

Multi-Class Image Classification using CNN and Tflite

Vishal Shah^{1*}, Neha Sajnani²

^{1,2}Student, Department of Computer Engineering, Chandubhai S. Patel Institute of Science & Technology, Anand, India

*Corresponding author: visushah99@gmail.com

Abstract: In recent years' machine learning is playing a vital role in our everyday lifelike, it can help us to route somewhere, find something for what we aren't aware of, or can schedule appointments in seconds. Looking at the other side of the coin besides machine learning Mobile phones are equivocating and competing in the same field. If we take an optimistic view, by applying machine learning in our mobile devices, we can make our lives better and even move society forward. Image Classification is the most common and trending topic of machine learning. Among several different types of models in deep learning, Convolutional Neural Networks (CNN's) have intimated high performance on image classification which are made out of various handling layers to gain proficiency with the portrayals of information with numerous unique levels, are the best AI models as of late. Here, we have trained a simple CNN and completed the experiments on the dataset called Fashion Mnist and Flower Recognition, and also analyzed the techniques of integrating the trained model in the Android platform.

Keywords: Android, Convolutional Neural Network, Deep learning, Image classification, Tflite.

1. Introduction

Competency of using mobile devices in today's era has made a significant difference by its rapid increase in utilization in the last few years which thereby does not seem to degrade over the coming 20+ years. On the other hand, Machine learning /Artificial Intelligence is at the summit of making the world a new place with wider availability of data and advancements in storage, computing and analyzing capabilities. Correctly consolidating them, these 2 aspects can make several opportunities and possibilities for the upcoming new generation.

Image classification is one of the broader fields of machine learning. Image classification is the technique to map numbers to symbols. To classify a dataset into various distinct classes, their relationship i.e. between data and their classes must be understood well for this thing to get achieved the computer must be trained well.

The classification methods are divided into Supervised and Unsupervised approaches.

A. Supervised Approach

Supervised Approach is the Quantitative analysis of remote

sensing of image data. In this technique, the learning process is adopted to form a mapping set of variables to another set of variables. To compare this, we can say that a teacher is needed in the learning phase [6].

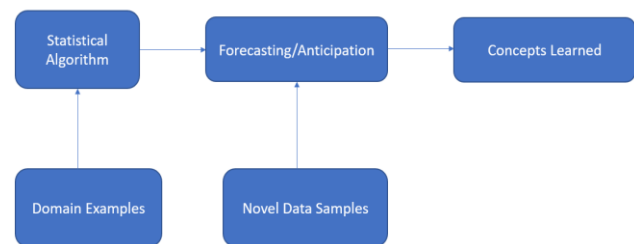


Fig. 1. Supervised Learning Approach

B. Unsupervised Classification

In unsupervised classification, the output is based on the software understanding and analysis without any user sample images. To compare this, we can say that a teacher is not needed in the learning phase [7].

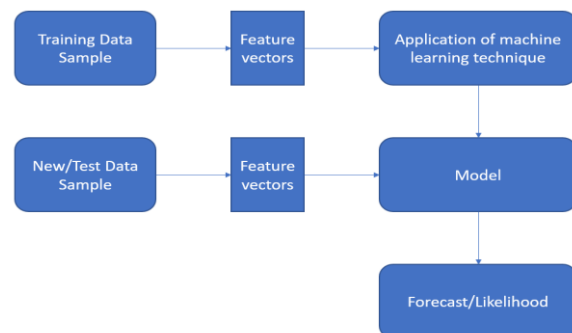


Fig. 2. Unsupervised Learning Approach

There are many machine learning algorithms developed by researchers for decades for classification of images like Naive Bayes Classifier, Nearest Neighbour, Support Vector Machines, Decision Tree, Random forest, Neural network, etc. So, there are 2 types of classification in a model.

C. Binary Classification

In this type of classification, the defined classes allowed are only 2 (e.g. A model that can predict whether the given data(image) is a cat or a dog).

D. Multi-Class Classification

In this type of classification, the defined classes allowed can be more than 2 (e.g. A model that can predict whether the provided data(image) is of a cat, dog or lion or more).

The most interesting thing one can do in this field is to apply machine learning concepts in mobile devices. E.g. Netflix, Google maps, etc. are ruling the world nowadays.

In this paper, we have proposed the various phases of the CNN algorithm with multi-class image classification on the android platform. Further Working of the CNN model is explained and integrating of the pre-trained multi-class model in android is demonstrated.

2. Related Work

Image classification is widely used in the field of machine learning nowadays. The two main companies involving machine learning most frequently are Facebook and Twitter which use the benefits to encourage the audience to tag people in their photos and much more [9]. The most important aspect of today's world "Security". Image classification further reaches the height to protect the world and the people in the most effective way. Previously, the people used to search or know other people by their names or by asking the physical contacts they have. To make this easier just think of an app that can help you to find people just by the images. The work in this field is still on.

Here we have proposed a little work on how image classification can be used to identify the images in a mobile device i.e. Android (here) also understanding the different image recognition fields. Image classification is used on the grayscale images that are not as effective as colored images are. Here the same work that can even classify a colored photo is demonstrated. And the integration phase is explained because the output on the console screen is not that effective as that on a handy device. Which can be used in any field for security or any social purpose.

3. CNN Model

Convolutional neural network has mainly 3 important layers called the Convolutional layer, pooling layer, and Fully-connected layer. Fig. 1 shows the LeNet-5 [1] architecture which was first introduced by Yann LeCun.

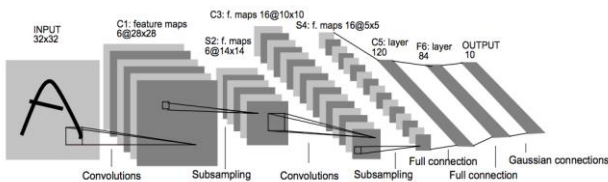


Fig. 3. LeNet-5 Architecture (Adapted from [4])

A. Convolution Layer

Convolution network has one of the most powerful biology-inspired artificial neural networks. The basic principles of the

neural network came from neural science. Convolution layer has many numbers of hidden layers. It is used for extracting features from an input image and then feeding it to the next layer. The most important parameter of this layer is kernels and the size of the kernels. The finest way to describe this layer is to imagine a matrix having a 4x4 dimension (called kernel or filters) that is on the top-left corner of the image and slide across the whole image and divided the image into vectors [2].

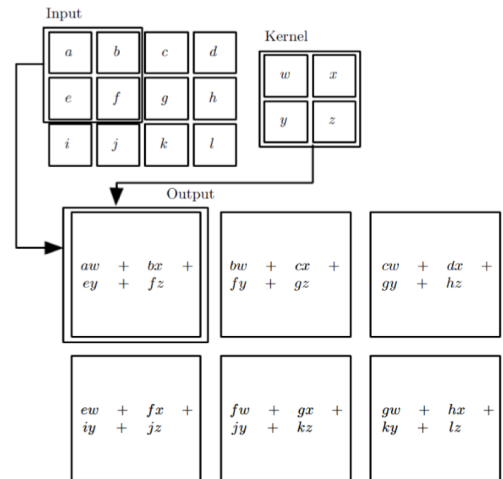


Fig. 4. Image Convolution

Imagine that a two-dimensional input image "I1" is taken as the input, and the same size of two-dimensional convolution kernel is represented by "K1"; the convolution of the input image is, as follows:

$$S(i,j)=(I1 * K1)(i,j) = \sum m \sum n (I1(m,n)K1(i-m,j-n))$$

The main purpose of this layer is to learn feature extraction from the inputs [5]. As the figure shown above, the Convolutional layer consists of several layers for extracting the features from the image [2]. For obtained new features, the input image matrix first convoluted with the kernel matrix and then the result sent to the non-linear activation function (i.e. ReLU, Sigmoid, etc.) that apply on every value of feature map [3].

B. Pooling Layer

This layer is added after the activation function that has been applied to the feature map output given by the convolution layer. The Pooling layer performs the 2 types of pooling operation Max Pooling [9] function takes the maximum value from the feature output matrix with different stride value and Average Pooling [10] which takes the average value of the matrix based on the stride value [5]. The main purpose of the Pooling function is to reduce the dimension of the matrix and increase the efficiency of the feature extraction.

C. Fully-Connected Layer

The main purpose of this layer is to combine all the features detected from the input image. The classifier has one or more

fully connected layers. This layer is placed before the convolutional output of CNN and the result of this layer is used to flatten the result before classification output came. For the classification SoftMax regression is most frequently used because this method gives the probabilistic distribution of output classes. Another method is Support Vector Machine (SVM) can also be used in place of CNN for classification [5].

4. Proposed Work

In the process of the depth convolutional network, it is obvious that the ability of the network will enhance as we adding up layers [5]. As we are adding more and more layers it will gain the accuracy of the network and correctly classify into given classes.

Based on this idea, we built a simple convolutional network to classify the images into multi classes. For that, we took the Fashion Mnist dataset to test our convolutional network. In this dataset, we have 60,000 training images and 10,000 testing images. We further divided the training images into validation images and then take 3 convolutional layers to train the model on those images.

1) Dataflow for a first convolutional layer

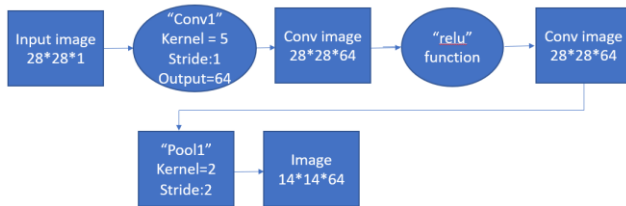


Fig. 5. First Convolution Layer

2) Dataflow for a second convolutional layer

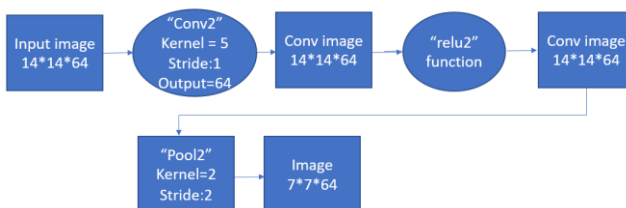


Fig. 6. Second Convolution Layer

3) Dataflow for a third convolutional layer

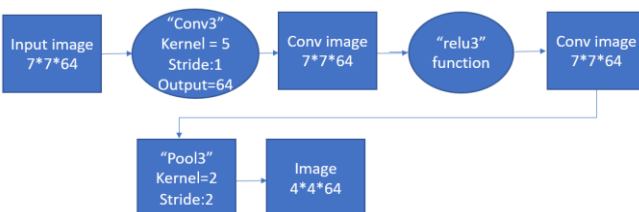


Fig. 7. Third Convolution Layer

After this 3-convolution layer, the output goes to the fully

connected layer and then the whole matrix was flattening and then gives the output in one of the classes.

As we tested our network on the Mnist dataset the accuracy we got is around 91% on training images and on validation images we got 89% accuracy. As these two accuracies are merely close to each other we conclude that our network was not overfitting on the data thus our network was performed very well on the dataset. In the figure below we plot a graph accuracy versus each epoch on the training and validation phases. It is clearly shown from the graph that accuracy increased by each epoch on both training and validation images.

A. Training Phase

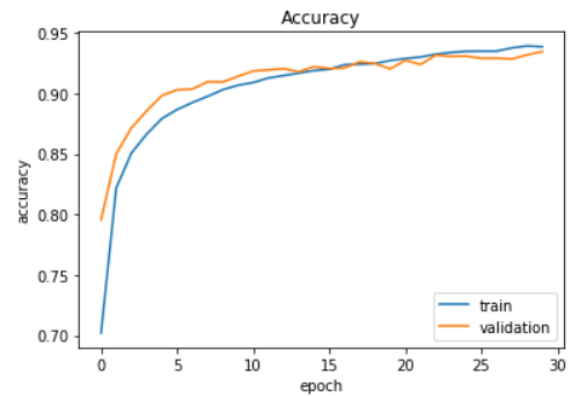


Fig. 8. Accuracy on training and validation phase

Further, we plot a graph also on the “loss” function. As we shown in the figure below the loss is continuously decreasing for both training and validation images so that we found good accuracy on these images.

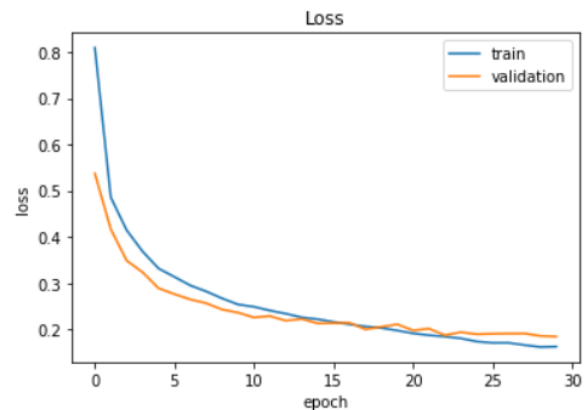


Fig. 9. Loss of training and validation phase

We were also tested our model on the human face dataset that took a colored image and classify it. Our created our dataset of the human faces and we get a moderate accuracy of 85% on the training dataset because we have a smaller number of the images for the dataset. Our model is useful for every multi-class classification of images so you can add any of your dataset for classification.

B. Testing Phase

Now, it's time to test our model on different images that we were not given to the model while training. On the test set, we gave a good accuracy of around 87% which is quite enough. Compared to the existing model which will work only on grayscale images while our proposed model also works for the RGB colored images that will take an original photo and classify it into the defined classes.

C. Integration Phase

Machine learning contributes to the ability for making your app learn and improve without being explicitly programmed to do so [13]. Android supports a wide area of machine learning tools and methods for everyone. To integrate a pre-trained model into an app there are several techniques. The most effective, secure and easiest way is to use Tflite (TensorFlow lite). Tflite is an open-source framework of deep learning which is used to run any pre-trained model on the IOS or Android platform [14].

A predictable workflow of tflite includes the following:

- Creating and training a model in python using TensorFlow
- Converting the TensorFlow model into tflite model using any tflite converter
- Deploying the model on a mobile device using tflite interpreter
- Optimizing the model accuracy and storage usage [15].

Table 1

Comparison of the existing platform and they're supported mobile OS and model format

ML platform	Mobile OS	GPU/CPU	Model Format
TensorFlow/lite	IOS, Android	Yes/Yes	TensorFlow
Caffe2	IOS, Android	Yes/yes	Caffe2, CNTK, Pytorch
CoreML	IOS	Yes/Yes	Keras, TensorFlow
Snapdragon	Android	Yes/Yes	Caffe, Caffe2, TensorFlow
Deep Learning Kit	IOS	Yes/Yes	Caffe

5. Conclusion and Future Work

This study focused on how to combine CNN trained models with android phones using Tflite. The paper also proposed some major concepts on the CNN model with Multiclass image classification. The paper also briefs the idea of how machine learning can be useful in the recent trending areas of the world focusing on security. It also presents some performance

measures for some algorithms. The convolutional neural network is the simplest way for the classification stuff and also reduced the computational cost. We can also use other algorithms for the classification but the convolutional neural network is efficient for the classification. We trained our model onto the fashion mnist dataset and also on the human face dataset which can classify colored images into related classes.

Acknowledgment

The authors would like to thank the Head Dr. Ritesh Patel and the principal A.D Patel of the Department of Computer science and Technology, Changa for the constant moral motivation and support. Further, we would like to appreciate the efforts of the reviewing team who spent their valuable time in reviewing our paper.

References

- [1] LecunY, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.
- [2] <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>
- [3] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in ICML, 2010, pp. 807-814.
- [4] <https://medium.com/@mgazar/lenet-5-in-9-lines-of-code-using-keras-ac99294c8086>
- [5] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple convolutional neural network on image classification," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, 2017, pp. 721-724.
- [6] https://en.wikipedia.org/wiki/Supervised_learning
- [7] http://wiki.awf.forst.uni-goettingen.de/wiki/index.php/Talk:Image_classification
- [8] <https://www.tensorflow.org/lite/guide/android>
- [9] <https://www.newgenapps.com/blog/how-image-recognition-will-transform-businesses>
- [10] Y. Boureau, J. Ponce, and Y. Le Cun, "A theoretical analysis of feature pooling in visual recognition," in ICML, 2010, pp. 111-118.
- [11] Hu, M.L.A.X., Recurrent Convolutional neural network for object recognition. 2015.
- [12] Lee, C. Y., P.W. Gallagher, and Z. Tu. Generalizing pooling functions in Convolutional neural networks: Mixed, gated, and tree. in International Conference on Artificial Intelligence and Statistics, 2016.
- [13] <https://developer.android.com/ml>
- [14] <https://stacks.stanford.edu/file/druid:qf934gh3708/EFFICIENT%20METHODS%20AND%20HARDWARE%20FOR%20DEEP%20LEARNING-augmented.pdf>
- [15] <https://www.coursera.org/learn/device-based-models-tensorflow>
- [16] G. M. Foody and A. Mathur, "A relative evaluation of multiclass image classification by support vector machines," in IEEE Transactions on Geoscience and Remote Sensing, vol. 42, no. 6, pp. 1335-1343, June 2004.
- [17] A. Mahajan and S. Chaudhary, "Categorical Image Classification Based On Representational Deep Network (RESNET)," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 327-330.