Image Classification using Convolutional Neural Network and Vision Transformer

***Abstract*— Image classification is a very popular computer vision area of research and is extensively studied over the recent past. Various popular deep learning architectures have emerged as a result which outperformed the traditional techniques that have been used. Convolutional neural networks in particular have emerged as the top option when it comes to image classification and other similar computer vision tasks. Popular architectures such as VGG net, Inception net and Resnets are all based on the convolutional neural networks. Another deep learning algorithms that have gained popularity in the recent past is the transformer architectures. Transformers have recently shown to perform outstandingly on natural language processing tasks. Recent approaches have tried to apply it to computer vision problems as well. In this research article, the prime focus of the work is to compare the performance of these two popular algorithms for the task of image classification. For this purpose three different architectures of convolutional neural network and vision transformers are designed and evaluated on three different benchmark datasets which are fashion mnist, cifar10 and cifar100. While evaluating the algorithms it has turned out that the transformer architecture has outperformed the convolutional neural network architectures in case of the cifar10 and cifar100 datasets. On fashion mnist, both the vision transformer and convolutional neural network performed equally well.**

Keywords—Image Classification, Convolutional Neural Network, Vision Transformer, Deep Learning.

I. INTRODUCTION

Image classification is referred to as the process of grouping images into one of the predefined groups or categories which is one of the fundamental areas of focus for modern day computer vision practitioners and researchers. Image classification serves as the basis for many other similar computer vision tasks. For instance object localization where the task is to localize the region of the classified object, object detection where the task of localization and classification is combined and image segmentation where the classification and localization is performed at the pixel level. Although the task of image classification seems pretty simple and obvious for a human, for a computer system however it is pretty challenging and requires quite some effort for accomplishing it. Contrary to humans, a computer system sees the image as an array of numbers. Thus slight variations in the contrast, orientation, angles, viewpoints etc could make it very challenging for the machine to identify and classify the image. The traditional approach in image classification consisted of a two stage approach. The first stage, which is a manual stage consisted of extracting valuable features from the image which carry the important information pertaining to the image. These handcrafted features are then further fed to a trainable classifier which performs the classification task based on the features. This is no doubt a sane approach of performing the task however the main limitation of this approach lies in the fact that the accuracy of the classifiers are highly dependent upon the quality of the extracted features. Thus if the extracted features are of good quality, the classifier would perform a reasonably good job in classifying the images. However if the handcrafted features are not good enough, i.e if the features do not carry the valuable intrinsic information of the image, then no matter how good the classifier architecture is, it will not perform a good job in classifying the images. More recently with the raise of deep learning based approaches which takes the advantage of multiple nonlinear layers of processing information for the task of extracting valuable features, image transformations and pattern recognition have tremendously improved the accuracies of the models and have automated the task of handcrafted features. Amongst the deep learning based algorithms, Convolutional Neural Networks (CNN) have become very popular in the field of computer vision. CNN emerged in inspiration taken from the visual cortex of animals [1] which first appeared in 1991 [2] by LeCun for the task of handwritten zipcode recognition. The algorithm operated directly on the input image without subjecting it to a prior preprocessing step as in the previous approaches. However the algorithm did not live up to the mark in complex problems due to the insufficient availability of the training data and the limited computing power. Krizhevsky in [3] in his famous model Alexnet based on the CNN succeeded in bringing the error rate down on the ILSVRC competition [4]. After the success of Alexnet, CNN became the prime focus of many researchers across the globe. Zeiler proposed ZFNet [5] and explored the visual technique of understanding CNN in detail. In [6] NIN network was proposed by Min Lin devising the technique to control the parameters and channels.

Another powerful neural network based architecture that has become quite popular in the recent past is the transformer model. Originally applied to the Natural Language Processing (NLP) tasks, transformer model emerged in [7] which is based on the attention mechanism. Due to its huge success in the field of NLP, researchers have focused their attention on applying the same mechanism in the field of computer vision. As discussed earlier, CNN have acted as the dominant algorithm in the computer vision area, but transformer have recently shown to be the potential alternative to CNN. In [8] Chen proposed the training of a sequence transformer to predict pixels and have achieved performance that is comparable to what the CNN have done for the task of image classification. Another transformer based architecture known as ViT [9] which takes image patches and apply the transformer model on it directly in order to perform image classification. ViT is a pure transformer architecture in its original form and it performed state of the art on many benchmark datasets in the category of image recognition. Apart from image classification, vision transformers have also been deployed in many other computer vision areas such as in [10] the architecture is applied to object detection which is developed at facebook. For semantic segmentation [11], video and image processing [12], [13] and other related areas. However it must be kept in mind that the vision transformer is still new and a lot of research is still underway exploring the potential areas that could utilize the unique architecture in order to solve many computer vision problems.

II. METHODOLOGY

For the task of image classification in this work, two different architectures known as CNN and vision transformer as

described earlier are utilized. In this section, each of the two architectures are explained in detail.

A. *Convolutional Neural Network*

CNN are a class of neural networks which works by learning some filter functions that captures the salient information present in an image. It does it using a math operation called as convolution. A convolution can simply be understood as a filter. What a filter do to an image is that it takes the image, convolves it with another matrix (which is the filter itself) and the output is a different image which is transformed from the normal space. So for instance if we take a filter which is an edge detector filter and do the convolution of it with an image, the output would be another image which would contain the same entities as the input image but all the edges inside it would be enhanced and the rest of the entities would be darkened or minimized. In this way, an edge detector filter extracts out the sharp edges from an image. In CNN many such filters are initialized randomly and they are then learned as the training process is performed by feeding it images. The number of filters used and the size of the filter are choices, also known as the hyper-parameters of the network. A typical filter size could be a 3 by 3 and the number of filters used could be 16, 32, 64 and so on. The larger the filter numbers, more parameters are needed to be learned and thus the model becomes heavy and complex. The filters are applied in layers. So one can apply 16 filters in the first hidden layer, 32 filters in the second and so on. The greater the number of hidden layers, the deeper the network becomes. Number of hidden layers are also a hyper-parameter. Alongside convolution layers, there is another operation that is performed which is known as pooling layer. The purpose of applying the pooling layer is to reduce the size of the output images so that it does not become too heavy for the model and also to highlight only the relevant pixels. A common pooling operation is the maxpooling operation in which a group of pixels are taken and they are replaced by the maximum pixel value. Other pooling operation could include the average pooling. Another hyper-parameter is the stride of the convolution. Stride specifies how many pixels to traverse before forming each of the output of the filter. The larger the stride, the smaller would be the output image size and vice versa. Once the convolution and pooling operations are declared, they are then followed by one or more layers of what is called the fully connected layers. The fully connected layers are the simple neural network layers where one have the perceptron and the non-linear activation. All of these operations are performed in specific order and the designed output model is called the architecture of the CNN. Three such architectures are devised for the task of image classification in this work. They are explained next.

1) *Tiny CNN*

The tiny CNN is the lighter model which has its architecture depicted in Fig. *1*. It is evident in the Fig. *1* that the tiny CNN contains a single convolution operation followed by the maxpooling layer and finally a fully connected layer. The size of the filter is kept 3 by 3 having a total of 32 filters and the maxpooling size is kept at 2 by 2. The default stride size is 1. The number of hidden units in the fully connected layer are 64. The activation function is relu with softmax in the final classification layer. Adam is used as the optimizer and the loss function is the categorical crossentropy.

2) *Small CNN*

The small CNN model is the second model that is tried out. The architecture of the model is depicted in Fig. *1*. This model has a total of 2 convolution layers and 2 maxpooling layers each following the convolution layers. The first convolution layer has total 16 filters while the second convolution layer has total 32 filters. The filter and pooling size are kept the same. These layers are then followed by a single fully connected layer having total of 128 neurons. The activation function and other parameters are kept same.

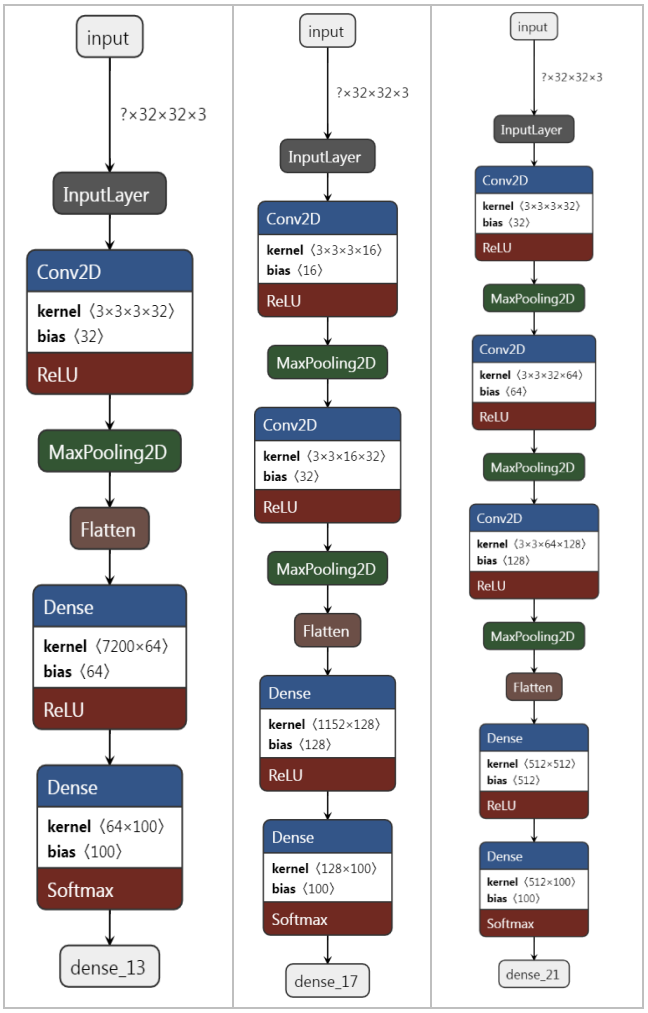

Fig. 1 CNN architecutres. Tiny CNN (left), small CNN (middle), base CNN (right).

3) *Base CNN*

Base CNN is the third and most complex CNN model utilized for the image classification task. The model architecture is depicted in Fig. *1*. This model contains a total of 3 convolution layers each followed by the maxpooling layer. The number of filters used are 32, 64, 128 in each respective convolution layer. These are then followed by the fully connected layer having a total of 512 neurons. The filter size are same 3 by 3 for the convolution layer and 2 by 2 for the pooling layers. The activation function in each convolution and fully connected layers is relu while the output activation is the softmax. The optimizer and other hyper-parameters are kept the same as in the previous architectures.

B. Vision Transformer

The general architecture of a vision transformer consists of the following major parts.

- Divide image into fixed size patches.
- Flat out the resultant patches.
- Synthesize low level linear embeddings from the resultant flattened patches of images.
- Apply a standard transformer encoder to the obtained sequence of image patches.
- Train the obtained model on image labels.
- Fine tune the model for image classification.

Most of the above mentioned steps are self-explanatory except for the transformer encoder which is described next. A transformer encoder consists of the following parts.

- Self-attention multi-head layer which is responsible for the concatenation of the linear attention layers to the corresponding right dimensions.
- Multi-layer perceptron which contains two layer with the activation of Gaussian linear error unit.
- Layer norm which is attached in prior to each of the block in order to enhance the performance and reduce the training time.

Furthermore, residual connections are added after each of the blocks. Residual connections allow the model to have parallel data flow from the previous layers to the next without making them to pass through any of the non-linear activation functions on the way. These connections have shown to improve the overall accuracy as in the Resnet architectures in case of CNN models. For the task of image classification which is the focus of this article, the multi-layer perceptron are put into use. They perform the task by having a hidden layer at the pre-training and a single linear layer for the task of fine tuning the model. Three vision transformer architectures are used for the task in this work which are discussed next. The basic overall architecture of the transformer model is the same as described previously, the difference is only observed in the number of times the basic architecture is stacked on top of each other. So for the tiny vision transformer the number of layers are kept at 4, for the small model the number of layers are kept at 6 and for the base model the layers are kept at 8. The first layer in each of the basic architecture is the normalization layer which is followed by the multihead attention layer of the transofmer. Finally the multi-layer perceptron layer is applied. Each of the three models are depicted in Fig. *2*. Additionally a dropout layer is added in order to account for the model over fitting on the training data. Over fitting is a grave issue in many of the deep learning models. It happens when the model is performing well on the dataset which it is trained on but it does not perform well on the actual test data. Over fitting mainly occurs due to one of these reasons. The model is too complex for the data. Training data is not adequate. In order to curb over fitting methods such as regularization and dropouts are used. In regularization, a penalty is set on the weights of the model thus they are kept to be minimum. Alongside that, a data augmentation layer is also added in order to enhance the training data. The augmentation operations used include flip, rotation and zoom. The optimization algorithm used is adam with weight decay option. Each of the models are trained for a total of 20 epochs.

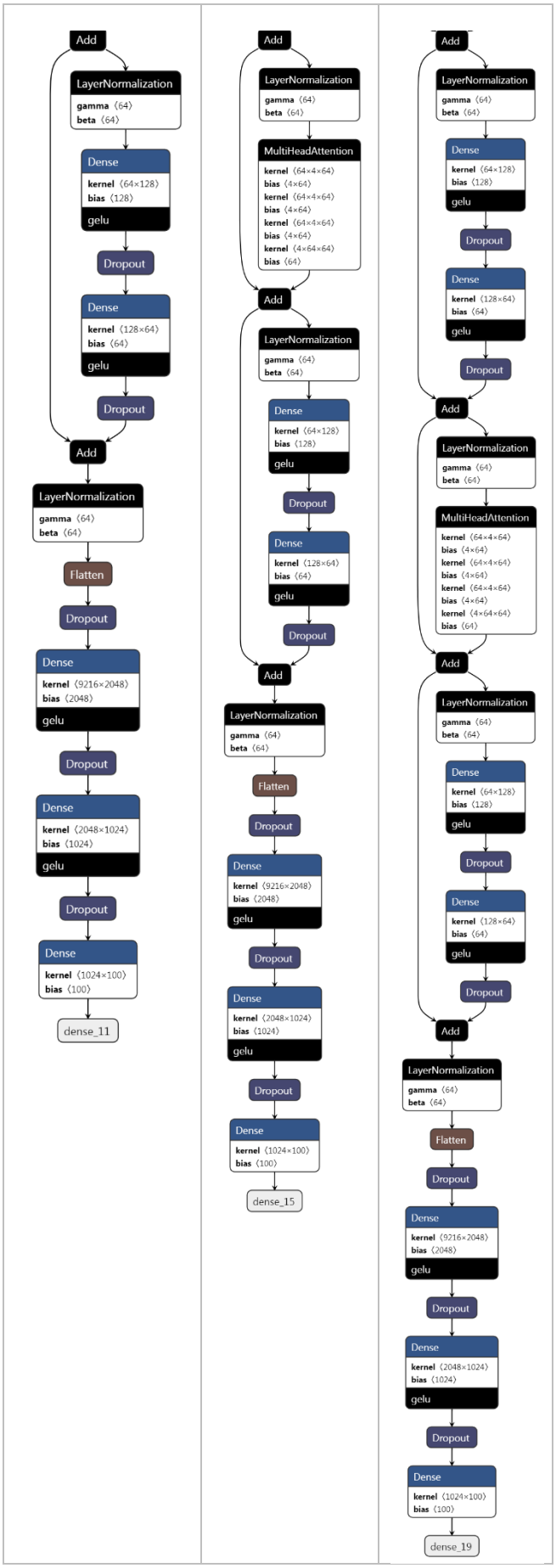

Fig. 2 Vision transformer architectures. Tiny (left), small (middle) and base (right).

III. Experiments and Results

The models described in the previous sections are trained and tested on three different datasets. These include the fashion mnist, cifar10 and cifar100 datasets. Each of them are explained below.

A. Datasets

1) Fashion MNIST

The popular fashion mnist dataset was proposed in 2017 by the zalando fashion research lab. The sole purpose of releasing this dataset was the fact that the original mnist dataset which is a handwritten digits dataset, became too easy for the algorithms to be evaluated on. Thus need was felt to set a new benchmark dataset in order to test and evaluate new state of the art machine and deep learning models. Fashion mnist dataset contains images of 10 different fashion related objects. Each of the image in the dataset has a size of 28 by 28 pixels. The images are in grayscale format with pixel intensity values ranging from 0 to 255 and the background pixels are represented by dark pixels having intensity value of 0. The classes in the dataset include T-shirt, Pullover, Trousers, Dress, Coat, Shirt, Sandal, Bag, Sneaker and Ankle Boot. These are depicted in Fig. *3*. The total images in the dataset are 70000. 60000 of the images are assigned to the training set and rest of the 10000 are in the test set. The images per class are evenly distributed which means that each of the class has 6000 images in the training set and 1000 in each of the test set class.

2) CIFAR 10

Cifar10 is another popular benchmark dataset mainly used by the researchers across the globe for evaluating their new devised algorithms and architectures. The dataset contains 60000 images stored in numpy array. The dataset is evenly distributed which means that each class in the dataset has an equal number of images which is 6000 per class. Amongst the 60000 images, 50000 images belong to the training set and rest of the 10000 belong to the test set. Contrary to fashion mnist, cifar10 dataset images are colored images which contains three color channels. The size of each image is 32 by 32 by 3 where 3 signifies the color channel. The 10 in cifar10 specifies the total classes in the dataset. These 10 classes include airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. These are the most common objects one can observe on a daily basis thus making it an appropriate place to start with for image classification models. The classes are depicted in Fig. *4*.

3) CIFAR 100

Cifar100 is the extended version of the cifar10 dataset, which instead of 10 contains a total of 100 classes. Each of the object has a superclass and then subclasses. For instance the superclass aquatic mammals has subclasses such as whale, otter, beaver, seal, and dolphin. The total number of images is still 60000 thus in cifar100, instead of 6000 images per class, it instead has 600 images per class. This signifies that the dataset is still evenly distributed but has less number of images per class than cifar10 dataset. The statistical details regarding all the datasets are tabulated in Table *1*. As evident from the presented table, the fashion mnist dataset has the highest total number of images having a total of 70000 images. The CIFAR datasets on the other hand has larger size and 3 color channels thus making each of its image bigger.

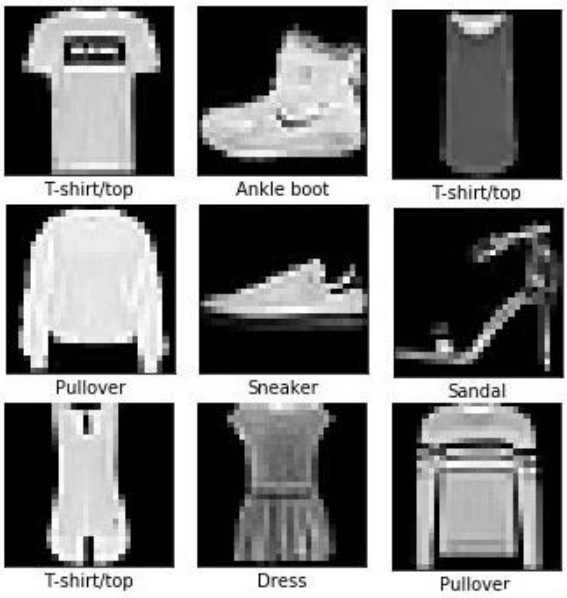

Fig. 3 Fashion mnist dataset classes.

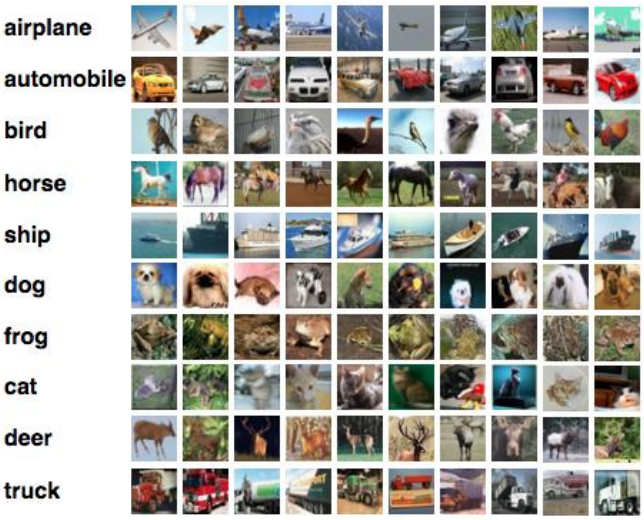

Fig. 4 CIFAR10 dataset classes.

Table 1 Fashion MNIST, CIFAR10 and CIFAR100 dataset statistics.

Attribute	Fashion MNIST	CIFAR 10	CIFAR 100
Total Images	70000	60000	60000
Training Images	60000	50000	50000
Test Images	10000	10000	10000
Image Size	28 × 28	32 × 32	32 × 32
Color Channel	1	3	3
Number of Classes	10	10	100
Images per Class	7000	6000	600

B. Convolutonal Neural Network

In this section, the results obtained with each of the CNN models are presented and described. While training the model, the categorical cross entropy loss of each of the model gradually decreases with each epoch and the accuracy of the models correspondingly increases which is evident from the loss and accuracy plotted against the epochs as depicted in Fig. *5* and Fig. *6* respectively which are obtained for the tiny CNN model and fashion mnist dataset.

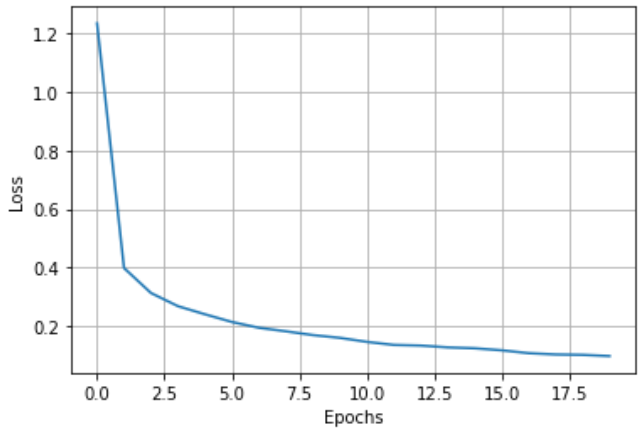

Fig. 5 Tiny CNN model loss plot against the epochs for the fashion mnist dataset.

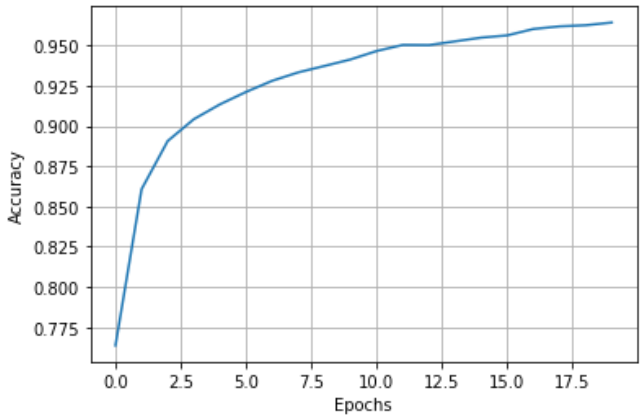

Fig. 6 Tiny CNN model accuracy plot against the epochs for the fashion mnist dataset.

The accuracy and loss scores for each of the three models and datasets are tabulated in Table *2*. As evident from the table, for the fashion mnist dataset, the tiny CNN model achieves and accuracy score of 90 which is slightly higher than the small and base CNN models. Although the tiny CNN model is the lighter model of all, still it managed to have higher accuracy score, however it must be noted that the rest of the small and base models also had almost similar accuracy scores which signifies the fact that all the models performed equally on the fashion mnist dataset and the slight variations could be regarded as the randomness in the weight initialization or other such factors. For cifar10 and cifar100 dataset, the accuracy scores of each of the models vary significantly. For cifar10, the tiny model totally fails to perform having a loss score of 2.3 and accuracy score of 0.10 which means that the model does not have the capacity to recognize the intrinsic patterns in the images and thus classify them correctly. The small model however manages to perform somehow better with a loss score of 1.74 and accuracy of 0.58. Finally, the base CNN model performs on top with highest accuracy of 0.67 in classifying the objects. For cifar100 dataset, the basic trend of accuracies is the same as expected with tiny CNN performing worst having accuracy of 0.01 and loss of 4.6. Small CNN performs somehow good with an accuracy score of 0.27 and base CNN again topping the list with the highest accuracy score of 0.31.

Table 2 Loss and Accuracy scores obtained with CNN models.

Model	Fashion MNIST		CIFAR10		CIFAR100	
	Loss	Acc.	Loss	Acc.	Loss	Acc.
Tiny CNN	0.42	0.90	2.3	0.10	4.6	0.01
Small CNN	0.38	0.89	1.74	0.58	3.37	0.27
Base CNN	0.47	0.88	1.88	0.67	4.83	0.31

C. Vision Transformer

The results obtained with vision transformer are evaluated in the similar way by comparing the obtained accuracy and loss scores associated with each of the model for each of the datasets. First observing the gradual decay of the lost score with each training epoch is depicted in Fig. *7* and the corresponding raise in the accuracy score in Fig. *8* which signifies the fact that the transformer models are learning with each training epoch and thus reducing the loss score gradually. These plots are obtained for the tiny transformer model and fashion mnist dataset in order to have a comparison with the CNN model described previously. By observing the plots, it becomes evident that the transformer model learns slowly than that of the CNN model. This is concluded due to fact that the loss curve drops gradually and slowly flats out with each training epoch. Contrary to this, the loss plot for the CNN model has sudden drop in the loss curve as the model starts learning in the initial epochs, and then quickly flats out in the later epochs. Similar trend is observed in the accuracy curves of the model.

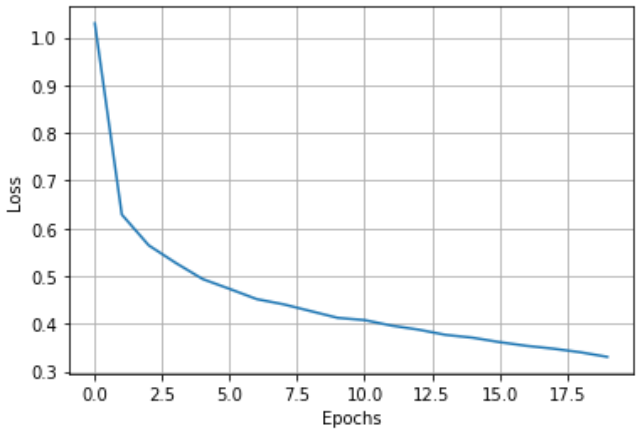

Fig. 7 Tiny VT model loss plot against the epochs for the fashion mnist dataset.

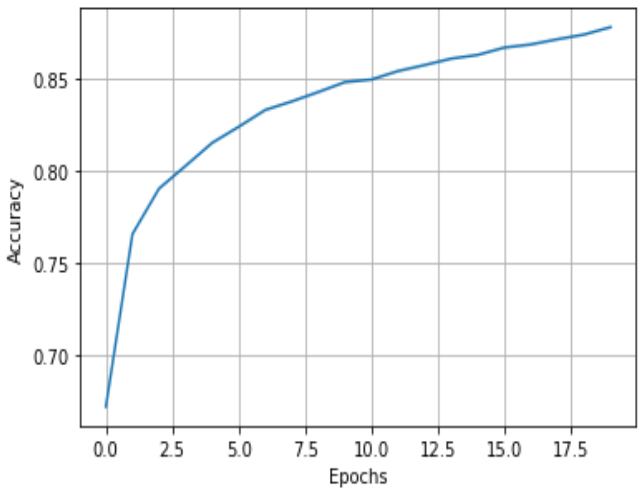

Fig. 8 Tiny VT model accuracy plot against the epochs for the fashion mnist dataset.

The accuracy and loss obtained with each of the vision transformer model for each of the dataset is tabulated in Table *3*. Unlike CNN models, the vision transformer models have performed almost equally for each of the architecture i.e tiny, small and base. For fashion mnist, the accuracy scores obtained by the tiny and small model is 0.90 and base model is 0.91. Similarly the accuracy scores for the cifar10 dataset is 0.78, 0.77 and 0.78 for tiny, small and base model respectively. In case of cifar100 dataset, the accuracy obtained with each model is 0.46. This signifies the fact that same performance is obtained for the 4, 6 and 8 layered transformer architecture and little variations is observed in both the accuracy and loss scores. The lowest loss score is obtained by the base model with the fashion mnist dataset which is 0.25 and the highest loss score is with cifar100 dataset for both the tiny and base model which is 2.03. The highest accuracy score is obtained with the fashion mnist dataset. As described earlier, the fashion mnist dataset has larger number of images than compared with the cifar10 and cifar100 dataset. This advocates the fact that the vision transformer model performs better when the number of training examples is larger. The lowest accuracy score is obtained with cifar100 dataset which is 0.46. This is expected since compared to the fashion mnist and cifar10 dataset, the cifar100 has 100 different classes and thus it is difficult for the model to correctly identify the classes for 100 object compared to 10 objects.

Table 3 Loss and Accuracy scores obtained with VT models.

Model	Fashion MNIST		CIFAR10		CIFAR100	
	Loss	Acc.	Loss	Acc.	Loss	Acc.
Tiny VT	0.27	0.90	0.63	0.78	2.03	0.46
Small VT	0.27	0.90	0.64	0.77	2.02	0.46
Base VT	0.25	0.91	0.63	0.78	2.03	0.46

D. CNN vs Vision Transformer

The performances obtained with CNN and vision transformer models are compared side by side in Table *4* where the accuracy score of each of the model obtained with each of the dataset is tabulated. From the table, it becomes clearly evident that the vision transformer is performing better than the CNN models for all the three datasets and all the three model architectures. For the fashion mnist dataset, the results for both the CNN and vision transformer are still comparable but the difference becomes clearly evident when the accuracy scores for the cifar10 and specially cifar100 datasets. The tiny CNN model achieves an accuracy score of 0.10 and 0.01 for the cifar10 and cifar100 datasets respectively. On the other hand, the accuracy score of tiny transformer model on the same datasets is 0.78 and 0.46, which deviates from that of the CNN model significantly. A similar trend is observed if the rest of the architectures are compared for the cifar10 and cifar100 datasets.

Table 4 Accuracy score for each of the three datasets and each of the six models.

Model	Fashion MNIST	CIFAR10	CIFAR100
Tiny CNN	0.90	0.10	0.01
Small CNN	0.89	0.58	0.27
Base CNN	0.88	0.67	0.31
Tiny VT	0.90	0.78	0.46
Small VT	0.90	0.77	0.46
Base VT	0.91	0.78	0.46

IV. Conclusion

In this research article, an important computer vision research area of image classification is explored. Various traditional techniques that have been utilized over the past are studied and their limitations have been identified. The recent advancement in machine and deep learning and their applications in the area of computer vision are discussed. The most popular deep learning algorithm known as CNN that is extensively used in many computer vision problems such as image classification, object detection, object tracking etc. Another popular algorithm that is recently used in NLP and have turned out to be very successful known as the transformer model is studied and its application in computer vision is discussed. Three CNN and vision transformer architectures are designed for the task of image classification which are called the tiny, small and base models. All these models are trained and evaluated on three benchmark datasets known as fashion mnist, cifar10 and cifar100. Analysis of the obtained results reveal that the vision transformer models have outperformed the CNN architectures in all three shapes. The transformer architectures performed particularly well in case of the cifar10 and cifar100 datasets. CNN architectures performed well only on fashion mnist dataset.

REFERENCES

[1] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *J. Physiol.*, vol. 195, no. 1, 1968, doi: 10.1113/jphysiol.1968.sp008455.

[2] Y. LeCun *et al.*, “Backpropagation Applied to Handwritten Zip Code Recognition,” *Neural Comput.*, vol. 1, no. 4, 1989, doi: 10.1162/neco.1989.1.4.541.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, vol. 2.

[4] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, 2015, doi: 10.1007/s11263-015-0816-y.

[5] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8689 LNCS, no. PART 1, doi: 10.1007/978-3-319-10590-1_53.

[6] M. Lin, Q. Chen, and S. Yan, “Network in network,” *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, pp. 1–10, 2014.

[7] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, vol. 2017-December.

[8] M. Chen *et al.*, “Generative pretraining from pixels,” in *37th International Conference on Machine Learning, ICML 2020*, 2020, vol. PartF168147-3.

[9] M. Is, R. For, and E. At, “An image is worth 16x16 words,” *Int. Conf. Learn. Represent.*, 2021.

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020, vol. 12346 LNCS, doi: 10.1007/978-3-030-58452-8_13.

[11] S. Zheng *et al.*, “Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers,” 2021, doi: 10.1109/CVPR46437.2021.00681.

[12] H. Chen *et al.*, “Pre-trained image processing transformer,” 2021, doi: 10.1109/CVPR46437.2021.01212.

[13] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, “End-to-End Dense Video Captioning with Masked Transformer,” 2018, doi: 10.1109/CVPR.2018.00911.