Introduction

The study aims to construct a deep learning model to accomplish a multi-label image classification task. This involves training a model to accurately assign multiple labels to each image with given labels and predicting new images with trained model.

The study is important for several reasons. Firstly, we can learn how to build an image learning network and how to fine-tune a pre-trained model on a specific dataset. Secondly, the task of muti-label classification is complex and challenging in the field of computer vision. By constructing a deep learning model to solve this problem, we can develop more accurate approaches for many computer vision tasks such as object recognition, content tagging and image retrieval. Finally, we can proficiently master the use of PyTorch, a powerful deep learning framework, through this study.

VIT (Vision Transformer) model is a computer vision model based on Transformer architecture, which is proposed by the Google Brain Team in 2020. (Google Research, 2020). VIT model divides input images into small patches of size 16 x 16, then processes each small patch into a certain length vector and then do the same as the original Transformer model. VIT is highly effective in image classification, object detection and image generation.

The motivation behind using the VIT model in a multi-label classification task stems from several advantages it offers over CNN-based architecture.

Firstly, the self-attention mechanism in Transformer architecture allows the model to capture the relationship between different parts of an image. For example, in an image where there are two objects to be detected, a boat and the sea, typically, the boat is above the sea. Then VIT can effectively capture this positional information, which is crucial for accurate multi-label classification.

Secondly, there are pre-trained VIT models on large-scale image datasets like ImageNet, they can be easily fine-tuned on specific tasks with relatively small amounts of data, leading to faster convergence and improved performance.

1. Related works

The ResNet model proposed by Kaiming He et al. in 2015 (Kaiming He, 2016) has achieved great success and influence in the field of computer vision. CNNs based networks containing residual blocks have achieved outstanding results in computer vision competitions.

Apart from CNNs based architecture, the VIT model, which was initially proposed by Alexey Dosovitskiy and others in 2020 (Dosovitskiy, et al., 2021), has also demonstrated outstanding performance. Although this paper was not the first to apply Transformer to computer vision tasks, its work has achieved significant results and also has driven numerous subsequent related researches. Instead of adding self-attention to CNN-like architectures or theoretically replacing the convolutions with Transformer, their work confirmed that when the Transformer architecture is directly applied to computer vision tasks, it can achieve quite impressive performance on large-scale datasets (14M-300M images).

Another research found that using a multi-layer perceptron (MLP) structure without employing Convolutional Neural Networks (CNNs) or Vision Transformer, can also achieve competitive results on large datasets (Tolstikhin, et al., 2021). They introduced MLP-Mixer, which comprises two MLP layers. One is applied to image patches to capture local information and another operates across patches to recognize relationship between various parts of an image.

1. Techniques

The model used in this paper is tuned from public model

“vit\_medium\_patch16\_reg1\_gap\_256.sbb\_in1k” which is downloaded from [timm/vit\_medium\_patch16\_reg1\_gap\_256.sbb\_in1k · Hugging Face](https://huggingface.co/timm/vit_medium_patch16_reg1_gap_256.sbb_in1k).

Methods:

1. Data Preprocessing:
2. Image resize:

Because the size of sample images are not same and the optimal image size for the chosen model is 224 x 224, it is necessary to resize the images to 224 x 224 pixels before training the model.

1. Image augmentation:

Image augmentation is a technique used to increase the diversity of training by applying transformations to the images. This helps improve the model’s generalization and robustness.

In this study, both random rotation and flipping are both applied to the input images. Specifically, all input images are randomly rotated by 10 degrees and randomly flipped vertically.

1. Normalization:

Normalization is the process of scaling the pixel values of images by subtracting the mean value and dividing by the standard deviation. The used mean and standard deviation are based on the ImageNet dataset’s mean and standard deviation.

1. Visual Transformer：

There are several layers in a visual Transformer model, they are arranged as follows:

1. Patch embedding:

The initial step involves splitting the original image into smaller patches, each sized 16 x 16 x 3 (height x width x channels). Given that the image in this study is 224 x 224 x 3, this results in 196 patches (224 x 224 / (16 x 16) = 196).

Following this, the patches are flattened into sequences, each with a length of 16 x 16 x 3 = 768. Consequently, the final input dimension for each image is 196 x 768.

Instead of splitting the image into patches and flattening them, a convolutional layer with a kernel size of 16 x 16 and a stride of 16 x 16 is employed, and the number of kernels is 768. After this convolutional layer, the output size becomes 14 x 14 x 768. After flattening the output, the same size of output (196 x 768) can be achieved.

In the selected model, the number of kernels is 384, resulting in an output size of 196 x 384 after the convolutional layer. However, to maintain consistency with the description in the original paper, the subsequent analysis uses the size of 196 x 768.

1. Positional encoding:

In a visual transformer architecture, which consists only of an encoder without a decoder, therefore a class token (represented as a vector) is concatenated with the image vector. The class token has a dimension of 1 x 1 x 768, and after being combined with the image vector, the resulting input image matrix now has a dimension of 197 x 768. This process enables the transformer model to predict the image class with this token.

In text transformer architectures, positional encoding is applied to text vectors to provide the model with information about the sequence order. Similarly, in visual transformer models, a learnable positional encoding is used to convey position information to the model. This positional encoding matrix, with the same dimensions as the image input matrix, is added to the image input matrix, thereby maintaining the original dimensions.

1. Transformer encoder:

Before entering multi-head attention layer, a layer normalization is applied to the input data. Layer normalization helps to stabilize and normalize the input data, improving the overall training and convergence of the model.

Layer normalization calculates the mean and standard deviation of each vector (each column of matrix) and normalizes the vectors bases on these values. This process help mitigate gradient vanishing or exploding during training, making the model more robust and facilitating faster convergence.

The second layer in a transformer encoder is a multi-head attention layer. In this stage, the input embedding vectors are processed using multiple attention heads.

Each attention head learns different attention weights to capture relevant information at different positions.

Firstly, the input embedding vectors and positional encodings undergo linear transformations. These linear transformation parameters are learned during training. Secondly, the transformed vectors are then split into multiple attention heads. Each head focuses on different patterns of data. Thirdly, three vectors: Query (Q), Key (K), and Value (V) are derived in each head through linear transformations. Fourthly, each Q is dotted with all Ks, and the results are transformed to probabilities through SoftMax function. Finally, using the attention probabilities, a weighted sum of the Vs is computed.

The outputs from all attention heads are concatenated and linearly transformed back to the original dimension (197 x 768).

1. Residual connection:

Residual connection (Kaiming He, 2016) is another important method in the model. By adding the input of the layer to the output of that layer, the model can address the problem of vanishing gradients and train deeper networks.

1. Multi-Layer Perceptron:

In original model, a multi-layer perceptron layer, is applied after Transformer encode layer, converting the data to 197 x 3072, then reducing it back to 197 x 768.

In the selected model, In order to reduce the number of parameters, only one MLP layer is used with the same input and output dimensions. In other words, the input data has a size of 197 x 384 and the output size is also 197 x 384.

1. DropPath and GELU:

DropPath is a regularization technique. Compared with the traditional dropout, DropPath randomly drops entire layers in the network during training while keeps them in testing.

The GELU activation function is defined as follows:

where represents the standard cumulative distribution function (CDF) of the Gaussian distribution. GELU can be seen as a smoother alternative to RELU, providing a balance between linearity and non-linearity.

1. Classifier head

The last layer in the selected model is a MLP layer with an input size of batch size x 197 x 384 and an output size of batch size 1000 (197 x 384 = 75,648 to 1000), because there are 18 classes in the labels, another dropout layer and a MLP layer is added to the mode to get the final output with a size of batch size x 18.

To convert these outputs into probabilities for each label prediction, a sigmoid layer is applied. The sigmoid function transforms the outputs to a probability between 0 and 1, representing the probability of each label being present in the input data. By using a threshold of 0.5, labels with probabilities larger than the threshold are set to 1, indicating that the corresponding labels are predicted to be present, while labels with probabilities less than or equal to the threshold are set to 0. This approach enables the model to perform multi-label classification by predicting the presence or absence of multiple labels simultaneously for each input sample.

1. Reasonability of the method
2. Vision Transformer for Image classification:

Vit model have shown great performance in various computer vision tasks, the self-attention and residual connection architecture makes it effective to understanding complex visual patterns.

1. Multi-label classification:

Vit model is well-designed for multi-label classification because the transformer architecture can capture information of the relationship for different parts of an image, which is especially important in multi-label classification where objects or targets may be distributed across the image.

1. Model size and efficiency:

The chosen VIT model with a size of 84.2 MB falls within the specified constraint of less than 100 MB. This makes it practical for deployment and usage in resource-constrained environments.

The average training time per epoch on an NVIDIA GeForce RTX 2070 GPU is approximately 4 minutes, which indicates that the model is easy to train.