Deep Indian Delicacy: Classification of Indian Food Images using Convolutional Neural Networks

Article · March 2018

CITATIONS
0

READS
766

1 author:

Shashi Rekha
Visvesvaraya Technological University
10 PUBLICATIONS **4** CITATIONS

SEE PROFILE

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Deep Indian Delicacy: Classification of Indian Food Images using Convolutional Neural Networks

Shamay Jahan[1], Shashi Rekha[2]. H, Shah Ayub Quadri[3]
[1]*M. Tech., semester 4, DoS in CSE, Visvesvaraya Technological University, Centre for PG Studies, Mysuru, India*
[2]*M. Tech., Assistant Professor, DoS in CSE, Visvesvaraya Technological University, Centre for PG Studies, Mysuru, India*
[3]*M. Tech., Data Scientist, Hyderabad, India*

Abstract*: *In this paper, the use of Deep Convolutional Neural Networks (DCNN) for Indian food image recognition and classification is explained. In Computer Vision, food image recognition is one of the most significant and promising applications for visual object recognition. Food image recognition in itself is a visual recognition challenge than any conventional image recognition. Due to huge diversity and varieties of food available across the globe, food recognition becomes very challenging. A deep learning algorithm, Convolutional Neural Networks (CNN) is implemented to recognize and classify the Indian food images. CNN is considered to be the best deep learning algorithm for image classification tasks because it automatically extracts and learns the features from the input images. An experimental study consists of a limited dataset consisting of 60,000 grayscale images of size 280*280 belonging to the ten classes of Indian food. The performance evaluation is based on the classification accuracy. A comparative study is carried out by training the model on CPU and GPU, and the running time is recorded. The model is trained taking only 10,000 images to study the importance of large dataset needed for developing a good model which outputs better accuracy. The graphical representation is also provided. The loss curve, accuracy curve is shown in the graph which in itself is explanation on the importance of large dataset to train the model and how the epochs improve the classification accuracy. The algorithm achieves a fairly good classification accuracy of 96.95% for correct classification of all the images tested in the dataset in just one epoch.

Keywords: Supervised Machine Learning, Pattern Recognition, Convolutional Neural Network (ConvNet/ CNN), Indian food dataset, food image classification, classification accuracy.

I. INTRODUCTION

Food is a necessity for the human existence. Since time immemorial, humans have rapidly progressed towards eating food that suits their taste buds concentrating more on the taste that will tingle their tastes. This led to a huge diversity of food. The food belonging to the same class too has diversity. The difficulty of food recognition increases, by the way it is presented. This raises a problem of recognition and classification of food items.

Object recognition is an integral part of Computer Vision that identifies an object in the given image irrespective of backgrounds, occlusion, the angle of view, lighting, etc. Numerous Machine learning techniques have been deployed for image recognition tasks.

Deep Learning is a subset of Machine Learning that has evolved fairly quickly over the recent times. CNN gained popularity after Krizhevsky et al. [1] won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) 2012 competition. CNN is a deep learning technique that has shown incredible performance for various machine learning and computer vision based problems. Hence, CNN is used for various image recognition (identification) and image classification (categorization) problems.

CNN is a Deep Neural Network (DNN) consisting of additional repetitive Convolutional and pooling layers which automatically learns the features from the given input image instead of feeding the hand-crafted features as observed in the traditional approaches.

Implementing a CNN model requires significantly large number of training images to train the parameters of the network. There are two ways to deal with this problem. One approach is to pre-train the model using a dataset consisting of large number of images of different categories. This approach of model pre-training is called *transferred learning*. Another approach requires modifying the existing dataset by applying affine transformations such as rotation, flip (horizontal, vertical), resize, etc., to expand the existing small dataset.

In this paper, a deep learning model is developed for image recognition and classification. The performance of the model is analyzed based on classification accuracy. Furthermore, model running time on GPU and CPU is also considered.

The following paper is divided into sections as follows: Section II deals with related work to CNN and the motivation factor for developing this project. Section III briefly explains the working of CNN for image classification. In section IV, the proposed

architecture of CNN Model for Indian food image recognition and classification is explained. Section V gives the experimental results of correctly classified images. In Section VI, the conclusion of the work done, hurdles encountered for developing the model and the related future work is explained.

II. RELATED WORK AND MOTIVATION

Significant work is carried out related to image recognition and classification of food images for various purposes such as dietary assessment, recognize diverse food available, analyze calorie intake and eating habits of people, Amazon Go's grocery image detections etc.

Japanese food image classification by Hoashi et al. [2] achieved 62.5% accuracy on 85 food item images collected from the Web. The technique used was multi-kernel learning for feature fusion.

In [3], The Pittsburgh Fast-Food Image dataset consisting American fast food images were recognized using statistics of pair-wise local features [4] and camera phone [5].

In [6], they used a dataset of 10 classes containing 1, 70,000 RGB colour images from Food Log (FL) and achieved a considerable accuracy of 93.8% for food recognition.

III. CONVOLUTIONAL NEURAL NETWORKS

In Machine Learning, [7] a Convolutional Neural Network (CNN, or ConvNet) is a class of deep, feedforward Artificial Neural Network that is widely used for analyzing the visual image recognition tasks. A CNN consists of an input layer, an output layer, and multiple hidden layers. The hidden layers are convolutional layer, pooling layer, loss layer (drop out), fully connected layer (dense layer – Multi-layer Perceptron).

[8]CNN is a multilayer Neural Network in which input to each layer is fed with small patches from the previous layer.

[9][10]Convolutional layer: is the first layer in CNN. It performs automatic feature extraction. A filter is an array of numbers (numbers are called weights/ parameters) denoted by *n*X*n* matrix, where (*n*<image size).

Filters (kernel/ neuron) slides over the input image from the top left corner moving towards right performing elementary multiplication of input image and filter. The performed operation is called **Convolution**. The multiplication result of each convolution is stored in an array. The matrix obtained after sliding the filter throughout the image is called '*activation map/ feature map'*. The filters can be regarded as feature identifiers and the convolutional layer extracts the features to be learned.

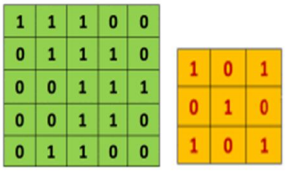

Figure 3.1: 5X5 input image and 3X3 Filter. Source [11]

After each convolution operation, a non-linear operation is performed called *ReLU* (Rectified Linear Unit) for normalization. A pixel-wise operation is performed on the feature map to replace negative values with zero. The obtained feature map is called the *'Rectified feature map'*.

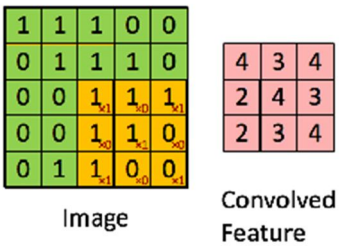

Figure 3.2: The convolution operation. The output matrix is called convolved Feature or Feature Map. Source [11]

Pooling layer: Spatial pooling is also called as subsampling or downsampling. Pooling reduces the image size to 1/4th of its actual size by retaining the most important information. The output is produced by activation performed over the rectangular regions (window). The most commonly used activation methods are average pooling and maximum pooling. This makes the output of CNN invariant to the position.

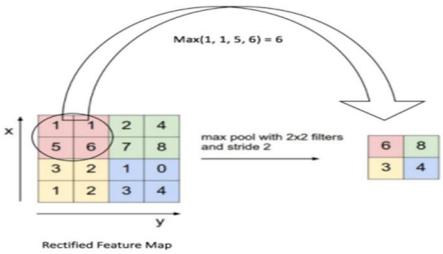

Figure 3.3: The Max-pooling operation. Source [11]

There are several advantages of using pooling operation. Some of them are:

A. It reduces the dimensionality of the input image making it manageable.

B. It reduces the number of parameters and computations in the network, thus reduces overfitting of the network which is a common issue in CNN.

C. Makes the network invariant to small distortions, transformations and translations made to the input image.

1) Fully connected layer: uses the features extracted by the convolutional layer with pooling layer to perform the classification of input image belonging to a particular class based on the training dataset. In this layer, each neuron in the layer is connected with each other neuron in the previous layer.

The Fully Connected layer is a traditional Multi-Layer Perceptron that uses a **s**oftmax activation function in the output layer.

The input to the Softmax activation function is a vector of arbitrary real-valued scores. It produces a vector of values between zero and one that sums to one for the class that the input image belongs to.

[13] The Softmax layer output is: $\sigma(\sum_i w_i x_i + b)$

This gives, for each class i, P (yi = 1; xi; w). For each sample x, the class i with the maximum Softmax output is the predicted class for sample x.

For image classification problem, each unit of the final layer indicates the probability of a particular class.

[9]The overall training process of the CNN is summarized in the following steps:

a) Step 1: Randomly initialize the filters, filter size and weights.

b) Step 2: The training images are split into mini batches and fed into the network through the input layer. CNN performs feed forward propagation extracting essential features from the image. i.e., Conv→ReLU→ Max-pooling→ Fully connected layer.

i. The CNN then finds the output probabilities for each class in training dataset.

ii. Since the weights are randomly assigned to the first training, output probabilities are also random.

c) Step 3: Calculate the total error at the output layer (summation over all 10 classes)

Total Error = ∑ ½ (target probability – output probability) ²

d) Step 4: Backpropagation algorithm is used to calculate the *gradients* of the error with respect to all weights in the network and use *gradient descent* to update all filter values/weights and parameter values to minimize the output error.

i. The weights are adjusted in proportion to the total error.

ii. When the same image is input again, output probabilities become closer to the target vector. This means that the network has *learnt* to classify this particular image correctly by adjusting its weights/filters such that the output error is reduced.

iii. Only the values of the filter matrix and connection weights get updated during the training process while the number of filters, filter size and the architecture remains unchanged.

e) Step 5: Repeat steps 2 to 4 for all images in the training set.

f) Step 6: Test the model with an unseen (new) image by giving input to the CNN and evaluate the model in terms of classification accuracy and Mean Square Error (MSE).

IV. EXPERIMENTATION

A.Dataset

The dataset used for experimentation is a public dataset of Indian food snacks taken from [12]. The dataset consists of 60,000 grayscale images of 280*280 pixels. There are 10 different classes of Indian food images. From the available 60,000 images, 50,000 images are used to train the model and the remaining 10,000 images are used test the model for correct predictions.

The ten classes of food include the following;

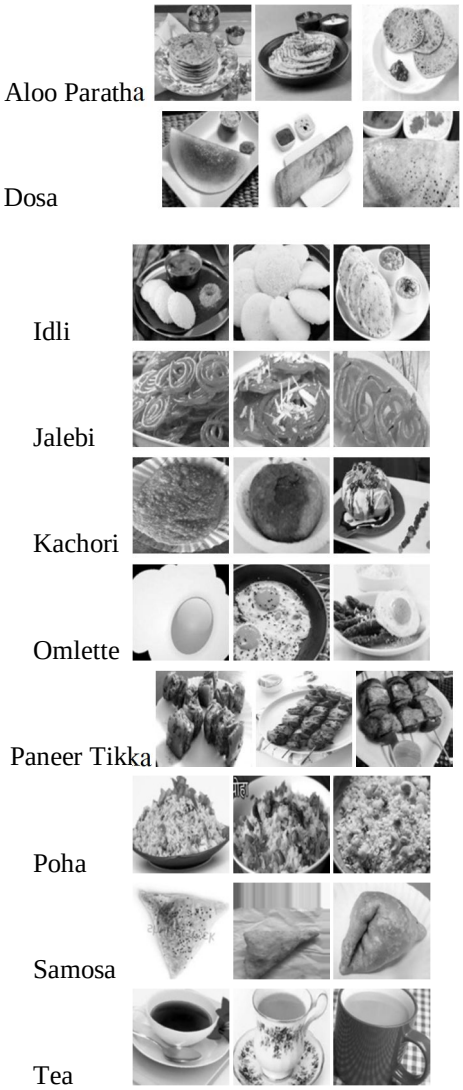

Figure 4.1: Indian Food image dataset

B. Proposed CNN Model

The proposed CNN architecture is designed with an input layer, 5 convolutional layers with ReLU operation and Max-pooling. This is followed by a fully connected layer (combination of dense layer and flatten layer) with ReLU and a dropout layer. Final output layer is fully connected with softmax activation function. ReLU operation and Max-pooling operation is applied to each convolutional layer retaining window size of Max-pooling window to 5X5. The learning rate is set to 1e-3 (i.e., 0.001)

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 276, 276, 32)	832
max_pooling2d_1 (MaxPooling2	(None, 138, 138, 32)	0
conv2d_2 (Conv2D)	(None, 134, 134, 64)	51264
max_pooling2d_2 (MaxPooling2	(None, 67, 67, 64)	0
conv2d_3 (Conv2D)	(None, 63, 63, 64)	102464
max_pooling2d_3 (MaxPooling2	(None, 31, 31, 64)	0
conv2d_4 (Conv2D)	(None, 27, 27, 128)	204928
max_pooling2d_4 (MaxPooling2	(None, 13, 13, 128)	0
conv2d_5 (Conv2D)	(None, 9, 9, 64)	204864
max_pooling2d_5 (MaxPooling2	(None, 4, 4, 64)	0
flatten_1 (Flatten)	(None, 1024)	0
dense_1 (Dense)	(None, 1024)	1049600
dense_2 (Dense)	(None, 10)	10250

Total params: 1,624,202
Trainable params: 1,624,202
Non-trainable params: 0

Figure 4.3: Proposed CNN Architecture with parameters

The input layer accepts a batch of grayscale images. The training dataset with 50,000 images is divided into 125 mini-batches. Each mini batch consists of 400 images (i.e., batch size is 400). 48,800 images are used for training while the remaining 200 images are used for validation. Filter size is fixed to 5X5 and remains unchanged for all the convolutional layers. The window size for pooling layers is also fixed to 5X5. Each image will be fed to the network from the input layer. In the first convolution layer, lines, points etc, are learnt by the model through convolution operation. This outputs 32 feature maps/activation maps on which ReLU operation is performed. Max-pooling operation is performed on these rectified feature maps. This output is fed as input to the second convolutional layer which produces 64 feature maps. This max-pooled output is fed to the third convolutional layer which will produce 128 feature maps. Similarly, fourth convolutional layer produces 64 feature maps, and the fifth convolutional layer outputs 32 feature maps. The last layer of the CNN is the fully connected layer where all the features leant by the convolution-pooling layers are combined so that the model can recognize and classify the input image. A dropout of 0.8 is applied in the fully connected layer. Dropout generalizes the learning of CNN and sets some of the neurons to 0 to prevent model overfitting. The fully
. connected layer has 1024 neurons. The output layer has 10 neurons which are connected to each of the 1024 neurons in the previous layer. Given an input image, only one neuron will be activated using one hot encoding based on the probability value using the softmax activation function. The model is the validated on 200 images and the validation accuracy is recorded.
[14] *'Adam' optimizer* is used instead of the classic SGD (Stochastic Gradient Descent) to update the network weights iteratively in the training data. Adam optimizer requires less tuning, less memory and also computationally efficient. '*Categorical cross entropy loss'* is used to measure the error at the softmax layer.

V. EXPERIMENTAL RESULTS

The model is trained on Nvidia GPU, GTX 1050 Ti 4 GB graphics card with 16 GB RAM and 250 GB SSD and 1 TB HDD.
The model is tested with 10,000 grayscale images. The model is evaluated for its classification accuracy. The result is provided in terms of percentage. A remarkable accuracy of 96.95% is achieved for test data with a loss of 0.143. The validation accuracy is 93.5% with a loss of 0.234.
The CNN image model is developed using Keras on top of Tensor flow in the Spyder environment.

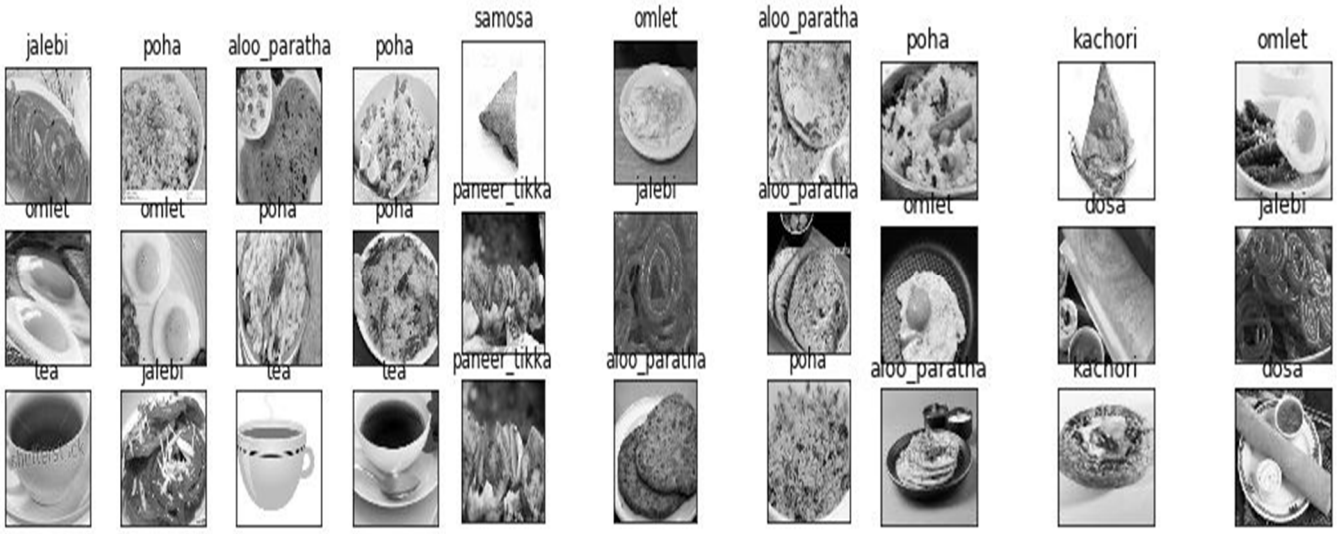

Figure 5.1: Classification results on test images

To study the importance of large dataset and the impact of epochs in improving the accuracy, the model is run on the GPU instance (Ge-Force 1050Ti, 4GB card) taking a small dataset sample of 10,000 images from the 60,000 images, out of which 9,800 images are used for training and the remaining 200 images are used for testing. The epoch is set to 10 with the batch size of 32. It took 14 minutes to train on GPU. The accuracy per epoch and the loss are represented in the tabular form. Also, the graphical representation is included for study. The same dataset was run on CPU (Intel iCore 7 with 16 GB RAM) and the running time recorded is 9 hours.
The study clearly shows that more dataset contributes to better leaning of the model, which in turns provides better classification accuracy. Due to the limited food images available, affine transformations had to be done on the dataset to increase the number of images.

Table I Accuracy at each epoch

Accuracy at each epoch		Accuracy (%)
Epoch	1	0.6371428571428571
	2	0.9492857142857143
	3	0.9751020408163266
	4	0.9775510204081632
	5	0.9846938775510204
		Accuracy (%)
Epoch	6	0.9964285714285714
	7	0.9823469387755102
	8	0.9878571428571429
	9	0.9934693877551021
	10	0.9972448979591837

Table II Loss at each epoch

loss at each epoch		Loss (%)
Epoch	1	1.0057004473649194
	2	0.16050662153153394
	3	0.08757065726063994
	4	0.07583146373308929
	5	0.061145501460212436
	6	0.013505996499759152
	7	0.065771930565896
	8	0.05465052229055318
	9	0.02561323014391041
	10	0.010072417010189235

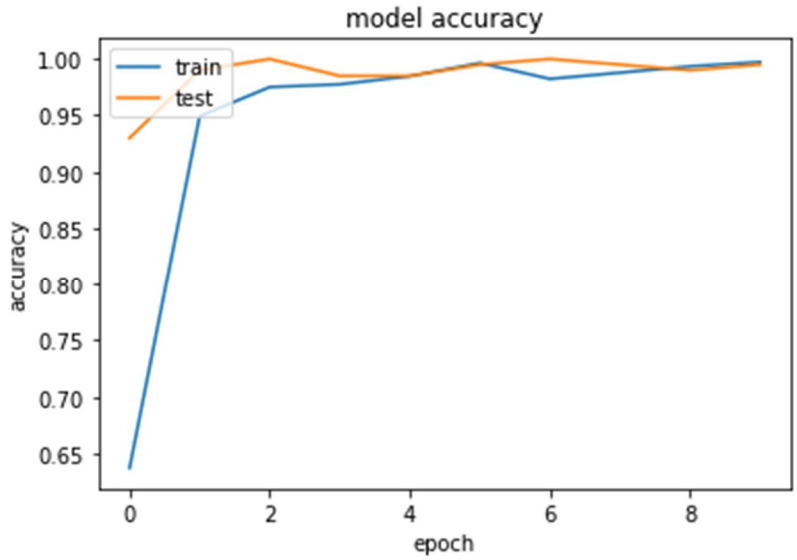

Figure 5.2: Model Accuracy on test and train images with 10 epochs.

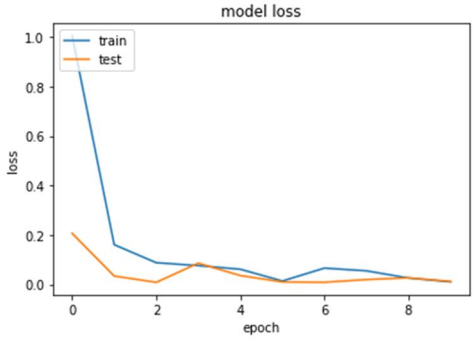

Figure 5.3: Model loss on train and test data with 10 epochs

VI. CONCLUSION AND FUTURE WORK

In this paper, the application of CNN for Indian food image classification is briefly discussed. The paper covers details on working of CNN. So far, no work has been done to classify the Indian food images. The model developed has provided remarkable classification accuracy of 96.95% with only one epoch.
Some hurdles that were encountered during the development of the model are;

A. It is very difficult to develop CNN model on a CPU because CPUs are computationally expensive and consume plenty of time for training and testing the model. Therefore GPU with more memory (RAM) is needed to perform computations.
B. Indian food image dataset has limited images (60,000 images). More training images contribute to better learning and better classification which enhances the classification accuracy. Due to limited images, affine transformations including rotation, translation and scaling are performed on the dataset to increase the training data
C. In [6] the importance of colour for feature extraction process is studied. The use of grayscale image affects the learning accuracy. Due to memory and computational constraints, grayscale images are chosen for the experiment. Grayscale images have a single channel whereas colour images have 3 channels, one for each Red, Green and Blue component.
D. Further enhancements can be made by increasing the number of training images, increasing the number of food classes, reducing image size, using colour images, increasing the number of training epochs, etc. In future, the image classification can be extended to smartphones where a person from a foreign state/ nation will be able to identify the particular food. Also, food image recognition and classification can be used for training the robots to identify and classify a variety of foods so that they can be deployed as butlers for serving in hotels.

VII. ACKNOWLEDGEMENTS

I express my gratitude to our professor Mr K. Thippeswamy for motivation to write a technical paper. I am very thankful for my guide Mrs Shashi Rekha. H, VTU PG Centre, Mysuru for all the guidance and support in making this project successful and guiding me throughout the write up of paper as her research area is Deep CNN.
I equally thank Mr Shah Ayub Quadri, Data Scientist and Deep learning expert, Hyderabad, for helping in project development, imparting necessary knowledge related to machine learning and for making corrections in the paper.

REFERENCES

[1] Krizhevsky, I. Sutskever & G.E. Hinton, Imagenet classification with deep convolutional neural networks, in Advances in Neural Information Processing Systems, 2012. In NIPS, pages 1106{1114, 2012.
[2] H. Hoashi, T. Joutou, and K. Yanai. Image recognition of 85 food categories by feature fusion. In IEEE ISM, pages 296{301, 2010.
[3] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. Pfid: Pittsburgh fast-food image dataset. In IEEE ICIP, 2009.
[4] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar.Food recognition using statistics of pairwise local features. In IEEE CVPR, pages 2249{2256, 2010.
[5] F. Kong and J. Tan. Dietcam: Regular shape food recognition with a camera phone. In IEEE BSN, pages 127{132, 2011.
[6] Kiyoharu Aizawa and Makoto Ogawa: Food Detection and Recognition Using Convolutional Neural Network. In conference ACM Multimedia, At Orlando, Florida, 2014{ DOI: 10.13140/2.1.3082.1120
[7] https://en.wikipedia.org/wiki/Convolutional_neural_network
[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition.
[9] https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/
[10] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
[11] http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

[12] http://github.com/NavinManaswi/IndianSnacks/tree/master/IndianSnacksdatasetand%20 code
[13] http://cs231n.stanford.edu/reports/2017/pdfs/607.pdf
[14] https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning

BIOGRAPHIES

Shamay Jahan is pursuing her M. Tech., in Computer Science and Engineering, at VTU PG Centre, Mysuru. Her field of interest includes image processing, machine learning, etc. She is now developing the CNN model for Indian food Image Classification

Shashi Rekha. H., is working as an Assistant Professor at VTU PG Centre, Mysuru. Her research interests are Image Classification, Data Mining in E-Health, Pattern Recognition, etc. She is pursuing her research in Big Data Analytics.

Shah Ayub Quadri has completed his M. Tech., in Software Engineering. He is working as a Data Scientist in Hyderabad. His area of interests are Programming in Python, R, C# etc., Image Processing, Data Science, Machine Learning, Artificial Intelligence etc.