International Journal of Computer Science Trends and Technology (IJCST) – Volume 10 Issue 5, Sep-Oct 2022

RESEARCH ARTICLE OPEN ACCESS

Image Classification using small Convolutional Neural Network

Mrs J Sarada,MCA,M.Tech,M.Phil,(Ph.D) [1], P Priya Bharathi [2]

[1] Associate Professor, Department of Computer Application

[2] Student, Department of Computer Application

[1], [2] Chadalawada Ramanamma Engineering College (Autonomous)

ABSTRACT

With an upsurge in the rate of data production, pervasive usage of cameras for automation and surveillance and the requirement of visual input for artificially intelligent devices all across the globe, there has been a rapid increase in the mass of image data being generated today. This gives rise to the essentiality of automated image processing required to simplify image related tasks. Automated image processing bridges the gap between the human visual system and the pixel level data of images. Deep Convolution Neural Networks are being deployed expansively to analyze, detect and classify images for a diverse number of tasks. These neural networks, similar to the human neural network, contain neurons with learnable weights and biases, which are trained to identify and classify different objects or features across the image. This paper presents a functional implementation of image recognition using a small convolutional neural network, proposing less complexity and yielding good classification accuracy for all tested data sets.

Keywords: - image Classification, Object Detection, Deep Learning, Convolutional Neural Networks

I. INTRODUCTION

The utilization of software for digital image processing hassoared due to the drastic rise in the volume of imagesdatabase, wider availability of cost-effective image databasesand the need for human-level object classification accuracy.Enormous progress has been accomplished in the field ofimage recognition and classification using deep convolutionalneural networks and machine learning in the past few years.The human neural system consists of a vast interconnectednetwork of neurons that communicate and exchange inputswith each other for processing the information around us.Convolutional Neural networks (CNNs), analogous to humanneural system, contain neurons with learnable weights andbiases, as shown in figure 1. This system is trained with avariety of data sets to extract, analyze and classify visualpatterns from image pixels. For example, Content BasedImage Retrieval neural networks are capable of extractingvisual features of image data such as patterns, edges, colors,shapes et-cetera and classifying these features to determinevisually similar objects or images. Convolving the image layerby layer is the principal behind image recognition using thisneural network. CNNs broadly incorporate convolutionallayers, pooling layers, hidden layers and fully connectedlayers. The convolutional layer embodies a set of selfsufficientfilters and every filter is autonomously convolvedwith the input image. We start by choosing a filter and sliding dot product of the filter and each part of the image simultaneously. These filters become classifying parameters that are learned by the CNN. The output from one convolutional layer, called feature map, is passed on further to other convolutional layers for deeper convolution. The weighted sum of input values are passed to the activation function, which determine the output of a given neuron after a given set of inputs [1]. Pooling layers work on each feature map independently and reduce the overall computations and avoid overfitting by decreasing the spatial size of the representation matrix. This layer removes redundancy and smoothens computations. The fully connected layers are responsible for classifying and mapping the learned features into the sample datasets. The optimization of the network model and increased accuracy can be achieved with the help of loss functions. These functions determine the degree of variability between the predicted and observed values. Smaller loss functions represent better models [2]. The input, which is linearly transformed by the neuron's weights and biases and non-linearly transformed by the activation function is then passed to the hidden layers for further processing to obtain the output, giving rise to feed forward propagation.

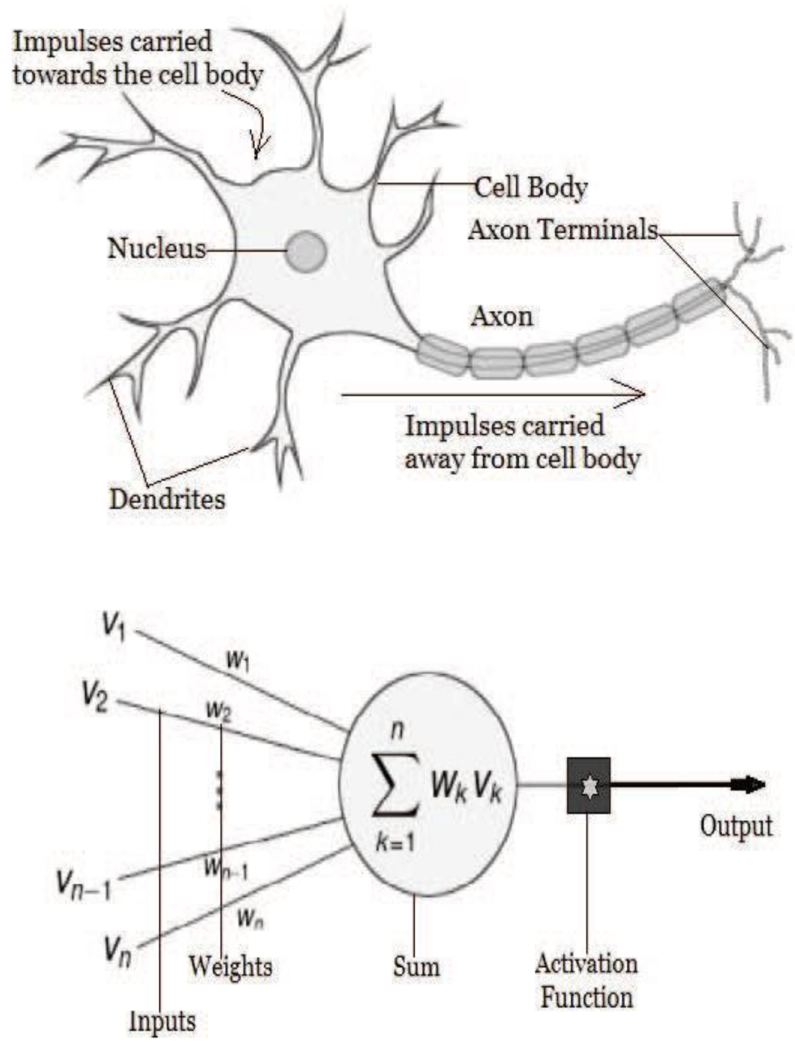


Fig.1 A Biological Neuron & an Artificial Neuron

We have proposed a simple Convolution Neural Network forimage recognition. Our model consists of 200 classes with 13layers, namely, 5 Convolution layers, 3 Max Pooling layers, 1Dropout layer, 1 flattening layer, 2 Fully Connected layersalong with one Softmax Layer. The dataset has been dividedinto the three categories of training, validation and test-set toavoid overfitting. Initially, our training accuracy increasesrapidly while rate of increase of validation accuracy is not sohigh. Meanwhile our validation loss doesn't follow the lineardecrease, instead it sometimes decreases and sometimesincreases.

II. RELATEDWORKS

Machine learning frameworks have made the demanding taskof implementing machine learning models much simpler.These frameworks help in acquiring datasets and providepretrained models with better refining. One such framework isGoogle's TensorFlow. Released in 2015, it's an open-sourcemachine learning library that allows dataflow programmingover various platforms. It can be implemented in multiplelanguages such as C#, C++, Java and R. TensorFlow'sprinciple lies in deploying tensors to power the learning. Thisframework is capable of training deep neural networks (In ourcase, a Convolutional Neural Network) to perform copioustasks which can solve real world problems, such as imagerecognition and classification, word embeddings, speechrecognition, sentiment analysis, natural language processingand so on [3].

TensorFlow enables the users to illustrate the movement ofdata through a progression of processing nodes. Distinctmathematical computations are represented by distinctindividual nodes in the graph and nodes are connected to eachother with edges, forming a network. Edges aremultidimensional data arrays, also known as tensors. Thesemathematical computations are written in C++ and the nodesas well as the tensors are python objects. TensorFlow allowsits users to design neural networks line by line, using pythonto conveniently couple high-level abstractions.Supervised learning, Unsupervised learning andReinforcement learning can be categorized as the threedifferent classes of machine learning algorithms. Our modelimplements supervised learning.In supervised learning, the dependent (target) variable is to bepredicted using independent (predictor) variables with the helpof mapping functions which map inputs to the desired outputs.The algorithm forges predictions based on the identifiedpatterns and observations over the cycle of training which iscontinued until the desired accuracy is reached. In simplewords, this category of machine learning algorithms trains themachine with the help of examples and the model is based onboth input and output data [4], as presented in figure 2.Classification and Regression are the two implementations ofsupervised learning.

III. PROPOSED SYSTEM ARCHITECTURE

We built a Convolution Neural Network for imagerecognition. Our dataset contains 200 classes (Subset ofImageNet dataset). Our model has 13 layers and consists of 5Convolution layers out of which 3 are followed by 1 MaxPooling Layer, 1 Dropout Layer, 1 Flattening layer and 2Fully Connected Layers along with one Softmax Layer.The dataset has been divided into 3 categories: Training,validation and test-set to avoid overfitting. The convolutionallayers are victualled with input images one after the otherfrom all the classes, during training. After convolution, someneurons are dropped at the Dropout rate of 0.8 and the outputis flattened before being finally passed to the fully connectedlayers. The number of outputs in the second fully connectedlayer match the number of classes which represent theprobability of an image for each class.The model is retained during training and is utilized to operateon our input image dataset to predict whether the given imagebelongs to the classes our model is trained on. Since labels areinput along with the training image dataset during training, theaccuracy achieved during training will be greater than thevalidation accuracy.

However, it is important to report thetraining accuracy in every iteration so as to be constantlyameliorating the accuracy in the training dataset. Eachiteration, or Epoch, ends with saving the model for re-iterationand reporting the accuracy. The input image is required to beread and pre-processed in a manner identical to the training, soas to obtain predictions. The saved model is restored, and thevalues of weights and biases learned from the previousiterations are used to predict the probability of the input imagebelonging to every class. We built a Convolution Neural Network for image recognition. We take only 200 classes. The model will identify and separate images of different classes. Here Instead of initializing our parameters with zeros, we initialize them with Random Normal distribution with mean 0 and very small standard deviation of 0.05.

We use Softmax Classifier to convert scores of each class into probability distribution and then use the Cross Entropy as ourloss function. For calculating the gradient and optimizing theweights, we deploy AdamOptimizer. We are trying to achieveminimum cost with a learning rate of 0.0001 (1e-4).The training consists of 100 Epochs with an input of minibatch of 32 images using dataset class which provides nextbatch of images for training. After each epoch, we calculatetraining accuracy, validation accuracy and the validation loss(value of cost function).

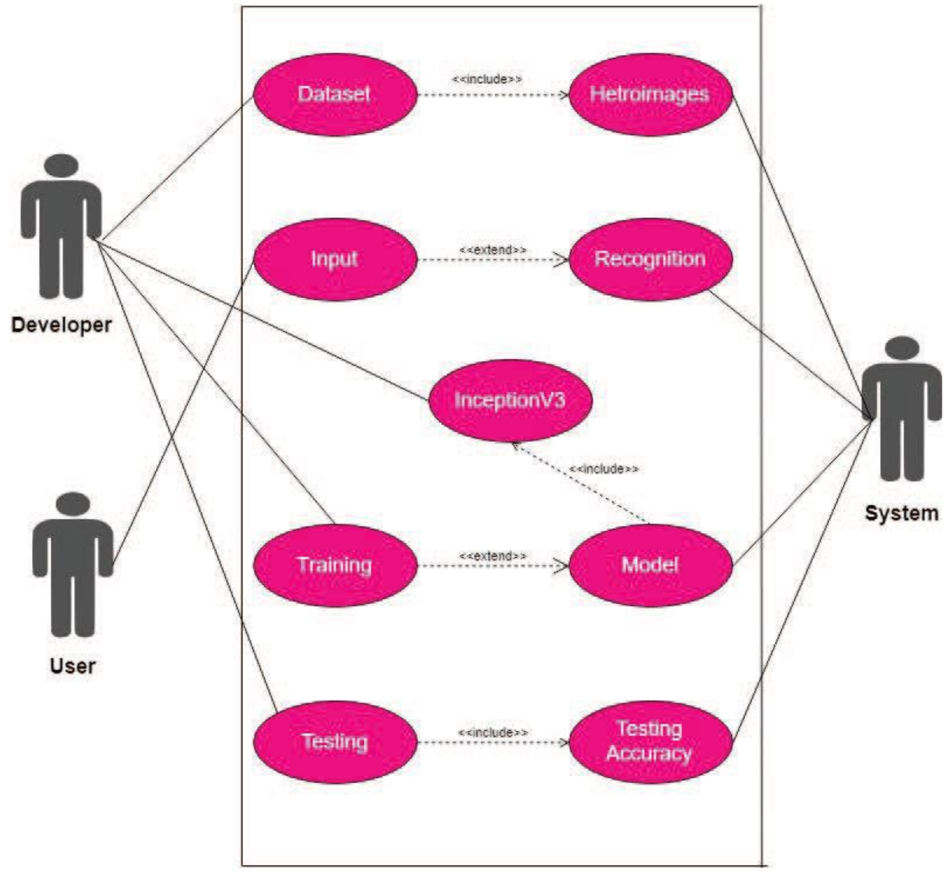


Fig.2 ML diagram of the proposed system

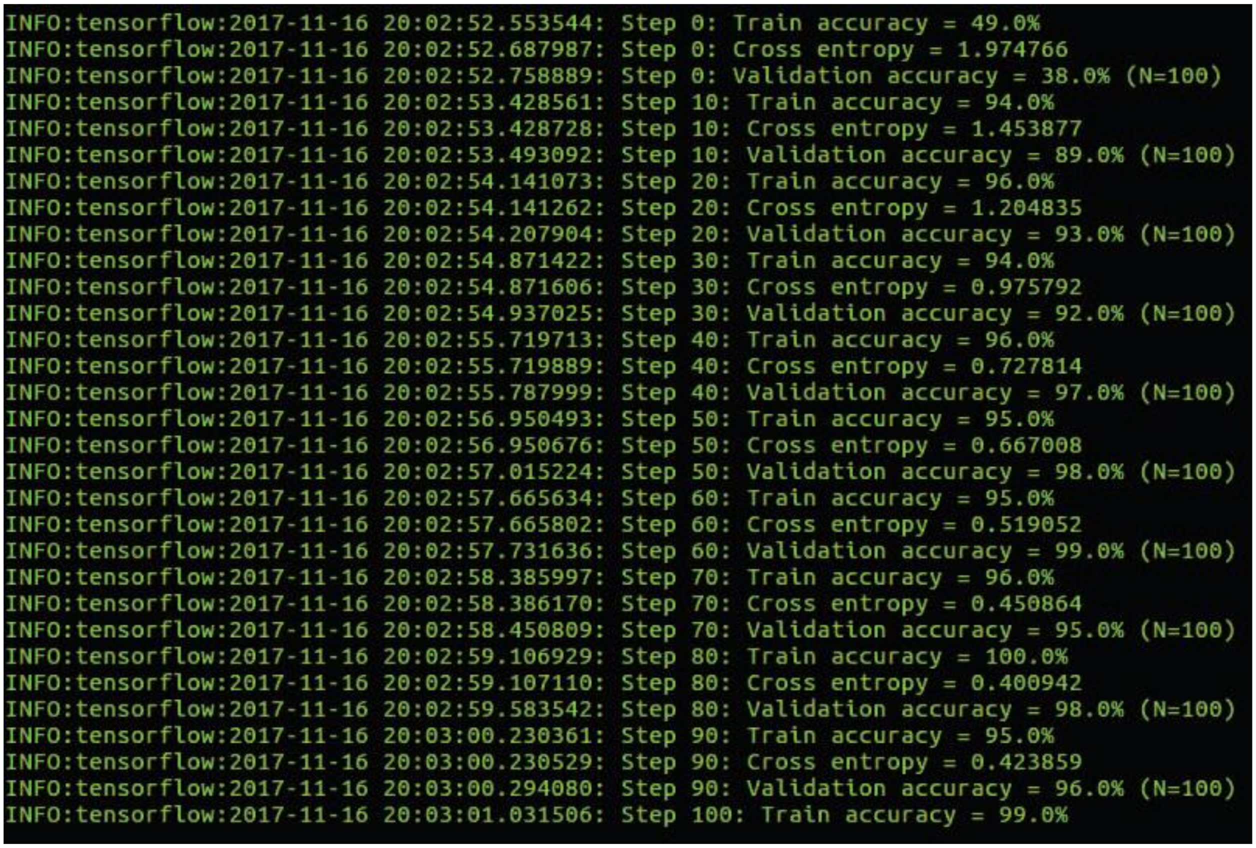
IV. RESULTS AND DISCUSSION

Our model started with low training accuracy as expected dueto vast number of classes but as the model trained itself, theaccuracy started to increase. By 30 Epochs our model reached50% training accuracy, a maximum of 45% validationaccuracy and a maximum of 5.1 validation loss.

We trained our model for more Epochs and finally after ourmodel reached 99% training accuracy and a minimum of 0.12validation loss, as shown in figure 5, which is a goodachievement considering the number of classes and the size ofdataset per class. 100000 images from 200 classes (500 per class) are taken as input to train the network and are fed to the first Convolution Layer. The convolution layer itself consists of convolution followed by max pooling. The first convolution layer consists of 32 filters of size of 3*3 and gives an activation map of 64*64*32 which is fed into max pooling with stride 2,2 decreases the height and width of

the image by half. We then apply ReLU Activation function to introduce non-linearity to the model. The output from previous layer is fed to the second convolution layer and whose output is fed into third convolution layer and the previous process is repeated. The output from these layers are fed into fourth convolution layerand fifth convolution layer where number of filters areincreased to 64 giving an activation map of 8*8*64. The output is now passed to dropout layer with dropout rate of0.8. Dropout is applied to 'drop-out' or obviate randomlyselected neurons during training. When these randomlyselected neurons are 'dropped-out', weight updates are notapplied to these neurons on the backward pass and they do nothelp in the activation of downstream neurons on the forwardpass temporarily.The output of the dropout layer is fed into the flattening layerto convert the multi ranked tensor to a 1 rank tensor which is

fed into first fully connected layer with 128 neurons andReLU function is applied to the output of the first fullyconnected layer which is then fed to the second fullyconnected layer which then passes the data into softmaxclassifier which then finally gives the class of the imageprovided to the network. We run the model with learning rate of 1e-4 and we use AdamOptimizer to minimize the costfunction and the values of weights and biases are updatedthrough backpropagation.



```
INFO:tensorflow:2017-11-16 20:02:52.553544: Step 0: Train accuracy = 49.0%
INFO:tensorflow:2017-11-16 20:02:52.687987: Step 0: Cross entropy = 1.974766
INFO:tensorflow:2017-11-16 20:02:52.758889: Step 0: Validation accuracy = 38.0% (N=100)
INFO:tensorflow:2017-11-16 20:02:53.428561: Step 10: Train accuracy = 94.0%
INFO:tensorflow:2017-11-16 20:02:53.428728: Step 10: Cross entropy = 1.453877
INFO:tensorflow:2017-11-16 20:02:53.493092: Step 10: Validation accuracy = 89.0% (N=100)
INFO:tensorflow:2017-11-16 20:02:54.141073: Step 20: Train accuracy = 96.0%
INFO:tensorflow:2017-11-16 20:02:54.141262: Step 20: Cross entropy = 1.204835
INFO:tensorflow:2017-11-16 20:02:54.207904: Step 20: Validation accuracy = 93.0% (N=100)
INFO:tensorflow:2017-11-16 20:02:54.871422: Step 30: Train accuracy = 94.0%
INFO:tensorflow:2017-11-16 20:02:54.871606: Step 30: Cross entropy = 0.975792
INFO:tensorflow:2017-11-16 20:02:54.937025: Step 30: Validation accuracy = 92.0% (N=100)
INFO:tensorflow:2017-11-16 20:02:55.719713: Step 40: Train accuracy = 96.0%
INFO:tensorflow:2017-11-16 20:02:55.719889: Step 40: Cross entropy = 0.727814
INFO:tensorflow:2017-11-16 20:02:55.787999: Step 40: Validation accuracy = 97.0% (N=100)
INFO:tensorflow:2017-11-16 20:02:56.950493: Step 50: Train accuracy = 95.0%
INFO:tensorflow:2017-11-16 20:02:56.950676: Step 50: Cross entropy = 0.667008
INFO:tensorflow:2017-11-16 20:02:57.015224: Step 50: Validation accuracy = 98.0% (N=100)
INFO:tensorflow:2017-11-16 20:02:57.665634: Step 60: Train accuracy = 95.0%
INFO:tensorflow:2017-11-16 20:02:57.665802: Step 60: Cross entropy = 0.519052
INFO:tensorflow:2017-11-16 20:02:57.731636: Step 60: Validation accuracy = 99.0% (N=100)
INFO:tensorflow:2017-11-16 20:02:58.385997: Step 70: Train accuracy = 96.0%
INFO:tensorflow:2017-11-16 20:02:58.386170: Step 70: Cross entropy = 0.450864
INFO:tensorflow:2017-11-16 20:02:58.450809: Step 70: Validation accuracy = 95.0% (N=100)
INFO:tensorflow:2017-11-16 20:02:59.106929: Step 80: Train accuracy = 100.0%
INFO:tensorflow:2017-11-16 20:02:59.107110: Step 80: Cross entropy = 0.400942
INFO:tensorflow:2017-11-16 20:02:59.583542: Step 80: Validation accuracy = 98.0% (N=100)
INFO:tensorflow:2017-11-16 20:03:00.230361: Step 90: Train accuracy = 95.0%
INFO:tensorflow:2017-11-16 20:03:00.230529: Step 90: Cross entropy = 0.423859
INFO:tensorflow:2017-11-16 20:03:00.294080: Step 90: Validation accuracy = 96.0% (N=100)
INFO:tensorflow:2017-11-16 20:03:01.031506: Step 100: Train accuracy = 99.0%
```

Fig.3 99% training accuracy achieved by the proposed model

V. FUTURE SCOPE AND CONCLUSION

We can use our model for transfer learning and train on newclasses by crawling images from internet and retraining ourmodel on those classes.We can increase number of classes in our model to up to 1000classes and train it on full ImageNet dataset (256*256px14,197,122 images).We can increase our accuracy as follows:

1. Including the Feature extraction on our model instead ofgiving raw input images to model.
2. We can increase our dataset from 500 images per class to atleast 1000 images per class but at the cost of time.
3. Image Preprocessing - Randomize data by rotating theimage, increase/decrease contrast and brightness, shifting theimage from its axis.
4.We can decrease the time taken to train our model bytraining it on distributed GPU's and faster CPU with highmemory.

Our model is able to 99% accuracy which is goodachievement. The model reaches 96% Validation accuracy.The Validation loss goes up to 0.12. Initially our model sawoverfitting to a great extent which was reduced by introducinga dropout layer before the flattening Layer. The validation losscould be decreased by introducing randomization in dataset.Even though the number of classes were high, and the numberof images were less, our model achieved a good accuracy inthe end with only 13 layers in the architecture. We could savethe model and use it for prediction of images belonging to anyof these 200 classes and get the probability score for imagebelonging to each class.Our model saw the dataset for first time and was trained fromscratch, so it required a greater number of Epochs to achieve agood accuracy which it did.

REFERENCES

[1] Yang, J., & Li, J. (2017). Application of deep convolution neural network. *2017 14Th International Computer Conference On Wavelet Active Media Technology And Information Processing (ICCWAMTIP)*. doi: 10.1109/iccwamtip.2017.8301485

[2] Amrutkar, B., & Singh, L. (2016). Efficient Content Based Image Retrieval Using Combination Of Dominant-Color, Shape And Texture Features And K-Means Clustering. *International Journal Of Engineering And Computer Science*. doi: 10.18535/ijecs/v5i1.10

[3] Rampasek, L., & Goldenberg, A. (2016). TensorFlow: Biology’s Gateway to Deep Learning?. *Cell Systems*, *2*(1), 12- 14. doi: 10.1016/j.cels.2016.01.009

[4] Ma, J., Wen, Y., & Yang, L. (2018). Lagrangian supervised and semi-supervised extreme learning machine. *Applied Intelligence*. doi: 10.1007/s10489-018-1273- 4

[5] Multimedia Data Classification Using CNN. (2018). *International Journal Of Recent Trends In Engineering And Research*, *4*(4), 557-561. doi: 10.23883/ijrter.2018.4274.relqz

[6] A. Hirano, A. Tsukada, K. Kasahara, T. Ikeda and K. Aoi, "Research on construction and verification of a color representation system of a cataract to assist design", *The Japanese Journal of Ergonomics*, vol. 54, no., pp. 2B1-1-2B1- 1, 2018.