shift and Zoom is done as seen in Fig 1(b), This will rotate by arbitrary degrees, then a crop is taken from the centre of the newly rotated image. Image is rotated by an arbitrary angle in range -10 to 10 degree with a probability of 0.5 which signifies 50 percent of the sample images from the dataset will be rotated randomly. Zoom effect is not particularly drastic for smaller rotations of between - 10 and 10

1.3 Normalization of Input Images

In image processing, it is recommended to normalize image pixel values relative to the dataset mean and standard deviation. This helps to get consistent results when applying a model to new images and can also be useful for transfer learning. Normalization is done to ensure better performance of the neural network. Normalization is an important step to achieve faster convergence. With unnormalized data, numerical ranges of features may vary strongly. If this raw data is inputted in the transfer learning model, slow convergence will occur. The dataset is normalized in z-score normalization as per given formula:

where, Vi'= z-score normalized values

Vi is the value of row E of i'th column.

$$V_i' = \frac{V_i - \bar{E}}{std(E)} \qquad std\,(E) = \sqrt{\frac{1}{(n-1)} \sum_{i=1}^{n} \left(V_i - E\right)^2}$$

$$E = \frac{1}{(n)} \sum_{i=1}^{n} V_i$$

Figure 2: Formulae used during Normalization

The steepest gradient is searched, which is somewhat in the correct direction but also possesses quite a large oscillation. This can be explained by the value of the chosen learning rate. A relatively large learning rate is required for the features with larger standard deviation since its range is quite large. However, this large learning rate is too large for some smaller parameters. The optimizer overshoots each step, which results in oscillation and hence slow convergence. In this project, for the HAM10000 dataset, the Mean was found to be (0.49139968, 0.48215827, 0.44653124) and standard deviation was found to be (0.24703233, 0.24348505, 0.26158768) for the RGB Channel.

2 System Architecture

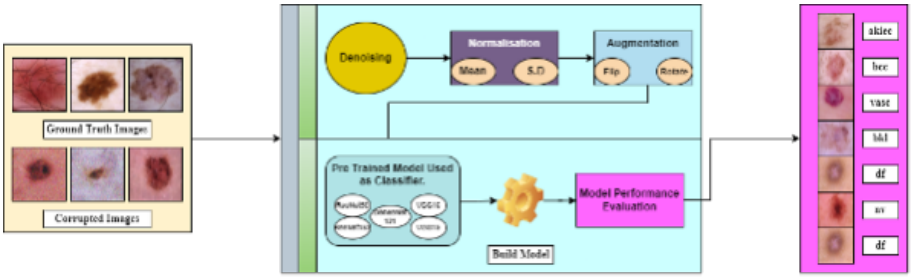


Figure 3: System Architecture

As seen in Fig 3, The benchmark dataset undergoes preprocessing methods such as denoising, normalization and augmentation to get balanced training samples in each class. Some images are corrupt, so denoising process is implemented. Augmentation is done to create balanced classes. Once images are preprocessed, training and testing sets are prepared accordingly and model is fit and used as a classifier.

3 Transformer Models

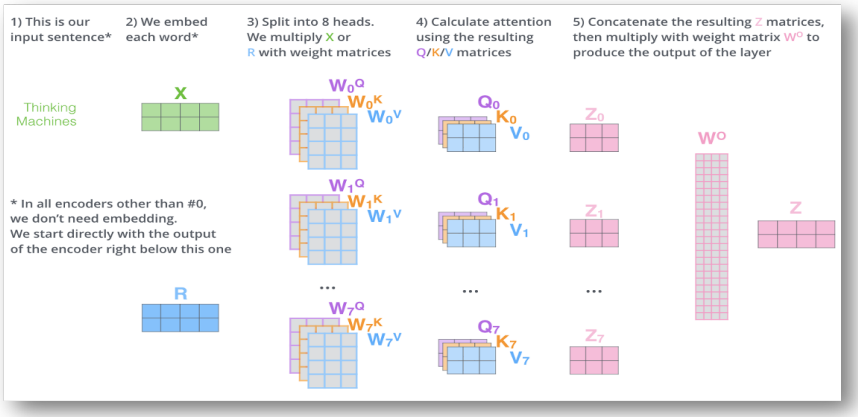
The core idea behind the Transformer models is the attention mechanism introduced in the paper "Attention is All You Need" by Vaswani et al. The attention mechanism allows the model to selectively focus on the relevant parts of the input sequence during training, allowing it to capture long-range dependencies effectively.

At a high level, the following are the core components of the Attention mechanism.

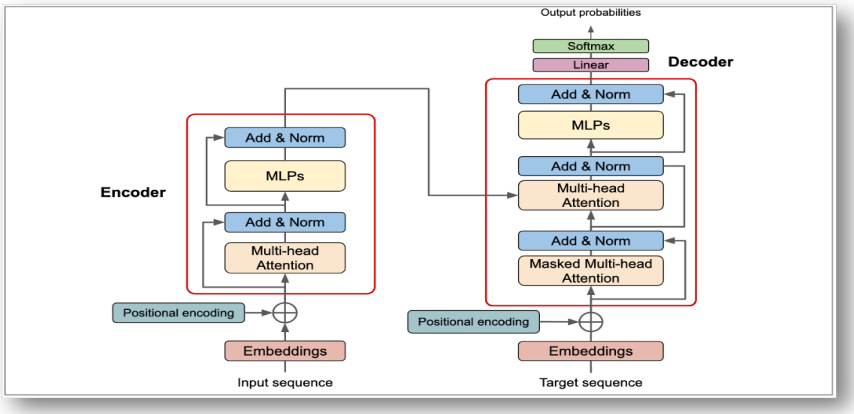
1. Query, Key, and Value Vectors: As shown in the above Figure, there will be three matrices W^Q, W^K, and W^V corresponding to Query , Key, and Value. The input embeddings X_1 and X_2 are multiplied with the matrices to get the respective vectors.

2. Attention Scores: The attention scores are calculated by taking the dot product of the query and key vectors. The output is normalized, and then the softmax operation is performed. Elements with higher weights receive more attention during the computation.

3. Weighted Sum of Values: Finally, the weighted sum of the value vectors is taken where the attention scores are the weights. This is represented by the vector Z, which emphasizes elements more relevant to the current context.



Multi-Head Attention: The above components of the attention mechanism form a single head of the Transformer. In order to learn more complex patterns, the Transformer models often use multiple heads, which involves having multiple Query, Key, and Value matrices. The results are then concatenated and transformed to produce the final output.



Transformer architecture - The above figure shows the transformer architecture. All the Vision Transformers models are encoder models, and the following are the core components of the Transformer Encoder.

1. Embeddings - This represents a numerical representation of the input sequence using some encoding method.

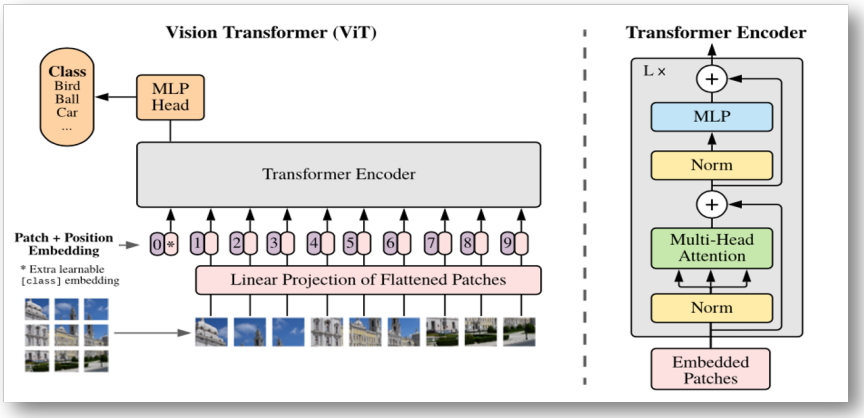
2. Positional Encoding - The Transformer lacks information about the ordering of the input embeddings, and as a result, positional encodings are added to the input embeddings to provide information about the position of each token.

3. Multi-Head Attention: Attention is performed to transform the input into vectors carrying relevant information.

4. Add & Norm: After the self-attention mechanism, a residual connection is applied, allowing the gradients to flow more easily during training, and finally, layer normalization is done.
5. MLP: Following the self-attention mechanism, the encoder applies a position-wise feedforward neural network independently at each position in the sequence. This network consists of fully connected layers and introduces non-linearity to the model.
6. Add & Norm: Similar to the self-attention sub-layer, residual connections and layer normalization are applied after the position-wise feedforward network.

The final output of the encoder is a set of context-aware representations for each token in the input sequence. These representations capture both the local and global contextual information, making them suitable for downstream tasks like machine translation, text summarization, or any other sequence-to-sequence task.

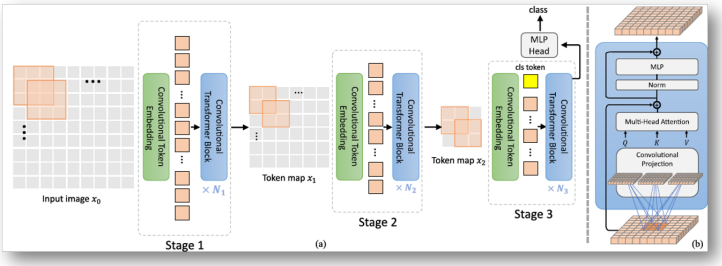
3.1 Vision Transformer (ViT)

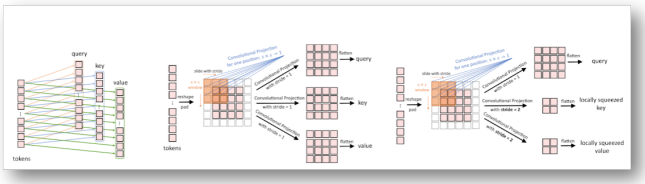


The standard Transformer receives as input a 1D sequence of token embeddings. The cost of applying attention to every pixel is quite high, and as an alternative, the input image is split into patches, and the linear embeddings of these patches are fed as the input to the Transformer. To handle 2D images, we reshape the image $x \epsilon \mathbb{R}^{H \times W \times C}$ into a sequence of flattened 2D patches $x_p \epsilon \mathbb{R}^{N \times (P^2.C)}$, where (H, W) is the resolution of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches, which also serves as the effective input sequence length for the Transformer. Positional encodings of the patches are fed in as input to retain the ordering of the patches.

The Transformer Encoder block architecture remains the same as the original. Similar to the BERT's [class] token, ViT includes a classification token whose output embedding is fed into the classifier to give a probability distribution over all the predicted classes.

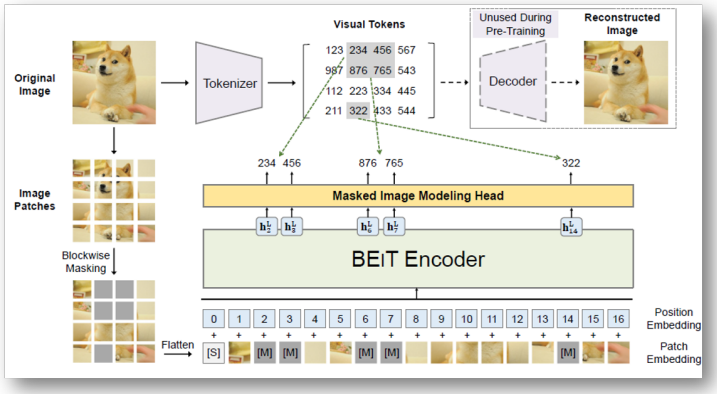
3.2 Convolutional Vision Transformer (CvT)





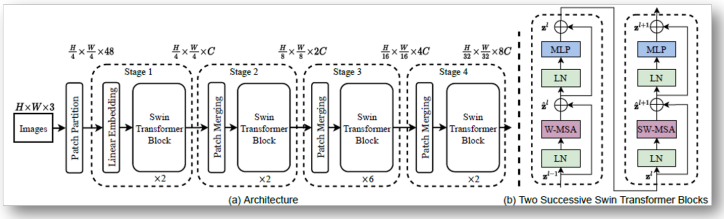
ViT required the model to be trained on large amounts of data to achieve state-of-the-art performance when compared with similarly sized CNN models. CvT aimed to combine the strengths of convolutional neural networks by introducing CNN properties such as shift, scale, and distortion invariance while utilizing the Transformer architecture advantages such as dynamic attention, global context, and better generalization.Similar to Vit, CvT divides the input image into patches, but instead of directly linearly embedding the patches, convolutional operations are performed on the patch embeddings to capture spatial contexts over a multi-stage hierarchy approach, similar to CNNs. Finally, the class token is fed into an MLP to get the predictions.

3.3 Bidirectional Encoder representation from Image Transformers (BEiT)



Inspired by BERT, BEiT extends ViT and uses masked image modeling task to pretrain vision Transformers. Similar to ViT, the image is converted into patches, and a few of the patches are masked/corrupted, and the objective is to predict the masked patches. The main difficulty is that no pre-defined vocabulary exists for the image patches. To alleviate this, before the pre-training task, an image tokenizer is trained using a variational autoencoder. A latent space representation for the image is learned, where the image is represented by discrete visual tokens according to the learned vocabulary. The pre-training task aims to predict the visual tokens of the original image based on the encoding vectors of the corrupted image.Once the BEiT is pre-trained, a simple linear classifier is added as a task layer, and the model is fine-tuned.

3.4 Hierarchical Vision Transformer using Shifted Windows (Swin Transformer)



One of the significant drawbacks of ViT was the quadratic computational complexity associated with high-resolution images. Swin Transformers, an extension of ViT, addressed the issues by introducing two key concepts - hierarchical feature maps and shifted window attention, which made them highly efficient. Swin Transformer constructs a hierarchical representation by starting from small patches and gradually merging neighboring patches in deeper Transformer layers. The linear computational complexity is achieved by computing self-attention locally within non-overlapping windows that partition an image. The number of patches in each window is fixed, and thus the complexity becomes linear to image size. The window-based self-attention module lacks connections across windows, which limits its modeling power. To introduce cross-window connections while maintaining the efficient computation of non-overlapping windows, a shifted window partitioning approach is used. Similar to other ViT models, the final layer is fed into a classifier to make predictions.

4 Results and Performance Analysis

We have used metrics such as Confusion Matrix, Learning curves for loss and accuracy and Accuracy, Precision, Recall and F1 Score to evaluate the prformance of different networks under different conditions. A brief overview of this metrics is stated below.

4.1 Confusion Matrix

Confusion matrix is a very popular measure used while solving classification problems. It can be applied to binary classification as well as for multi-class classification problems. Confusion matrices represent counts from predicted and actual values.

Actual	Negative	Positive
Negative	TN	FP
Positive	FN	TP

Figure 4: Confusion Matrix Classification

In the above figure “TN” stands for True Negative which shows the number of negative examples classified accurately. Similarly, “TP” stands for True Positive which indicates the number of positive examples classified accurately. The term “FP” shows False Positive value, i.e., the number of actual negative examples classified as positive; and “FN” means a False Negative value which is the number of actual positive examples classified as negative.

The confusion matrix for CNN models are as follows:

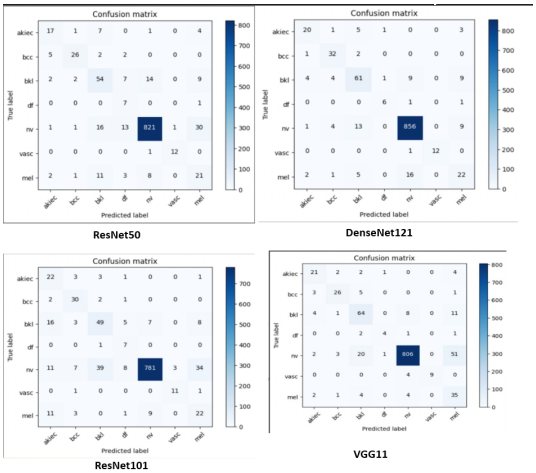


Figure 5: Confusion Matrix of CNN Models

The confusion matrix for transformer models is as follows:

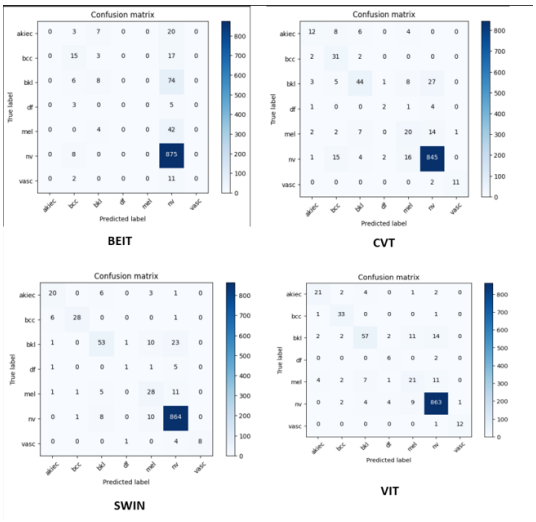


Figure 6: Confusion Matrix of Transformer Models

4.2 Accuracy, Precision, Recall and F1 Score

Performance metrics in the project's classifications report include accuracy, precision, recall, and F1 score, which are calculated on the basis of TP, TN, FP, and FN. TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Figure 7: Metrics

CNN Models	Accuracy
DenseNet121	**91%**
ResNet50	87%
ResNet101	84%
VGG11	87%

Transformer Models	Accuracy
SWIN	**92%**
VIT	92%
BEIT	81%
CVT	87%

Figure 8: Accuracy Comparison

	precision	recall	f1-score
akiec	0.63	0.57	0.60
bcc	0.84	0.74	0.79
bkl	0.60	0.61	0.61
df	0.22	0.88	0.35
nv	0.97	0.93	0.95
vasc	0.92	0.92	0.92
mel	0.32	0.46	0.38
accuracy			0.87
macro avg	0.64	0.73	0.66
weighted avg	0.90	0.87	0.88

ResNet50

	precision	recall	f1-score
akiec	0.71	0.67	0.69
bcc	0.76	0.91	0.83
bkl	0.71	0.69	0.70
df	0.75	0.75	0.75
nv	0.97	0.97	0.97
vasc	1.00	0.92	0.96
mel	0.50	0.48	0.49
accuracy			0.91
macro avg	0.77	0.77	0.77
weighted avg	0.91	0.91	0.91

DenseNet121

	precision	recall	f1-score
akiec	0.35	0.73	0.48
bcc	0.64	0.86	0.73
bkl	0.52	0.56	0.54
df	0.30	0.88	0.45
nv	0.98	0.88	0.93
vasc	0.79	0.85	0.81
mel	0.33	0.48	0.39
accuracy			0.84
macro avg	0.56	0.75	0.62
weighted avg	0.88	0.84	0.85

ResNet101

	precision	recall	f1-score
akiec	0.66	0.70	0.68
bcc	0.79	0.74	0.76
bkl	0.66	0.73	0.69
df	0.67	0.50	0.57
nv	0.98	0.91	0.94
vasc	1.00	0.69	0.82
mel	0.34	0.76	0.47
accuracy			0.87
macro avg	0.73	0.72	0.71
weighted avg	0.91	0.87	0.89

VGG11

Figure 9: Classification Report for CNN Models

	precision	recall	f1-score
akiec	0.00	0.00	0.00
bcc	0.41	0.43	0.42
bkl	0.36	0.09	0.15
df	0.00	0.00	0.00
mel	0.00	0.00	0.00
nv	0.84	0.99	0.91
vasc	0.00	0.00	0.00
accuracy			0.81
macro avg	0.23	0.22	0.21
weighted avg	0.71	0.81	0.75

BEiT

	precision	recall	f1-score
akiec	0.57	0.40	0.47
bcc	0.51	0.89	0.65
bkl	0.70	0.50	0.58
df	0.40	0.25	0.31
mel	0.41	0.43	0.42
nv	0.95	0.96	0.95
vasc	0.92	0.85	0.88
accuracy			0.87
macro avg	0.64	0.61	0.61
weighted avg	0.88	0.87	0.87

CvT

	precision	recall	f1-score
akiec	0.88	0.70	0.78
bcc	0.67	0.97	0.79
bkl	0.83	0.70	0.76
df	0.42	0.62	0.50
mel	0.60	0.46	0.52
nv	0.96	0.98	0.97
vasc	1.00	0.54	0.70
accuracy			0.92
macro avg	0.76	0.71	0.72
weighted avg	0.92	0.92	0.91

SWIN

	precision	recall	f1-score
akiec	0.75	0.70	0.72
bcc	0.80	0.94	0.87
bkl	0.79	0.65	0.71
df	0.46	0.75	0.57
mel	0.50	0.46	0.48
nv	0.97	0.98	0.97
vasc	0.92	0.92	0.92
accuracy			0.92
macro avg	0.74	0.77	0.75
weighted avg	0.92	0.92	0.92

VIT

Figure 10: Classification Report for Transformer Models

5 Conclusion

The project's main objective was to compare how the modern state-of-the-art Vision Transformer models compared against the CNN architecture models, specifically in the context of the image classification task utilizing the HAM10000 dataset.

DenseNet121 emerged as the most proficient among the CNN models, achieving a notable accuracy of 91%. **Conversely, both the base Vision Transformer (ViT) and SWIN Transformer surpassed all CNN models, exhibiting a superior accuracy of 92%.**

This outcome underscores the capability of Transformers to capture comprehensive global context information by concurrently considering relationships across all positions within the input sequence. In image classification, comprehending the entire contextual information within an image is imperative for precise predictions, a task for which Transformers prove well-suited. Additionally, they demonstrate increased parameter efficiency, derive advantages from pre-training on extensive datasets, and adaptability to diverse resolutions without reliance on fixed-size convolutional filters.

The ongoing, active research in Vision Transformers suggests potential avenues for future exploration. Subsequent work could involve investigating and refining novel architectures to enhance the performance of Vision Transformers further.

6 References

1. Wu, Haiping, et al. "Cvt: Introducing convolutions to vision transformers." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

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

3. Bao, Hangbo, et al. "Beit: Bert pre-training of image transformers." arXiv preprint arXiv:2106.08254 (2021).

4. Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

5. U˘gur Fidan, ˙Ismail Sarı, and Raziye K¨ubra Kumrular. Classification of skin lesions using ann. In 2016 Medical Technologies National Congress (TIPTEKNO), pages 1–4. IEEE, 2016.

6. Mobeen ur Rehman, Sharzil Haris Khan, SM Danish Rizvi, Zeeshan Abbas, and Adil Zafar. Classification of skin lesion by interference of segmentation and convolotion neural network. In 2018 2nd International Conference on Engineering Innovation (ICEI), pages 81–85. IEEE, 2018.

7. M Monisha, Alex Suresh, BR Tapas Bapu, and MR Rashmi. Classification of malignant melanoma and benign skin lesion by using back propagation neural network and abcd rule. Cluster Computing, 22(5):12897–12907, 2019.

8. Marwan Ali Albahar. Skin lesion classification using convolutional neural network with novel regularizer. IEEE Access, 7:38306–38313, 2019.

9. Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. nature, 542(7639):115–118, 2017.

10. Fangfang Han, Huafeng Wang, Guopeng Zhang, Hao Han, Bowen Song, Lihong Li, William Moore, Hongbing Lu, Hong Zhao, and Zhengrong Liang. Texture feature analysis for computer-aided diagnosis on pulmonary nodules. Journal of digital imaging, 28(1):99–115, 2015.