Image Classification Using Convolutional Neural Networks

Prayas Patnaik
Erik Jonsson school of Engineering and Computer Science
University of Texas at Dallas Richardson, Texas
pxp210038@utdallas.edu

Sirisha Satish
Erik Jonsson school of Engineering and Computer Science
University of Texas at Dallas Richardson, Texas
sxs210095@utdallas.edu

***Abstract*—In this report, we discuss Convolutional Neural Networks (CNN) which is a type of artificial Neural Network in the field of Deep Learning, that is extensively used for image or any object recognition or classification.It has less dependence on pre processing, minimizing the human efforts in the development of its functionalities.The performance of CNN for image classification tasks has outperformed Neural Network performance.**

***Index Terms*—Image recognition, CNN, Deep Learning.**

I. Introduction

Convolutional neural networks (CNNs) are a sub part of ANN. ANN stands for Artificial Neural Network. It has obtained popularity vision tasks. It is a special kind of neural network that is developed with the potential to extract distinctive features from visual data. They have the power to identify useful information in visual data, they are used in face detection and recognition. It belongs to a specific type of neural network that is designed to be able to extract unique characteristics from image data.
CNNs use numerical data just like other kinds of neural networks. The images that are sent into these networks must therefore be transformed into a numerical representation. Since images are composed of pixels, they are numerically transformed before being sent to the CNN.
Using multiple layer types, convolutional layers, and fully connected layers. CNNs can detect and accommodate spatial characteristics of image through backpropagation.

II. Background work

CNN is type of artificial neural network designed to learn directly from image data. They are heavily used in perception problems and have advanced state-of-the-art in perception applications like image classification,object detection etc. They have been highly successful in text categorization using NLP as well. The patterns in the image data, such as edges, gradients ,shapes are very well recognized by convolutional neural networks that's why CNNs are highly effective in computer vision tasks.It is a feed-forward neural network,which can have layers up to 25.

We have worked on the image classification problem, in which we are trying to identify the objects in the images. We have used the CIFAR10 dataset for our problem. It is a small image dataset and consists of 60000 32x32 color images and are labeled with 10 classes. The main disadvantage of Neural Networks is that the number of trainable parameters increases as the depth increases. A large number of parameters results in larger training times, higher chances of overfitting, etc. This is where CNN comes to the rescue and helps us to tackle these problems. Neural Networks does not use the information in the data because flattening operation always leads to loss of data.The spatial features of the image is lost.The earliest layer of the network should focus on the local region with out the regard for the contexts of the image in the distant regions. This is called locality principle. Eventually, the local representation can be aggregated to make predictions at the image level.Feature map is a container which keeps featured captured from different channels and used for predictions. Convolution Neural Network and CNN extract useful information from the image using convolution, pooling operations, and store that in feature maps and then pass that valuable information to the neural network for training. In this way, it solves the problem of training time and overfitting and gives better results than neural networks.

III. Theoretical and Conceptual study

Human brain has the capability of processing large chunks of data.In human brain each neuron is connected to another neuron and occupies the perceptive field collectively.Each neuron has receptive field .Receptive field is a region in the sensory region of the brain where electrical response can drive a neuron. The electrical response has to be above certain threshold value to fire the neuron in the brain.This working of brain has inspired scientist to create a structure to simulate that in the form of neural network and CNN.CNN uses different filters for catching valuable information in an image like edge detection filter etc. The convoluted output can pick up on lines, curves, and other simpler patterns before moving on to more complex patterns like faces and objects. By using a convolutional neural network, one may provide computers sight.

- **Convolutional Neural Network Architecture** - Convolution Neural Network has three layers:

– Convolutional Layer
– Pooling Layer
– Fully Connected Layer

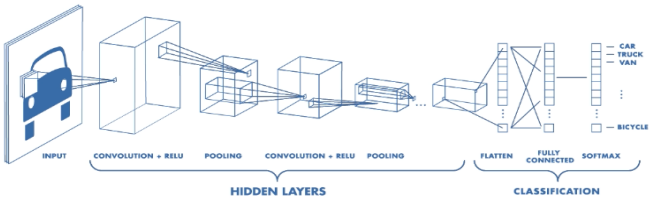


Fig. 1. Architecture of a CNN

a) **Convolution Layer** -
Convolution layer is the fundamental part of the CNN.It performs significant amount of the computation of the network. It performs matrix dot product operation between input image matrix and the filter weights. Filter weights is also known as kernel weights. It consist of learnable parameters and the other is the image patch of the image also known as receptive field. Alth.

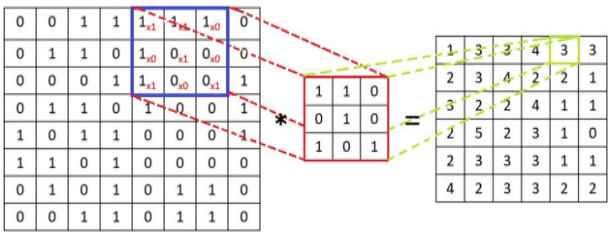


Fig. 2. Convolutional Layer

In the forward propagation operation, the kernel/filter traverses along the image's height and width and outputs the receptive field of the image. The sliding size of the kernel is known as a stride. As a result, a tow dimensional version of an image knows as the an activation map is produced, which indicates the kernel's behavior at each point in the image.

b) **Pooling Layer** -
By generating a summary statistic from the neighboring outputs, the pooling layer substitutes the network's output at specific locations. As a result, the representation's spatial size is reduced, which reduces the amount of computation and weights needed. Every slice of the representation is individually handled for the pooling operation.
Several pooling algorithms are available, such as the L2 norm. The mean of the neighborhood matrix , and a weighted mean distance from the central pixel.Max Pooling is the most widely used operation to downsample the input which reports the highest value from the matrix.

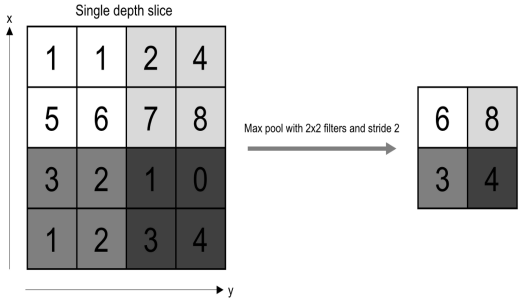


Fig. 3. Pooling Layer

c) **Fully Connected Layer** -
In a FCNN, all nodes in the layer are fully connected to all nodes in the previous and the corresponding next layer. The output from the previous layer is flattened before going into the FC layer as the input. This is implemented using flatten function in python.It is a matrix operation of laying out the the number across all axis in a sequence.

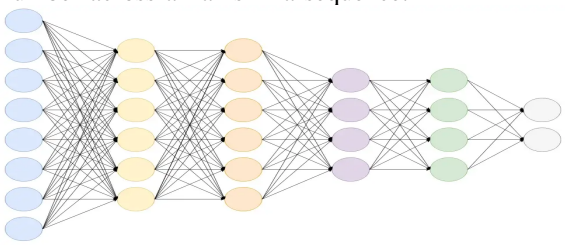


Fig. 4. Fully Connected Layer

The aligning of the representation of input and output is performed by the FC layer.

- **Transfer Learning**
Transfer learning is a term used in machine learning to describe the use of a previously trained model for another problem. In transfer learning, machines use information obtained from previous projects to improve their predictions about new projects. Transfer learning transfers the knowledge of an already trained machine learning model to another closely related problem.
The most significant benefits of transfer learning include reduced training time, better neural network performance, and a lack of substantial amounts of data. Transfer learning is useful in situations when access

to large amounts of data is not always feasible but is required to train a neural network from scratch.
Transfer Learning is used when there is not enough annotated data to train our model with and, when a pre-trained model that was developed using comparable data and tasks that already exists.

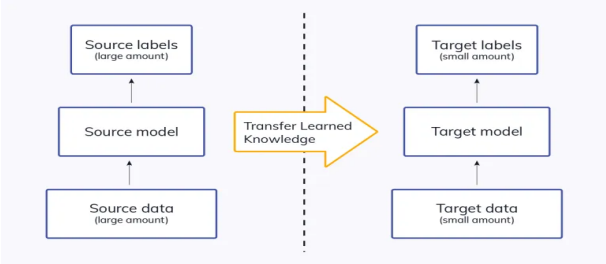


Fig. 5. The idea of transfer learning

i) **VggNet**
We have used VggNet architecture for our project. VGG - Visual Geometry group, it is a conventional deep Convolutional Neural Network architecture with numerous layers. This was proposed by K. Simonyan and A. Zisserman in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The digits after the term VGG denotes the number of layers.16 and 19 layers.The VGGNet, designed as a deep neural network and has outpperformed other architectures like LeNet AlexNet etc. It is still being used heavily in the industry for perception tasks.
Convolutional filters are used in the creation of the VGG network.
We have used the VGG19 architecture to train our model. This CNN is 19 layers deep with eights, formed by 16 convolutionals and 3 fully connected(FC) layers and classifies images based on the dataset.The popular dataset for testing the working of these architecture is imageNet. This dataset has an image size of 224x224 and 3 channels with its mean RGB value subtracted.

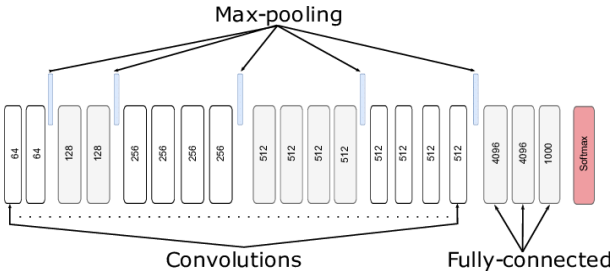


Fig. 6. Architecture of the network VGG19

IV. Implementation Details

A. Approach

In this project, we have implemented the building blocks of CNN from scratch. Images can be classified into two broad categories single channel(grayscale) images or multichannel images(RGB) images. We have tried to do image classification on grayscale images or to convert the multichannel images to grayscale and then use it for classification. The task is to use minimal information and accurate predictions.

B. Technical Details

We have implemented the process in two parts:

- Grey Scale Conversion of the Dataset. We have used the following equation for conversion:

$$I = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (1)$$

- **Design Rule** - Designed sequentially and call the respective functions during forward propagation and backward propagation. The calling order of the functions during backpropagation is exactly the opposite of forward propagation. Creating separate classes for Convolutional, Pooling, Softmax, FC layer, and Activation operations. Each of the classes contains two basic operations
  - Forward Propagation
  - Backward Propagation
- **Convolution Operation** - The class implemented for Convolution is ConvolutionLayerOperation. It contains 4 functions. The constructor of the class takes two variables and those are the size of the filter and the number of filters for convolution. The second function is the ImagePatch function which is used for taking out the slice from the input Feature map. It passes the slice and the indexes of both vertical and horizontal directions to the forward propagation function for convolution operations. It convolves the slice with the filter and adds it to the output. The BackPropagation function receives the error Matrix from the previous operation and uses that to calculate the errorMatrix for the current Layer and update the filter weights using it.

$$w = w - \alpha * dloss \quad (2)$$

- **Pooling Operation** - This operation is mainly for down sample the incoming input. The constructor of the class requires two parameters. The parameters are the size of the filter and the type of pooling. We have included max pooling and average pooling. The class has an image Patch function that gives a slice of the input to the forward propagation function which computes the max or mean of the slice based on the type of pooling chosen and appends it to the output. In this step, there is no updation in the filter weights. We calculate the error and pass it to the next operation.

- **Fully Connected Layer** - This class comes into the picture when we are done with multiple convolutions

and pooling operations and want to pass the input to the neural network. Usually flattening operation and multiplying the weights with the input and adding bias comes within the FC layer operations. We have defined separate classes for flattening the matrix. so that the class can help us in reshaping back the flat matrix for the computation of losses during backpropagation.

- **Activation Layer** - Activation class for introducing non-linearity to the network and better gradient calculations as well. We have three choices for activation function (ReLU,Sigmoid and Tanh).

$$sigmoid = \frac{1}{1+e^{-x}} \tag{3}$$

$$tanh = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{4}$$

$$ReLu = max(0,x) \tag{5}$$

The class contains member functions which calculates the first order derivative of the activation function and uses that for the calculation of gradients in backpropagation.

- **Softmax Layer** - This is the last layer in our architecture. It calculates the softmax output and basically returns the probability distribution of the classes. The class with the highest probability is the predicted class for the input Image.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K \tag{6}$$

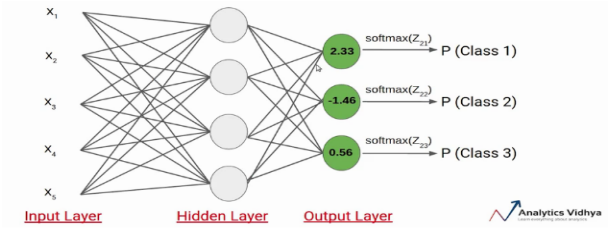
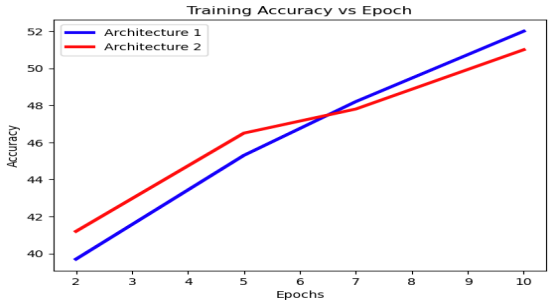


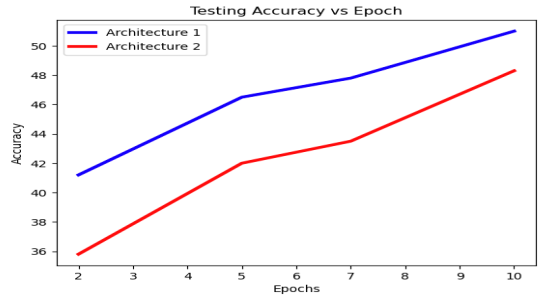
Fig. 7. Softmax

- **Transfer Learning** - We have used vgg19 architecture for comparison with our CNN model. Firstly, we have used the pre-trained model for training and then unfreeze few layers and trained for multiple epochs.

V. Results and Analysis

We have implemented CNN with two different Configurations and compared their accuracy with respect to Transfer Learning. Based on our results, learning rate plays an important role for the overall performance of the architecture. if the learning rate is high , the network does not converge properly and the bounces back and forth and failing to find the optimal solution. upon reducing the learning rate , the network starts performing better and gives better accuracy. Increasing the number of filters helps us to capture different spatial characteristics of the image and helps to classify better. Transfer learning gives training accuracy of 50.19 and testing accuracy 53.24 using Adam optimizer and 0.05 learning rate without unfreezing any layer. Transfer Learning results are better than the scratch implementation of CNN by 4.74%





CNN Performance Table			
Architectural Details	**Epoch**	**Training Accuracy**	**Testing Accuracy**
No. of Filters - 10 Size of Filter - 3x3 Pooling Matrix - 2x2 No. of Hidden Layers - 50 Learning Rate - 0.05	2	39.7	33.4
	5	45.3	42.9
	7	48.2	45.7
	10	52	47.6

Fig. 8. Results Table - I

CNN Performance Table			
Architectural Details	**Epoch**	**Training Accuracy**	**Testing Accuracy**
No. of Filters - 20 Size of Filter - 3x3 Pooling Matrix - 2x2 No. of Hidden Layers - 100 Learning Rate - 0.05	2	41.2	35.8
	5	46.5	42
	7	47.8	43.5
	10	51	48.3

Fig. 9. Results Table - II

VI. Conclusion and Future Scope

The implementation is a basic CNN architecture with limited freedom. We can include stride parameters while calculating the convolution output and make the code compatible with

multi-channel images and add different optimizers like Adam, momentum, etc for filter weight calculations, and include a learning rate scheduler to adjust the values of the learning rate during training to reach global minimum and not stuck on local minima.

REFERENCES

[1] Bansal, M., Kumar, M., Sachdeva, M., Mittal, A. (2021). Transfer learning for image classification using VGG19: Caltech-101 image data set. Journal of Ambient Intelligence and Humanized Computing, 1-12.

[2] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O. Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8(1), 1-74.

[3] S, Premanand. "Convolution Neural Network - Better Understanding!" Analytics Vidhya, 16 July 2021, https://www.analyticsvidhya.com/blog/2021/07/convolution-neural-network-better-understanding/.

[4] Parekh, Manan. "A Brief Guide to Convolutional Neural Network(CNN)." Medium, 16 July 2019, medium.com/nybles/a-brief-guide-to-convolutional-neural-network-cnn-642f47e88ed4.

[5] Hussain, M., Bird, J. J., Faria, D. R. (2018, September). A study on cnn transfer learning for image classification. In UK Workshop on computational Intelligence (pp. 191-202). Springer, Cham.

[6] Benhur, S. (2021, March 3). CNN architectures from scratch. DataDrivenInvestor. https://medium.datadriveninvestor.com/cnn-architectures-from-scratch-c04d66ac20c2

[7] "Stanford University CS231n: Deep Learning for Computer Vision." Stanford University CS231n: Deep Learning for Computer Vision, cs231n.stanford.edu. Accessed 29 Nov. 2022.

[8] Albawi, S., Mohammed, T. A., Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). Ieee.

[9] Chauhan, R., Ghanshala, K. K., Joshi, R. C. (2018, December). Convolutional neural network (CNN) for image detection and recognition. In 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC) (pp. 278-282). IEEE.

[10] Jenny. "Basic Convolutional Neural Network (CNN) Architecture." Medium, 18 Feb. 2021, medium.com/helyx/basic-convolutional-neural-network-cnn-architecture-646543e416d2.