# Kent State University


MIS-64061: Advanced Machine Learning

Spring 2022

Final Project

Fashion MNIST Image Classification using CNN & VIT - TensorFlow

By:

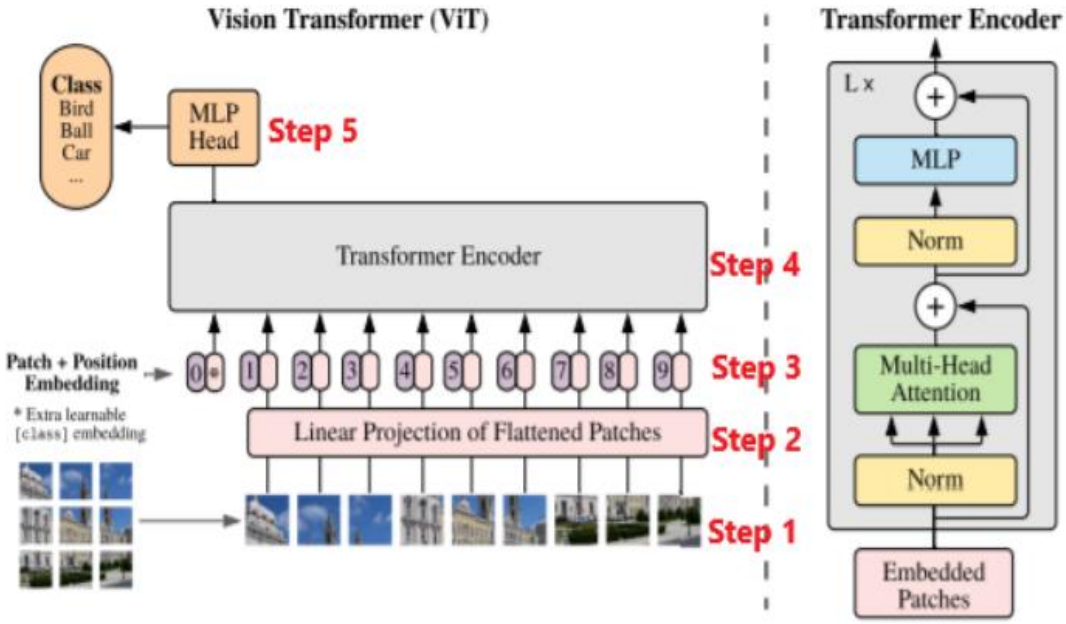
Jayasri Maditati

# Table of Contents

## Introduction:

In 2017, Google AI unveiled a new neural network architecture called Transformer and they claimed that transformer worked better than the leading approaches. In four years, it has become talk of the town and revolutionized the NLP field with its improvements in efficiency and accuracy. Along with neural architectures, transformers bring new capability to Machine Learning. In 2021, 'An Image is Worth 16X16 Words' which was presented in International Conference for Representation Learning (ICLR), by Alex Dosovitskiy et.al. showed for the first time how Transformers can be implemented for Computer Vision tasks and outperform CNN in image classification tasks. As part of this project, the Vision Transformer architecture will be explained, the implementation is performed on the Fashion MNIST dataset and provide the detailed comparison of the CNN and VIT models.

## Vision Transformer Architecture:




Source: AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

The high-level steps to implement the Vision Transformer in TensorFlow are outlined below in nutshell.

**Step 1:** Split the image into fixed-size patches.

**Step 2:** Flatten the 2D image patches to 1D patch embedding and linearly embed them using a fully connected layer.

**Step 3:** Unlike other architectures, transformers have no idea of sequence or any positional information. So, we use the positional embeddings to the patches to retain positional information.

**Step 4:** Transformer Encoder has alternating layers of multiheaded self-attention and MLP blocks. Layernorm (LN) is applied before the self-attention block and the MLP block. The residual connections are applied after every block to avoid vanishing gradient problem. This block helps learn local and global dependencies in the image.

**Step 5:** A classification head is implemented using MLP with one hidden layer at pre-training time and a single linear layer for fine-tuning for image classification.

## Implementation of VIT:

In the above section, VIT architecture has been summarized. I have used fashion MNIST dataset for the implementation purpose.

**Dataset: Fashion MNIST**

Fashion-MNIST is a dataset of images of fashion accessories—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from ten classes. Fashion-MNIST was intended to serve as a replacement for the original MNIST dataset for benchmarking machine learning algorithms.

The full implementation of the VIT on Image classification using TensorFlow code is found here:

https://github.com/JayasriMaditati/Advanced_ML.git

As the main objective of this project is to provide the comparison between CNN and Transformers for the Fashion MNIST dataset.

## Comparison between CNN and VIT:

| CNN | VIT |
|---|---|
| Feature Map | Attention Map |
| Pixels | Patches |
| Batch Normalization | Layer Normalization |
| High Inductive Bias | Weaker Inductive Bias |
| 2D neighborhood - Kernel | Spatial Relationships – training from scratch |
| Not sensitive to dataset size | Sensitive to dataset size |

Now I would like to highlight few topics which requires the detailed explanation in terms of comparison.

- **Receptive field range:**

CNNs use 3x3 or 5x5 size kernels, so each layer can only have a corresponding field of view. And the field of view expands as it propagates through the layers, but the expansion is linearly increasing with depth.

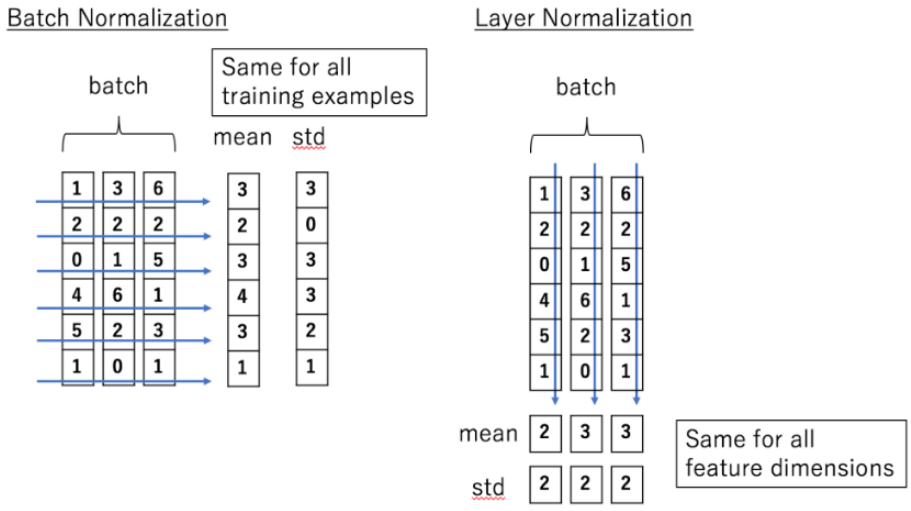
The Transformer, on the other hand, uses Self-Attention, which allows the network to see the entire image from the initial layer. Since each patch is treated as a token and all of them are correlated and calculated, it is possible to learn global features from the beginning.

- **Impact of dataset size and inductive bias:**

The main difference is — In CNN, the kernels help us to learn/understand about the 2D neighborhood structure; but in transformers, 2D structure is not used and the positional embeddings at the initialization time carry no information about the 2D position of the patches and all spatial relations between the patches has to be learned from scratch. Since it has to learn the relations from scratch, it needs lot of data to outperform CNN. So, the dataset size plays a key role in the vision transformer.

- **Normalization:**

To improve the stability and performance of the model, we used Batch Normalization in CNN which is done by scaling with the mean and standard deviation for each training example (Batch) whereas the Layer normalization which is used in transformers is done by computing mean and sd for each feature.




*Batch Normalization and Layer Normalization(*Source)

A less known issue of Batch Norm is that how hard it is to parallelize batch-normalized models. Since there is dependence between elements, there is additional need for synchronization across devices. While this is not an issue for most vision models, which tends to be used on a small set of devices, Transformers really suffer from this problem, as they rely on large-scale setups to counter their quadratic complexity. In this regard, layer norm provides some degree of normalization while incurring no batch-wise dependence.

- **Texture Perturbations:**

Compared to the CNN model, the Transformer model (ViT) is relatively robust to texture perturbations [1].

## Results:

- **Accuracy & Loss:**

Below are the results for all the models created as part of the implementation process. All the models are trained for 30 epochs.

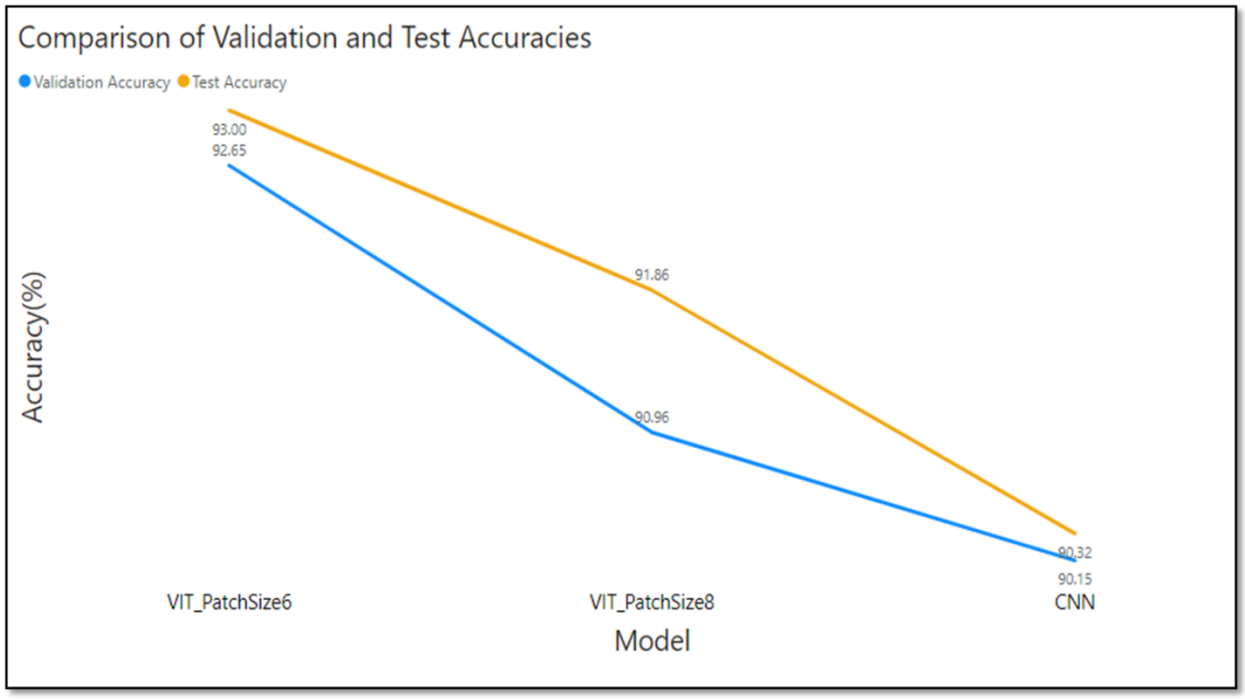
| Model | Validation | | Test | |
|---|---|---|---|---|
| | Loss | Accuracy | Loss | Accuracy |
| VIT_PatchSize6 | 0.2045 | 92.65 | 0.2016 | 93.00 |
| VIT_PatchSize8 | 0.2411 | 90.96 | 0.2238 | 91.86 |
| CNN | 0.4661 | 90.15 | 0.2764 | 90.32 |

The vision transformer models have been trained from scratch incorporating the data augmentation and drop out techniques and the CNN was trained without any regularization techniques and results are as part of baseline Performance.

From the above results, we can see that the accuracy is high for the VIT models than the CNNs. However, we can see a slight increase in terms of accuracy than CNN baseline model performance. This is due to the medium sized input dataset. The performance would have been exceptional if the input would have been large.

- **Impact of Patch size on the performance:**

Patch size is a significant hyperparameter in the VIT architecture and I tried to evaluate the model performance by tuning the patch size and found out that accuracy increases as the patch size decreases which is evident in the below chart.




## Conclusion:

Although Vision Transformer has achieved good accuracy than CNN, it has a limitation. To obtain good accuracy, the model requires larger datasets due to weak inductive bias. However, to overcome this, various improvement methods have been proposed.

One such attempt is to reduce the amount of data required by using CNNs: DeiT [2] uses a knowledge distillation framework, where CNNs are used as the teacher model and knowledge is fed to the Transformer model. By doing so, the transformers can outperform CNN irrespective of dataset size.

Finally, I would like to conclude that Transformers completely replaced RNNs in NLP. Now, they aim to replace Convolutional Neural Networks (CNNs). It is a promising model that might make CNNs extinct in the future, but not yet. It is still challenging for the model to perform on the smaller datasets and other computer vision tasks, such as image segmentation and detection.

# References:


1. Shikhar Tuli, Ishita Dasgupta, Erin Grant, Thomas L. Griffiths. Are Convolutional Neural Networks or Transformers more like human vision? arXiv(2021)

2. Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, Hervé Jégou. Training data-efficient image transformers & distillation through attention. arXiv(2020)

3. https://towardsdatascience.com/understand-and-implement-vision-transformer-with-tensorflow-2-0-f5435769093

4. Recent Developments and Views on Computer Vision x Transformer | by Akihiro FUJII | Towards Data Science

5. https://keras.io/examples/vision/image_classification_with_vision_transformer/#build-the-vit-model

6. https://www.section.io/engineering-education/vision-transformer-using-transformers-for-image-recognition/