Image Classification: Transformer models vs BrandImageNet

Libera Università degli Studi Guido Carli

Department: Impresa e Management,

Business and Marketing Analytics

Supervisor: **Candidate:**

Francisco Villarroel Alberto de Leo

Anno Accademico 2023/2024

Table of Contents

1. Introduction

The rapid evolution of technology has profoundly impacted various sectors, including marketing, where visual content plays a pivotal role in shaping brand perceptions. Research in the domain of computer vision and deep learning for highly performant models, surpassing levels of interpretability, would help in proper image classification. The comparative abilities of Transformer models are explored through the BrandImageNet framework for the classification of images in the domain of visual marketing (Liu et al., 2020).

One of the primary reasons for conducting such a study is the massive impact that the use of visual content, increased with the advent of image-based social media, creates in the way it organizes perceptions of consumers regarding a brand. The use of more precise model to analyse and classify images can aim business giving important insight on how the brand is perceived by the consumer and on the brand's portrayal on social media platforms. Nowadays visual marketing plays a pivotal role in the strategies of businesses. With the use of visual content, brands can communicate messages, engage consumers and influence their spending behaviour. This is possible due to the increasing use and power that social media have in the life of every person. Every day we are exposed to an increasing number of ads and other types of graphic content, so the understanding of how the brand is perceived by users becomes crucial.

The model that we use as a benchmark is the BrandImageNet. It is a multi-label deep convolutional neural network model for predicting the perceptual brand attributes from the consumer-generated images shared on social media. The model will support firms in understanding the portrayal of their brands through attributes such as "glamorous", "rugged", "healthy" or "fun". In this way, they will track current brand portrayal and get precise insights into consumer perception and attitudes. Despite its innovative approach, BrandImageNet has a few important weaknesses. The main limitations concerning the model are the computational expense and complexity of training it. Furthermore, CNN models have been prohibitive for any smaller firm or individual researcher, without large resources. These reasons underline the need of better scalable and efficient models.

A great advance in the natural language processing (NLP) field is made by the Transformer-based model. This advance is enabled thanks to their particular architecture,

specially by their self-attention mechanism, which gives the possibility to efficiently capture dependencies across two elements in a sequence, regardless of their range. That innovation allows the parallel processing of input, facilitating quicker training times and better performance compared to RNNs and CNNs. The use of Transformer models has been expanded by researchers who have hypothesized that the self-attention mechanism inherent in Transformers could also be advantageous for image classification tasks. To implement Transformers for image classification task, some adjustments have been made to their architecture. After the adaptation was introduced the Vision Transformer that outperformed all the state-of-the-art models with CNNs and RNNs in image classification. Its ability to pre-train on huge datasets and then fine-tune on smaller, task-specific datasets guarantees strong performance in different contexts.

To overcome the limitations of traditional CNNs, various Transformer models have been developed for computer vision tasks. Architecture in any model comes up with addressing the challenges of image processing in detail. Some of the key models include Pyramid Vision Transformer (PvT), Convolutional Vision Transformer (CvT), Vision Transformer (ViT), and Swin Transformer. Each model has peculiar characteristics that make it specific for a different type of task:

- Pyramid Vision Transformer (PvT): by using the pyramid architecture, it builds multi-scale feature maps that outperform in tasks, especially in detection and segmentation.
- Convolutional Vision Transformer (CvT): this introduces convolutions into the ViT model to capture local details and deal more reasonably with computational constraints.
- Vision Transformer (ViT): it breaks the input images into fixed-size patches and processes them through a series of Transformer blocks using global self-attention mechanisms to classify the images.
- Swin Transformer: this is the hierarchical design using shifted windows for local self-attention. It allows efficient applications for object detection and image segmentation capabilities by reducing computational complexity.

The main aim of this research is to compare the performances of Vision Transformers and Swin Transformers with BrandImageNet in classifying pictures for visual marketing,

highlighting the potential of advanced Transformers. To do so, these models are trained on a dataset of images that are labelled according to brand attributes, and comparing them according to accuracy, specificity, and overall model performance. By overcoming the limitations of traditional CNNs and utilizing the advantages of Transformers, this research aims to develop more effective and scalable models for analysing brand perception through social media imagery.

2. Related Literature and Transformer for Images

The main stream research of the paper concerns the impact that visual marketing could have on brand perception. This is enabled using computer vision and deep learning models, specifically *Transformer models* and *BrandImageNet.*

Transformers have become the preferred models for performing Natural Language Processing (NLP) tasks by effectively capturing long-range dependencies through self-attention mechanisms (Vaswani et al. 2017). Unlike recurrent neural networks (RNNs) and convolutional neural networks (CNNs), transformers' advantage is due to the parallel process of input data that enables faster training times and better performance on large datasets. They offer the possibility to be adapted to different tasks and computational efficiency, allowing models to be trained with more than a hundred billion parameters without saturating model performance. Inspired by the success of the transformers applied to NLP and assuming that the self-attention mechanism could also be beneficial for image classification tasks, it was proposed to use the same architecture, with few modifications, to perform image classification (Dosovitskiy et al. 2020). The properties of transformers are being used also in this context. In fact, Transformers model can be pre-trained on a large dataset and then used in a smaller dataset. This allows us to outperform state-of-the-art of the base CNN and RNN on image classification.

2.1 BrandImageNet

BrandImageNet is a model developed to help firms monitor their brand portrayal on social media by mapping images to specific perceptual attributes. This model allows firms to measure how their brands are perceived along various attributes such as "glamorous", "rugged", “healthy” and “fun”. For instance, if consumer images tagged with a fashion

brand frequently score high for "glamorous" in the BrandImageNet model, it indicates that consumers perceive the brand as glamorous. Conversely, images tagged with an outdoor apparel brand might frequently score high for "rugged," aligning with a different brand identity. The practical application of this technology is significant for brand management, allowing companies to see how their branding efforts are perceived and potentially adjust their strategies based on real-time data from social media imagery (Liu et al., 2020).

2.1.1 Model architecture

It was developed using a multi-label convolutional neural network (ConvNet), which was fine-tuned from the Berkeley Vision and Learning Center (BVLC) Reference CaffeNet model. To be more precise, the BrandImageNet model has four key attributes: glamorous, rugged, healthy, and fun. These attributes were chosen as they are relevant to both the apparel and beverage sectors, for which the model is targeted, for a meaningful differentiation of brands and were available attributes from Y&R's BAV (BrandAssetValuator) consumer-brand-perception survey.

The BrandImageNet model begins by resizing images to 227x227 pixels with three colour channels (RGB), aligning with the input dimensions required by the pre-trained model used for initialization. This preprocessing step ensures compatibility with the subsequent layers. For feature extraction, the model employs five convolutional layers. These layers apply various filters to detect essential features such as edges, textures, and shapes within the images. To introduce non-linearity, ReLU (Rectified Linear Unit) activation functions are interspersed between the convolutional layers. After specific convolutional layers, max-pooling layers are incorporated to reduce the spatial dimensions of the feature maps. This reduction helps control overfitting and decreases the computational load. Once feature extraction is complete, the resulting feature maps are flattened and passed through three fully connected layers. These layers learn higher-level representations of the input data by combining features detected in the earlier layers. The fully connected layers enhance the model's ability to make accurate predictions.

Finally, the output layer uses sigmoid activation functions such that the architecture can predict by generating probabilities corresponding to each brand attribute. This design

will let the model easily take care of multi-label classification since all of the attributes will be taken separately to predict multiple labels for a single image.

2.1.2 BrandImageNet Limitations

BrandImageNet has some significant limitations, especially when compared to transformer-based models. Identifying brand attributes in images is a quite subjective task, which means different people might have different opinions about what attributes are the right ones for the specific image. This subjectivity can cause inconsistencies in the data labels, which makes it a problem for the model. (Liu et al., 2020).

Scalability and efficiency are additional issues, since, nonetheless their great power, they can be computationally expensive to train and scale. Deep networks with many layers require substantial computational resources, making the training process time-consuming and costly. This can hinder the scalability of the model for larger datasets or more complex tasks (Goodfellow, Bengio, & Courville, 2016). Finally, the complexity and resource requirements of implementing and fine-tuning deep CNN models require expertise in deep learning and access to substantial computational resources. This complexity can limit the accessibility of the model for smaller firms or individual researchers, presenting a barrier to its broader applicability (LeCun, Bengio, & Hinton, 2015).

2.2 Different Transformer models

Until now, the management of self-attention has been a challenge while developing the transformer models under computer vision. Various transformer models have evolved to support the special challenges under tasks of image processing that keep changing quickly. These models in the computer vision domain use the self-attention mechanism developed for the natural language processing domain to improve their performance over tasks like image classification, object detection, and segmentation. This section further explores some of the leading transformer models in this domain: Pyramid Vision Transformer (PvT), Convolutional Vision Transformer (CvT), Vision Transformer (ViT), and Swin Transformer. In so doing, they help us find the architectural innovations in processing visual data and the certain advantages they have.

2.2.1 Pyramid Vision Transformer

Pyramid Vision Transformer (PvT) (Wang et al. 2021) generates multi-scale feature maps through a pyramid structure. This architecture is particularly effective for tasks that require multi-scale representations, enhancing performance in object detection and segmentation. PVT is an efficient model similar to a Vision Transformer but equipped with a pyramid structure, making it very efficient for tasks requiring detailed predictions. This design is particularly useful for detailed prediction tasks because it allows the model to handle more precise inputs effectively. The processing of deeper layers by the model leads to shorter input sequences and a corresponding reduction in computational requirements. Additionally, it used a special layer called the "spatial-reduction attention layer" to minimize the resources needed when dealing with high-resolution features.

2.2.2 Convolutional Vision Transformer

Convolutional Vision Transformer (CvT) (Wu et al. 2021) design introduces convolutions into the ViT architecture. This design features a hierarchical structure where each stage starts with a convolutional embedding that reshapes and processes token sequences. This method captures local information and reduces sequence length while increasing token feature dimensions, similar to CNNs. Additionally, CvT replaces the linear projection in self-attention blocks with a convolutional projection, enhancing local spatial context and managing computational complexity. With this approach, there's an improvement in efficiency with a small impact on performance.

2.2.3 Vision transformer

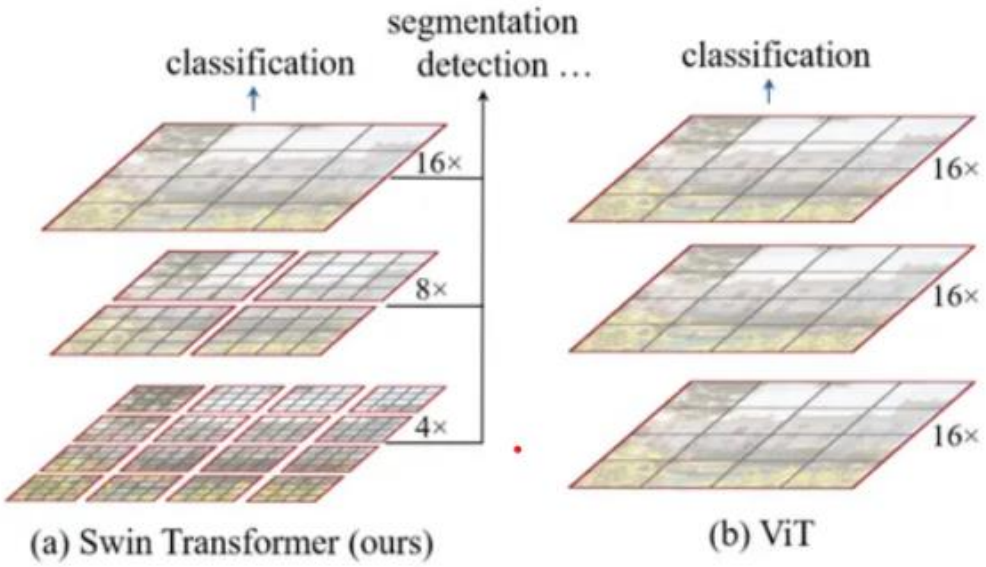
The *Vision Transformer (ViT)* (Han et al.,2020) architecture utilizes self-attention mechanisms for image processing. It begins by dividing an image into fixed-size patches, each of which is embedded into a high-dimensional vector. These embeddings are then input into a series of transformer blocks that comprise a multi-head self-attention layer and a feed-forward layer. The self-attention layer calculates attention weights for each pixel by examining its relationship with all other pixels in the image. More specifically, this process gives the possibility for the model to focus on different parts of the input sequence simultaneously. The feed-forward layer applies a non-linear transformation to

the output from the self-attention layer, enhancing the representation. The multi-head attention mechanism allows the model to attend to various segments of the image simultaneously, providing a comprehensive understanding of the image's content. Furthermore, the ViT architecture includes a patch embedding layer that divides the image into smaller patches, mapping each one to a high-dimensional vector. These patch embeddings are then processed through the transformer blocks. The final output of the ViT architecture is a class prediction, obtained by passing the output of the last transformer block through a classification head, which typically consists of a single fully connected layer. This approach allows the Vision Transformer (ViT) to use its self-attention mechanism effectively for image classification, achieving top performance on many benchmark tests.

2.2.4 Swin Transformer

The *Swin Transformer* (Liu et al., 2021) is a variant of the Vision Transformer that builds a feature map hierarchy by re-arranging image patches in deeper layers. Moreover, it achieves linear computational complexity concerning input image size by carrying out self-attention only within each local window. This special structure of the model makes it versatile and suitable for image classification and tasks that require detailed recognition. The main difference with the original ViT is that the former one produces feature maps of a single low resolution and with quadratic computation complexity to input image size due to computation of self-attention globally, while the Swin Transformer hierarchical approach produces multi-scale feature maps that make it notably effective for tasks such as object detection and image segmentation. Swin transformer development is a big step for Transformer-based models, offering a scalable and efficient approach to image processing tasks. Its peculiar combination of local attention mechanisms with a hierarchical architecture provides a flexible framework to achieve state-of-the-art results across several computer vision applications.

Figure 1 Swin transformer vs Vision Transformer architecture



2.3 Transformer Application

Transformers have been adapted to process image data, offering new capabilities in computer vision. With them it's possible to perform various visual tasks:

- Image Classification: Vision Transformers (ViT) and its variants have achieved great performance such as those of ImageNet.
- Object Detection: The DEtection TRansformer (DETR) (Carion et al. 2020) offers a novel approach to object detection, using a simpler process and increasing detection accuracy.
- Image Segmentation: recently researchers have extended transformers to medical image segmentation tasks, resulting in good models. (Pu et al. 2024)

Transformer models have significantly affected the performance on image processing tasks. This is made possible due to their new architecture and methods that outperform the classic CNN and RNN not only in performance but also in efficiency. In this paper, we focus on the image classification task by using two types of models: ViT and Swin Transformer.

Table 1: Transformer Characteristic comparison

Feature	Convolutional Vision Transformer	Pyramid Vision Transformer	Vision Transformer	Swin Transformer
Architecture	Combines convolutional layers with transformers for local and global feature extraction	Uses a pyramid structure with progressive resizing and down sampling of features	Uses transformer blocks directly on image patches with global self-attention	Hierarchical structure with shifted windows for local self-attention
Attention Mechanism	Convolutional projections before self-attention layers	Spatial-reduction attention layers to manage high-resolution features	Global self-attention over all patches	Local self-attention within shifted windows
Image Representation	Maintains spatial structure via convolutional layers	Multi-scale feature maps with spatial down sampling	Image divided into fixed-size patches	Hierarchical representation captures both local and global contexts
Training Efficiency	Improved efficiency due to convolutional layers reducing sequence length	Reduced computational cost with spatial-reduction attention layers	Requires large datasets and computational resources for optimal performance	More efficient training with local attention and shifted window mechanism
Performance	Good performance on both local and global feature extraction tasks	Effective for prediction tasks (e.g., object detection, segmentation)	State-of-the-art performance on image classification benchmarks like ImageNet	High performance on image classification, object detection, and segmentation
Applications	Suitable for tasks requiring both local and global feature extraction	Dense prediction tasks, such as object detection and segmentation	Primarily image classification; can be fine-tuned for other tasks	Versatile for image classification, object detection, and segmentation
Limitations	Still computationally intensive, complexity in combining CNN and transformer layers	Can be complex due to multi-scale and spatial-reduction mechanisms	High computational cost, requires large datasets	Complexity in window shifting mechanism, still computationally demanding

3. Empirical Study (Data, Model, Results)

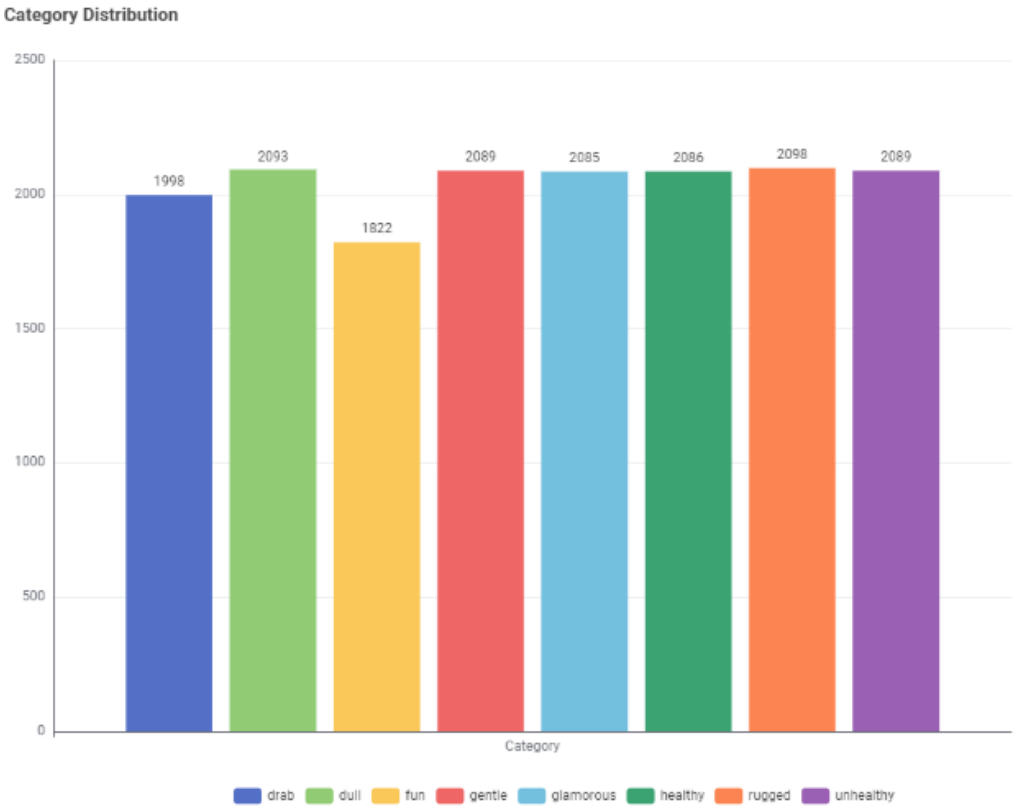
3.1 Data

To train a multi-label image-classification model of brand attributes we need an annotated data set consisting of images labelled with the respective attribute that they represent. Since we are not performing an object-detection task, which is typically trained on large public data sets of labelled images, there's no existing data set that is annotated with brand attributes for each image. To have a fair comparison between the two transformer models and the BrandImageNet, I use the same images (Liu et al., 2020) that have been used to implement the last cited model.

To develop models, researchers created an annotated dataset using images from Flickr, a popular online photo-sharing platform. Flickr has been used in numerous research in visual and social networks (e.g. McAuley and Leskovec, 2012) because it allows users to share photos with titles, descriptions and free from tags. Another advantage that Flickr has compared to other social networks is its advanced search engine, which utilizes user-provided text labels, image content, and clickstream data (Stadlen 2015), which made Flickr an ideal source for gathering relevant images with specific requisites. The search engine's ability to rank images based on a large user consensus helps ensure that the top search results strongly associate with the search terms.

For each brand attribute (glamorous, rugged, fun, healthy) the researchers queried the relevant term on Flickr and collected approximately 2,000 images from the top search results as positive examples (i.e., images depicting the attribute). They also collected negative examples (i.e., images not depicting the attribute) by querying antonyms of each attribute (e.g., "drab" for "glamorous," "gentle" for "rugged," "unhealthy" for "healthy," and "dull" for "fun") and again gathered about 2,000 top-ranked images.

Figure 2 Attributes Category distribution



Additionally, images collected for other attributes and their antonyms were used as negative examples, provided they were not already included as positives for the current attribute. For instance, for the attribute "healthy," they used "healthy" to collect positive examples and "unhealthy" along with other unrelated terms like "glamorous," "rugged," "fun," "drab," "gentle," and "dull" for negative examples. This method ensured a comprehensive set of negative instances. In total, the annotated dataset consisted of 16,360 images, each labelled for brand attributes.

3.2 Vision Transformer Model

The Vision Transformer (ViT) applies the transformer architecture, originally designed for natural language processing, to image classification tasks. The input image is divided into N patches, each of size $P \times P$ pixels. If the image size is $H \times W$ (height × width), the number of patches N is calculated as:

$$N = \frac{H \times W}{P^2}$$

(e.g. the input image is divided into fixed-size patches, typically 16×16 pixels. For an image sized 224×224, this results in 196 patches).

Each patch is then flattened into a vector and linearly projected into a higher-dimensional embedding space of size D ($P \times P \times C$ where C is the number of channels, typically 3 for RGB images). For a 16×16 RGB patch, this would be a vector of length 768 (16 * 16 * 3). These vectors are then projected into a higher-dimensional space, using a linear transformation that generates an embedding vector for each patch. Since transformers lose the spatial knowledge of an image, positional encodings of patch embeddings are added to the patch embeddings. This information helps the model understand the order and relative positioning of those patches within the original image.

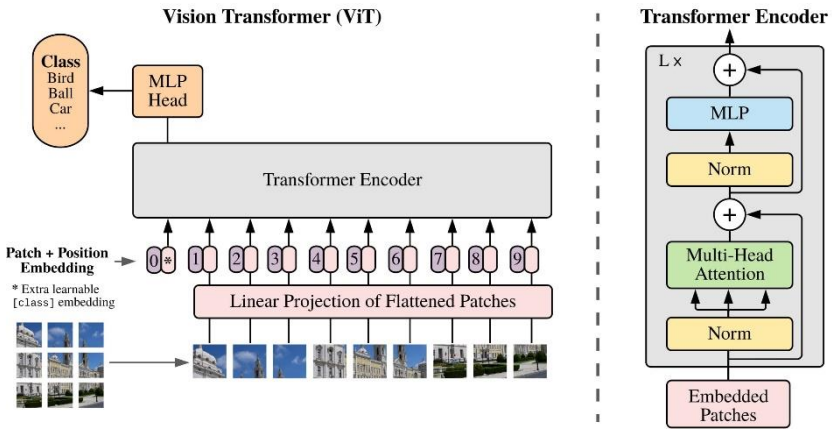
The transformer encoder further proceeds from the sequence of patch embeddings with positional encodings included. The encoder consists of multiple transformer blocks, where each block contains a multi-head self-attention mechanism and a feed-forward neural network. This part in each transformer block allows the model to address different parts of the images simultaneously.

Self-attention calculates attention scores between all pairs of patches, allowing the network to evaluate the attention that needs to be given to each patch with respect to other patches. This involves calculating query (Q), key (K), and value (V) matrices. The attention score for each pair of patches is computed using the dot product of the query and key vectors, then scales and applies a soft-max to get the attention weights, which are finally used to create a weighted sum of value vectors.

After the self-attention layer, a feed-forward neural network is applied element-wise to each of the patch embeddings by passing them concentratedly through two linear transformations with the application of a ReLU activation function in between. Each self-attention and feed-forward network is followed by layer normalization and residual connections. The main purpose of the residual connection is to avoid vanishing gradients and improve the gradient flow during optimization. After these steps, there's the transformer encoder. The transformer encoder is essentially a stack of these blocks, which process embeddings from the previous block, allowing the model to learn increasingly complex representations. A special classification token (CLS token) is added to the beginning of the sequence of patch embeddings at the input stage. This token aggregates information from all patches through the self-attention mechanism. The final embedding

of the CLS token, after passing through all the transformer blocks, is used as the image representation for classification. This last portion is then passed through a fully connected (dense) layer to get output class probabilities. Usually, ViT models are pre-trained on huge datasets to learn robust features. Pretraining helps the model capture general visual features useful for a wide range of tasks. Then the model can be fine-tuned on a specific dataset, which is smaller and more specialized than the dataset used for pretraining. This fine-tuning process allows the model to adjust to the specific details of the classification task, leading to improved accuracy and performance. Utilizing the capabilities of the transformer architecture, ViT models attain state-of-the-art results in image classification tasks, showcasing their robustness and versatility over a wide range of datasets and applications. For the image classification task performed, the model has been pre-trained with "google/vit-base-patch16-224" checkpoints. Vision Transformer (ViT) with the aforementioned checkpoint, is pre-trained on ImageNet-21k (14 million images, 21,843 classes) at resolution 224x224, and fine-tuned on ImageNet (1 million images, 1,000 classes) at resolution 224x224.

Figure 3 Vision Transformer architecture



3.3 Swin Transformer Model

The Swin Transformer (Shifted Window Transformer) is a hierarchical vision transformer which processes images in a completely new way with the help of shifted windows. The process begins by partitioning the input image into non-overlapping patches and embedding it into a higher-dimensional space, after which the resulting

patches are flattened and projecting it using a trainable linear layer, resulting in patch embeddings.

The model employs a series of patch merging layers, which progressively merge adjacent patches to reduce spatial dimensions while increasing feature dimensions. This hierarchical structure creates representations similar to the multi-scale feature maps in convolutional neural networks (CNNs). Unlike the Vision Transformer (ViT), which uses global self-attention, the Swin Transformer adopts locally self-attention in non-overlapping windows, which attends to only a small part of the image in one step for high efficiency. In each stage, the input is partitioned into non-overlapping windows, and self-attention computation is operated within each window independently. This reduces computational complexity from quadratic to linear with respect to the input size, making the model more efficient. To facilitate cross-window connections and improve the modelling of relationships between distant patches, the windows are shifted in subsequent layers. For instance, the first layer of windows will cover certain areas, and the next layer will shift these windows by a fixed number of pixels to achieve an overlap and interaction between neighbouring patches.

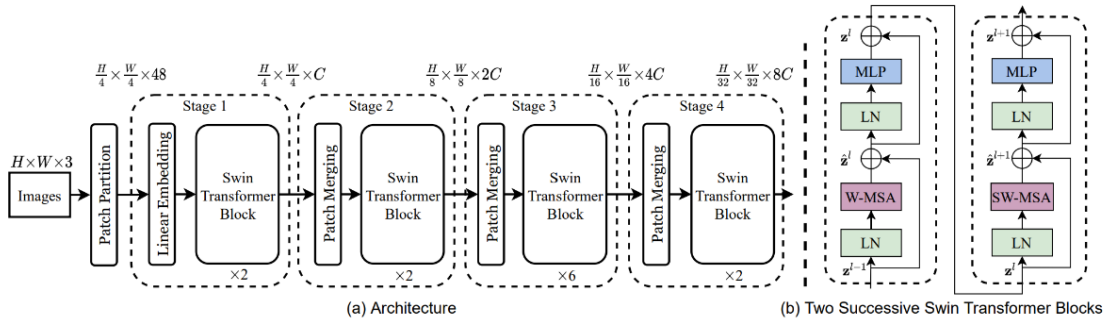
Inside each window, we compute multi-head self-attention: first, for the patches inside the window, we compute three matrices—query (Q), key (K), and value (V)—and then attention scores are computed on them to determine how important each patch is with respect to the other patches. After the self-attention layer, the output passes through a feed-forward network, consisting of two linear layers with a ReLU activation function in between. This network adds non-linearity and further processes the features. Each self-attention and feed-forward block is followed by layer normalization and residual connections, which help stabilize training and improve convergence.

For classification tasks, the final output features are pooled using global average pooling to create a single feature vector for the entire image. This feature vector is then passed through a fully connected layer to obtain the final class probabilities. Swin Transformer hierarchically represents the input image and combines the shifted window mechanism to achieve high efficiency and performance in different computer vision tasks.

For the image classification task implemented in this paper, the “swin-base-patch4-window7-224” checkpoint model pretrained on large datasets is used to learn robust features. This pretraining helps the model capture general visual features, which can then

be fine-tuned on specific tasks with smaller datasets, ensuring high accuracy and performance.

Figure 4: Swin Transformer architecture



3.4 Models train

After importing the images and pre-processing them as mentioned in sections 3.2 and 3.3 to apply the ViT and Swin Transformer, the training part of the two models starts. To train the model effectively, begin by defining the loss function and selecting an optimizer.

The loss function measures the discrepancy between the model's predictions and the actual labels, guiding the optimization process. For classification tasks, Cross-Entropy Loss is commonly used due to its effectiveness in handling probabilistic outputs. This loss function increases as the predicted probability diverges from the actual label, providing a clear signal for the model to adjust its parameters and improve accuracy (Goodfellow, Bengio, & Courville, 2016).

Once the loss function is defined, an optimizer to update the model parameters during training is chosen. The Adam optimizer is a popular choice because it adapts the learning rate for each parameter based on the first and second moments of the gradients. This adaptive approach combines the benefits of AdaGrad and RMSProp, leading to efficient and robust performance across various problems (Kingma & Ba, 2014). The optimizer's role is crucial as it determines how the model's parameters are adjusted in response to the computed gradients, aiming to minimize the loss function.

In order to improve the robustness of the model and to prevent overfitting, The dropout regularization is used. It is a popular regularization technique used to prevent overfitting in neural networks. During the training process, dropout works by randomly

setting a fraction of the input units to zero at each update cycle. This ensures that the model does not depend too much on some units by randomly deactivating a proportion of the input units to zero whenever a model is being updated. In other words, it basically forces more general features so that a model does not rely heavily on one predetermined feature. The dropout rate determines the fraction of units to drop; for example, a dropout rate of 0.1 means that 10% of the units will be set to zero during each forward pass (Srivastava et al., 2014). For purposes of the Swin Transformer model, dropout is applied over hidden layers and attention mechanisms. By using the dropout, the model can prevent overfitting and improve generalization.

The training process involves a structured loop where the model learns from the data iteratively. In each epoch, which constitutes a complete pass through the training dataset, the model processes batches of images. During this phase, the input data is passed through the model to generate predictions. The loss is then computed by comparing these predictions to the true labels using the pre-defined Cross-Entropy Loss function. With the gradients computed, the optimizer updates the model parameters to minimize the loss. Adam, in particular, adjusts the learning rate for each parameter individually, which can lead to faster convergence and better overall performance (Kingma & Ba, 2014). Tracking the loss after each batch helps monitor training progress and ensures that the model is learning effectively. If the loss decreases consistently, it indicates that the model is improving its predictions.

Adjustments to the learning rate, batch size, or other hyperparameters may be necessary based on the results (Bishop, 2006). For this purpose, a grid search to find the best hyperparameters is performed, finding that the best parameters for the models are:

- Epochs: 6,
- Batch size: 12,
- Learning rate: 0.001.

A grid search is also performed to choose between the two models, Vision Transformer and Swin Transformer. With the above hyperparameters, the best performing model is the Swin transformer, so the following discussion of the results is made using the last cited model. This iterative approach ensures that the model learns effectively from the data,

optimizing its parameters to improve predictions and generalize well to new, unseen data (Aggarwal, 2023).

3.5 Results

After training the Swin Transformer model, it is crucial to analyse and discuss the results to point out its performance, strengths, and weaknesses and further propose any improvement areas. Key evaluation metrics such as accuracy, specificity, AUC and ROC are used to assess the performance of the model. With these evaluation metrics we provide a global perspective of the model's effectiveness in the classification task. High accuracy indicates that the model is correctly classifying a large portion of the test samples, while specificity evaluates the ability of a test to recognize properly the negative cases and exclude them without falsely identifying them.

3.5.1 ROC and AUC

The two most common performance measurements for classification problems are the AUC and ROC curve. The ROC (Receiver Operating Characteristics) curve is the graphical representation of the effectiveness of the binary classification model. This provides a description of the true positive rate versus the false positive rate at different classification thresholds. AUC (Area Under the Curve), which represents the area under the ROC curve, measures the overall performance of the binary classification model. As both true positive rate and false positive rate range between 0 to 1, the area will always lie between 0 and 1, and greater value of AUC denotes better model performance. The AUC measures the probability that the model will assign a randomly chosen positive instance a higher predicted probability compared to a randomly chosen negative instance. Since the model is performing a multi-label task, the ROC and AUC represent whether an image belongs to a specific category or not.

The ROC curve for the "Fun" class indicates moderate performance with an AUC of 0.84. This curve suggests that the model can reasonably differentiate the "Fun" class from other classes, although this is the lowest value of the AUC, suggesting a potential for enhancement.

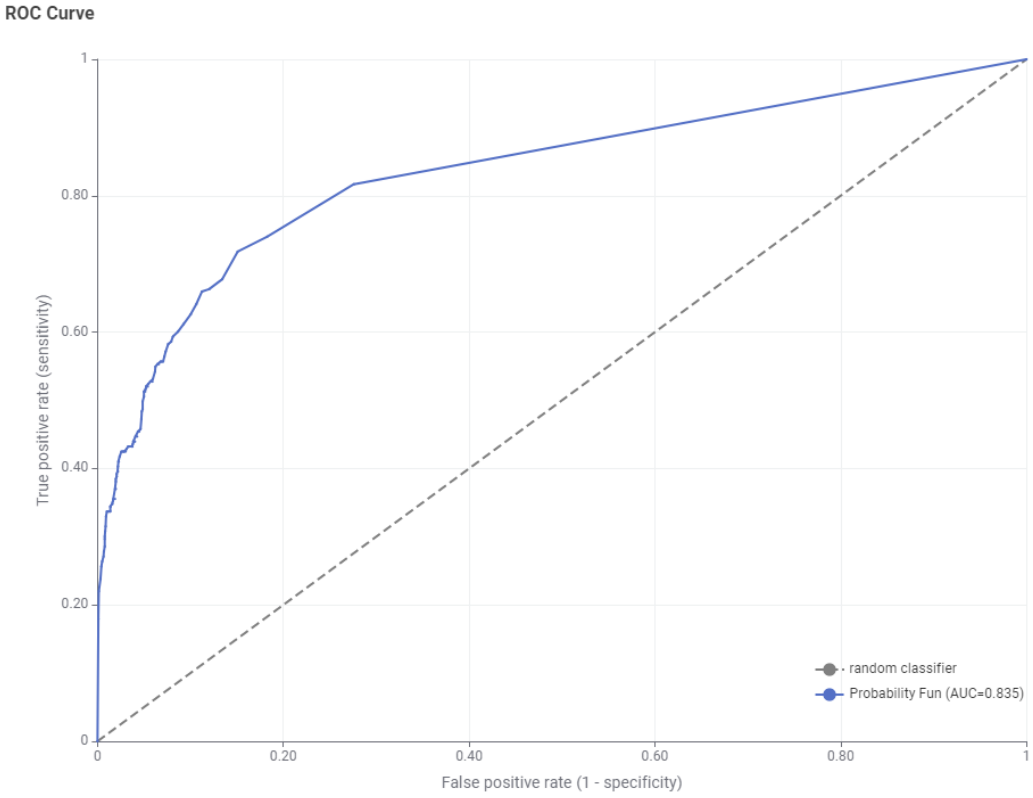
The ROC curve for the "glamorous" class demonstrates excellent performance, with an AUC of 0.85. This high AUC attests that the model is very proficient at distinguishing

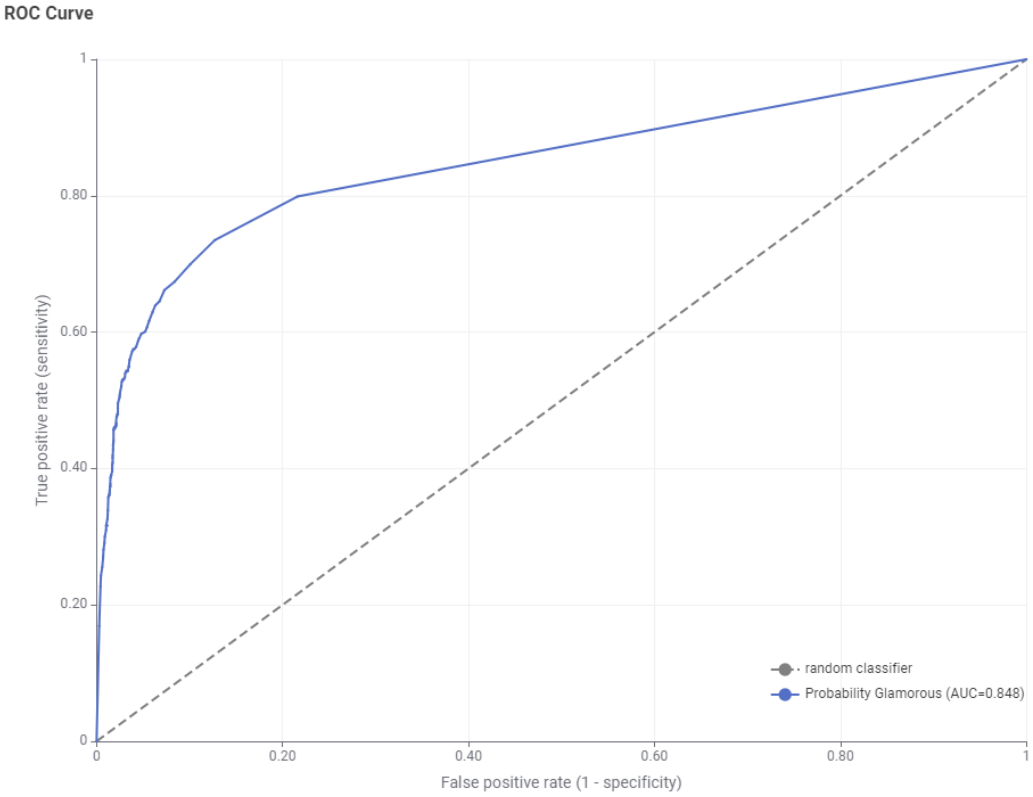
the "Glamorous" class from the others, with minimal overlap between true positive and false positive rates.

The ROC curve for the "healthy" class shows good performance with an AUC of 0.87. The model performs well in picking out class "healthy" from other classes, demonstrating a high degree specificity.

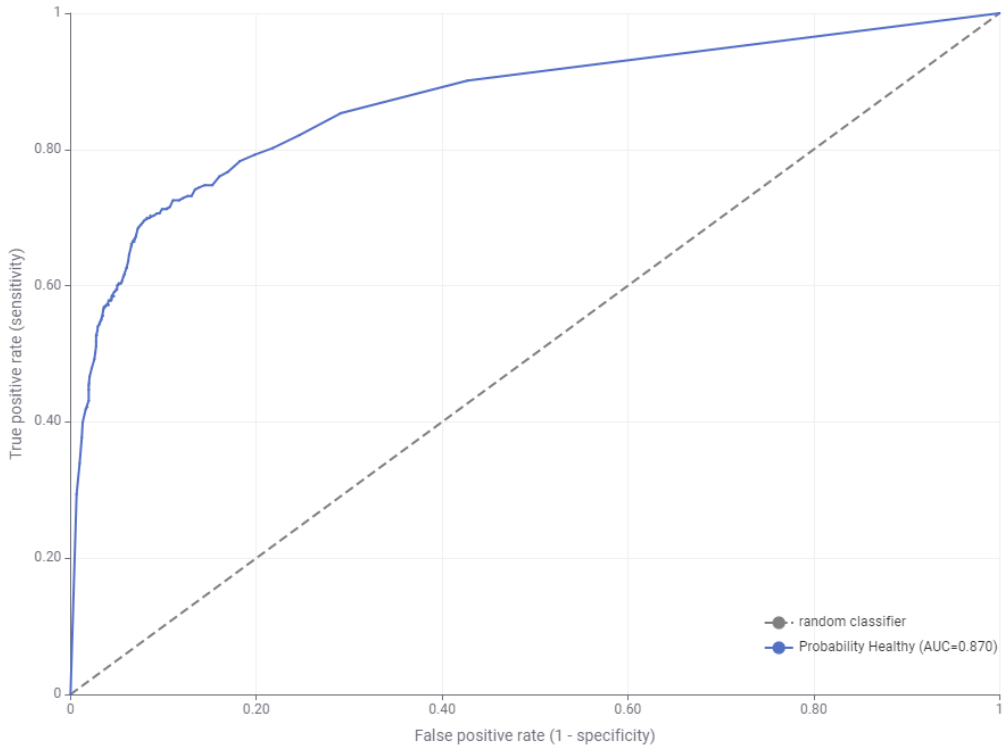
The ROC curve for the "rugged" class also indicates strong performance, since its AUC is 0.88. This curve suggests that the model is effective at identifying the "Rugged" class, maintaining a fair balance between both false positive and true positive rates.
Overall, the Swin Transformer model is performing pretty well across all classes, as indicated by the high AUC values for most classes (0.86 on average). With these results, the Swin Transformer states that it is a robust model for multi-class classification tasks.

Figure 5: ROC and AUC

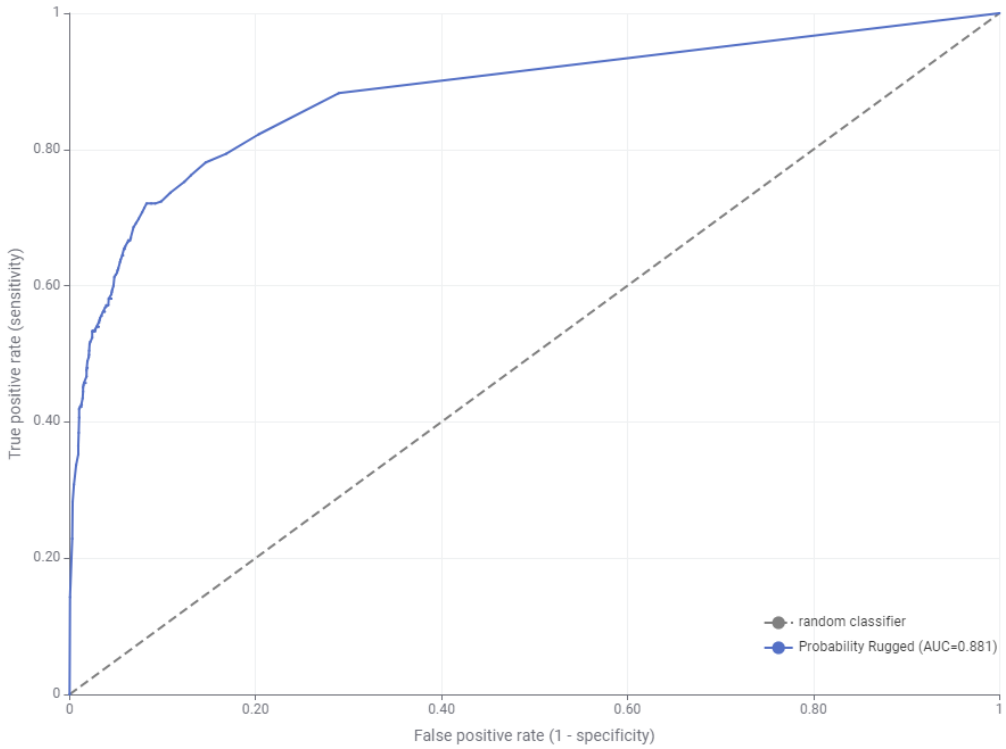




ROC Curve



ROC Curve



3.5.2 Accuracy and Specificity

Accuracy is a metric used to evaluate the performance of a classification model. It measures the proportion of all correct predictions (both true positives and true negatives) out of the total number of predictions made. Accuracy gives a straightforward assessment of how often the model is correct. Accuracy is calculated using the following formula:

$$Accuracy = \frac{True\ Negative\ (TN) + True\ Positive\ (TP)}{Total\ Predictions}$$

Specificity, also known as the true negative rate, is a measure used to evaluate the performance of a binary classification test. It measures the proportion of actual negatives that are correctly identified by the test. Specificity is calculated using the following formula:

$$Specificity = \frac{True\ Negative\ (TN)}{True\ Negative\ (TN) + False\ Positive\ (FP)}$$

For the category “fun,” the model achieves a specificity of 95.5%, indicating that it accurately identifies 95.5% of the negative cases for the "fun" class. This high specificity means that the model is highly effective at distinguishing "fun" from non-"fun" instances. The corresponding accuracy for this category is 90.1%, suggesting that the model generally classifies "fun" instances correctly, both in identifying true positives and true negatives.

For the category “glamorous,” the model attains a specificity of 97.6%, meaning it successfully identifies 97.6% of the negative cases for the "glamorous" class. This high specificity shows that the model is proficient at differentiating "glamorous" from non-"glamorous" instances. The corresponding accuracy for this category is 91.6%, indicating that while the model is generally effective, its performance is slightly lower compared to other classes, potentially due to a higher rate of false positives or false negatives.

For the category “healthy,” the model achieves a specificity of 93.7%, indicating it correctly identifies 93.7% of the negative cases for the "healthy" class. This high specificity suggests that the model is very adept at distinguishing "healthy" from non-

"healthy" instances. The corresponding accuracy for this category is 89.7%, which means the model correctly classifies "healthy" instances most of the time, identifying both true positives and true negatives.

For the category “rugged,” the model reaches a specificity of 95.7%, indicating it accurately identifies 95.7% of the negative cases for the "rugged" class. This high specificity demonstrates that the model is very effective at distinguishing "rugged" from non-"rugged" instances. The corresponding accuracy for this category is 90.8%, the highest among the four classes, indicating that the model excels in correctly classifying "rugged" instances. From them, we can conclude that the Swin Transformer achieves better results compared with BrandImageNet, in fact, both the overall accuracy (90.5% for the Swin Transformer and 90.25% for the BrandImageNet) and the AUC (0.87 for the Swin Transformer and 0.85 for the BrandImageNet) are higher for the transformer-based model.

Table 2 Comparison of the results using different algorithms

Brand Attribute	Algorithm	AUC	Accuracy
Rugged	BrandImageNet	0.88	0.91
	Swin Transformer	0.88	0.908
Fun	BrandImageNet	0.81	0.91
	Swin Transformer	0.84	0.901
Glamorous	BrandImageNet	0.85	0.88
	Swin Transformer	0.85	0.916
Healthy	BrandImageNet	0.86	0.91
	Swin Transformer	0.87	0.897
Overall	BrandImageNet	0.85	0.9025
	Swin Transformer	0.86	0.905

4. Conclusion and further research

4.1 Limitations of Transformer-based models

Transformer-based models, while powerful and versatile, have several limitations that need to be addressed. One of the biggest limitations is that they are computationally expensive. Transformers, particularly in their standard form such as Vision Transformers, demand substantial computational resources. The central operation of a Transformer is self-attention, which, in practice, has complexity that is quadratic in the length of the sequence and when scaling, this makes the models very expensive in terms of computation and memory. Another limitation is the requirement for huge datasets. In fact, transformers generally need massive amounts of labelled data to achieve high performance. This requirement can pose a considerable challenge for tasks where labelled data is limited or costly to acquire. Although pretraining on extensive datasets followed by fine-tuning can help address this problem but, doing so introduces additional complexity and increases computational costs. Transformers are also prone to overfitting due to their high capacity and flexibility, especially during training in small or noisy datasets. To make the model less predisposed to overfitting, can be used regularization techniques such as dropout, data augmentation, and careful hyperparameter tuning. Another disadvantage of this type of model is interpretability. The self-attention mechanism, while powerful, complicates the interpretability of decisions made within the model. Unlike simpler models where feature importance could be understood very easily, a Transformer model often acts like a black box, where understanding and explaining its behaviour is hard. Scalability is a further concern. While Transformers excel in capturing dependencies in sequential data, handling very long sequences can be problematic. Swin Transformer, that uses windowed attention, and memory efficient attention developed techniques take on this challenge but with the trade-off to the complexity and performance. An additional problem deriving from transformer is adapting them to different domains. While they are highly effective in domains where they have been pre-trained, transferring them to significantly different domains might not always yield the same level of performance. Domain-specific adaptations and fine-tuning are necessary, which can be resource-intensive.

Nevertheless, the benefits provided by Transformer-based models many times surpass the challenges related to them, especially in those tasks where great accuracy and ability to capture complex patterns is fundamental. Further research can try to find ways to overcome these limitations by developing more efficient architecture, improving interpretability, and reducing the computational load related to training and deploying these models.

4.2 Conclusion

In this research, the potential of Transformer-based models has been explored, specifically Vision Transformers (ViT) and Swin Transformers, toward image classification tasks in visual marketing compared to traditional BrandImageNet. The results deriving from the two Transformer-based models clearly outperformed the BrandImageNet model in every evaluation metrics as shown in the Table 2. This gives an indication on how Transformers model made an advancement in the computer vision domain, especially in dealing with image classification tasks.

In conclusion, the use of Transformer-based model instead of CNN-based marks a significantly growth in the field of image classification for visual marketing. The higher performance of Vision Transformers and Swin Transformers creates new opportunity for businesses to gain more insights of brand perception and consumer behaviour through social media imagery. Future research could focus on further optimization of these models, exploring their applications with other attributes different from the ones used is this paper, and addressing any remaining limitations that they have concerning computational efficiency and scalability.

5. References

Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia. (2017). Attention Is All You Need.

Dosovitskiy, Alexey & Beyer, Lucas & Kolesnikov, Alexander & Weissenborn, Dirk & Zhai, Xiaohua & Unterthiner, Thomas & Dehghani, Mostafa & Minderer, Matthias & Heigold, Georg & Gelly, Sylvain & Uszkoreit, Jakob & Houlsby, Neil. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

Liu, L., Dzyabura, D., & Mizik, N. (2020). Visual listening in: Extracting brand image portrayed on social media. *Marketing Science*, *39*(4), 669-686.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015). https://doi.org/10.1038/nature14539

Wang, Wenhai & Xie, Enze & Li, Xiang & Fan, Deng-Ping & Song, Kaitao & Liang, Ding & Lu, Tong & Luo, Ping & Shao, Ling. (2021). Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions.

Wu, Haiping & Xiao, Bin & Codella, Noel & Liu, Mengchen & Dai, Xiyang & Yuan, Lu & Zhang, Lei. (2021). CvT: Introducing Convolutions to Vision Transformers. 22-31. 10.1109/ICCV48922.2021.00009.

K. Han et al., "A Survey on Vision Transformer," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 87-110, 1 Jan. 2023, doi: 10.1109/TPAMI.2022.3152247.

Liu, Ze & Lin, Yutong & Cao, Yue & Hu, Han & Wei, Yixuan & Zhang, Zheng & Lin, Stephen & Guo, Baining. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. 9992-10002. 10.1109/ICCV48922.2021.00986.

Pu, Q., Xi, Z., Yin, S. *et al.* Advantages of transformer and its application for medical image segmentation: a survey. *BioMed Eng OnLine* **23**, 14 (2024). https://doi.org/10.1186/s12938-024-01212-4

McAuley, J., Leskovec, J. (2012). Image Labeling on a Network: Using Social-Network Metadata for Image Classification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds) Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7575. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33765-9_59

Stadlen A (2015) Find every photo with Flickr's new unified search experience. Accessed May 7, 2015, https://blog.flickr.net/en/2015/05/07/flickr-unified-search/.

Google/VIT-base-patch16-224 · hugging face google/vit-base-patch16-224 · Hugging Face. Available at: https://huggingface.co/google/vit-base-patch16-224.

Microsoft/Swin-base-patch4-window7-224 · hugging face. microsoft/swin-base-patch4-window7-224 · Hugging Face. https://huggingface.co/microsoft/swin-base-patch4-window7-224

Kingma, D.P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *CoRR, abs/1412.6980*.

Srivastava, Nitish & Hinton, Geoffrey & Krizhevsky, Alex & Sutskever, Ilya & Salakhutdinov, Ruslan. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 15. 1929-1958.

Bishop. (2006). *Pattern recognition and machine learning*. Springer New York.

Aggarwal, C. C. (2023). *Neural Networks and deep learning a textbook*. Springer International Publishing AG.