ASIAN JOURNAL OF PHARMACEUTICAL AND CLINICAL RESEARCH

INNOVARE ACADEMIC SCIENCES
Knowledge to Innovation

Special Issue (April)

Online - 2455-3891
Print - 0974-2441

Full Proceeding Paper

A STUDY ON APPLICATION OF MACHINE LEARNING AND COMPUTER VISION FOR RETAIL PROJECTS

RIMA BORAH, RAJARAJESWARI S

Department of Computing Science and Enineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India. Email: rajarajeswari.s@vit.ac.in

Received: 23 January 2017, Revised and Accepted: 03 March 2017

ABSTRACT

The motivation for developing computer vision is the human vision system which is the richest sense that we have. To us, vision seems an easy task of just seeing objects in daily life and identifying them, but in reality, our eyes along with the brain are processing information of around 50 images every second with millions of pixels in each image. Most of these images obtained are currently just looked at by people. The challenging task is to process images from all these cameras and allow automation of tasks never before considered. Neural networks help us in making cameras intelligent enough to understand the images it captures. Convolutional neural networks (CNN) are trained to give image classification results of good accuracy, with the challenge to improve utilization of computing resources. Google Net is in its essence a deep CNN that uses inception architecture to attain leading edge results for classification and detection problems. In this paper, a study was made on applications of computer vision techniques in retail and customer strategic projects. Further, it was analyzed that if cameras trained with CNN can work well enough to be deployed in retail market scenarios to automate sales and stock supervision.

Keywords: Computer vision, Neural network, Convolutional neural network, Deep learning, Inception architecture.

 DOI: http://dx.doi.org/10.22159/ajpcr.2017.v10s1.20522

INTRODUCTION

Cameras can be found everywhere nowadays in - webcam, surveillance, smartphone, satellite, medical imaging, etc., most of the images obtained from these sources are currently not intelligently interpreted by the computer. The challenge is to process these images to seamlessly automate tasks and reduce human effort and decrease rate of failure. This is where computer vision finds its application. Computer vision is an interdisciplinary field that deals with how to make computers comprehend high-keyed understanding from sources such as digital images and videos [1]. Computer vision tasks include methods for acquiring, processing, analyzing, and understanding digital images. It mainly utilizes image processing and pattern recognition to realize smarter systems. Some examples to be considered – the webcam as a tracking device in place of the mouse, the surveillance cameras to identify bicycle thieves and alert security, the smartphone cameras to capture images of signs and automatically translate them, medical cameras to diagnose conditions more reliably than best expert, and many more.

The aim is to apply computer vision and machine learning to extract information from images and the use this information to design smart customer-centric products at the stock keeping unit (SKU) level.

One of the most important problems of visual recognition is object classification. Instances of visual recognition problem related to image classification are object detection and image captioning [2]. Convolutional neural networks (CNNs) are the state of the art tool to achieve good image classification results. Image classification problem means the task of allocating an input image one label from a determined fixed set of class labels. This is the most standout problem in computer vision. Regardless of the simplicity of this problem, it has a great variety of real-world applications [2].

LITERATURE STUDY

The task of image classification is not a straightforward one for a computer. To competently classify images the computer needs to learn from enough training examples, so as to beat the challenges of image classification from the standpoint of a computer vision algorithm. Some of these challenges are [2]:

- Viewpoint variation: A single instance of an object can be positioned or oriented in numerous ways with respect to facing the camera (Fig. 1).
- Deformation: Many entities of concern may not always be rigid bodies and can therefore be distorted in extreme ways (Fig. 2).
- Scale variation: Objects belonging to the same class may often exhibit variation in their size. For example, humans, animals, and vegetable.
- Occlusion: The objects of interest can be obstructed from view (occluded). Sometimes, only a small portion of an object could be visible. Hence, make identification and classification a challenging task (Fig. 3).
- Illumination: Illumination (brightness/darkness) condition can have

Fig. 1: Viewpoint variation

Fig. 2: Deformation challenge

radical effect on the pixel level and hence make classification difficult (Fig. 4).

- Background clutter: The object(s) of concern may merge into their environment, making them hard to categorize (Fig. 5).
- Intra-class variation: The classes of interest can often be reasonably wide-ranging. There are many different types of these objects, each with their own appearance varying in color, shape, height, and other feature attributes (Fig. 6).

A good image classification model must be invariant to the cross product of all these variations, while concurrently keeping in mind the sensitivity to the inter-class variations [2].

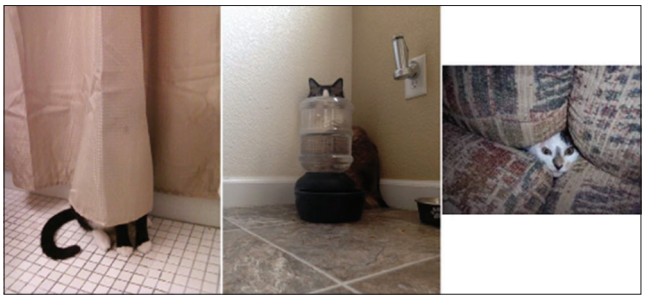

Fig. 3: Occlusion challenge

Fig. 4: Illumination challenge

Fig. 5: Background clutter challenge

Fig. 6: Intra-class variation

METHODS

Any image is denoted by a 3-dimensional collection of numbers. For example, if an image is 640 pixels wide, 480 pixels tall, and has color channels red, green, and blue. The image consists of 640 x 480 x 3 numbers = 921600 numbers. Each number is a number that ranges from 0 (black) to 255 (white). The image classification task is to turn these million numbers into a single label, such as "car" or a "human" [2].

CNN

CNN are motivated biological variants of multi-layer perceptron (MLP) [3]. The connectivity arrangement or pattern between its neurons in a CNN is inspired by the organization of the animal visual cortex. CNNs make use of spatially-local connection amongst neurons of contiguous layers (Fig. 7). The inputs to any of the hidden units in layer say m are formed from a subset of units in the previous layer say m-1. These are the units that have spatially neighboring receptive fields [3].

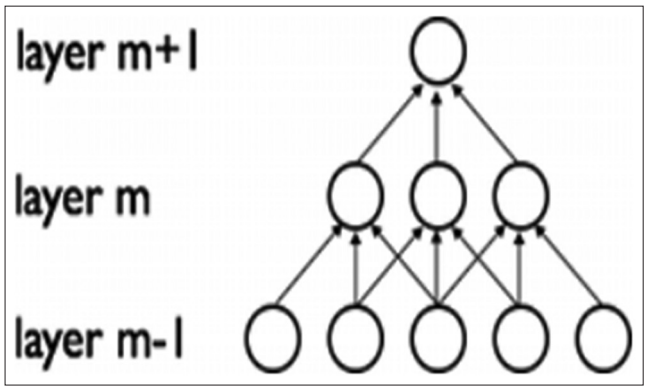

Fig. 7: Connectivity in a convolutional neural network

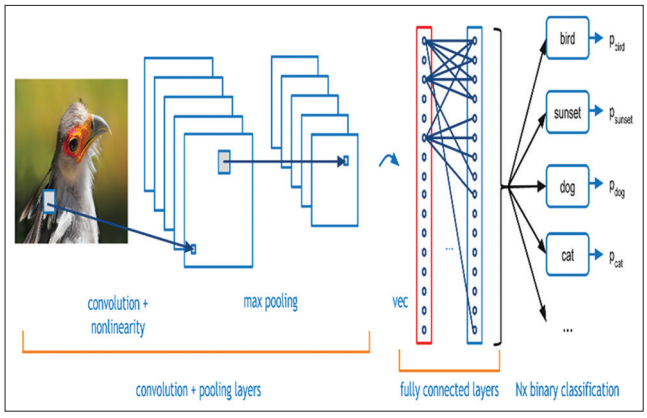

Fig. 8: Convolutional neural network architecture layout

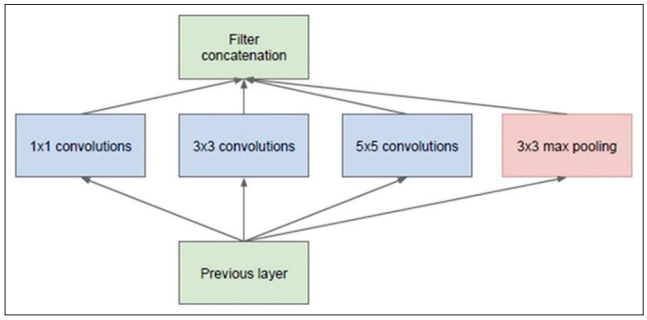

Fig. 9: Inception module, naïve version

CNNs are in essence a sequence of layers and each of these layers transform one volume of activations to another over a differentiable function. In general, three key layers build up CNN architecture: Convolutional layer, pooling layer, and a fully-connected layer [2].

- Convolutional layer will calculate the output of neurons that are linked to local regions in the input. Each will calculate a dot product amid their weights and a small region they are connected to in the input volume.
- Pooling layer does downsampling operation along the width and height of the 3-D arrangement of the pixel array.
- Fully connected layer outputs the class probabilities. Dimension of a fully connected layer can have a volume for example (1×1×10), where each of the 10 numbers match to a class score for each of the 10 classes (Fig. 8).

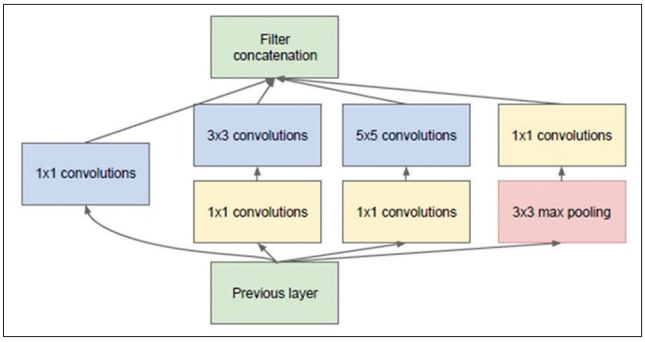

Fig. 10: Inception module, with dimensionality reduction

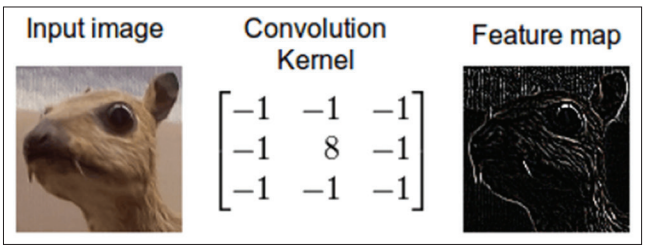

Fig. 11: Convolution filter map

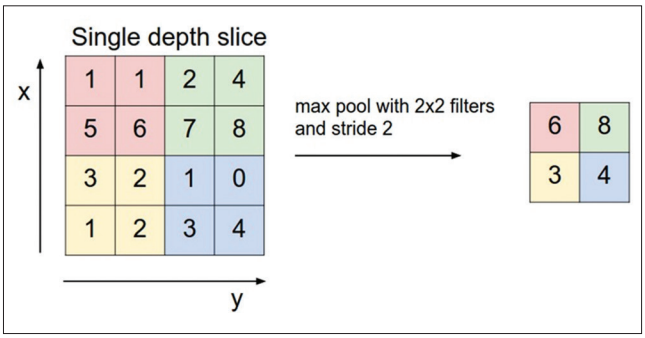

Fig. 12: Convolution filter map

Inception architecture

The inception architecture is a deeper and wider deep neural network architecture that has given the top results in the image net large-scale visual recognition challenge 2014 (ILSVRC 2014) [1]. Inception architecture will contain of the follow components:

- Convolution layer: 1×1, 3×3, 5×5,
- Pooling layer: Average, and max,
- Drop out layer,
- Softmax activation.

This architecture is based on two main ideas:

1. The approximation of a sparse structure with spatially repeated dense components.
2. Reduce computational complexity using dimension reduction.

The module acts as multiple convolutions filter inputs that are processed on the same input. It does also pooling at the same time. All the results are then combined. This allows the model to take advantage of multi-level feature extraction from each input. For instance, it extracts large (5×5) and local (1×1) features at the same time.

Convolutional filters with different sizes can cover different clusters of information. By finding the optimal local construction and repeating it spatially (assuming translational invariance), the module approximates the optimal sparse structure with dense components. For convenience of computation, 1×1, 3×3 and 5×5 filters are used (Fig. 9).

Stacking the inception modules on top of each would lead to an exploding number of outputs. Thus, the use of 1×1 convolutions for dimensionality reduction = while this keeps the computational complexity in bounds, it has a shortcoming. The low dimensional embedding represent the data in a compressed and non-sparse form. Therefore, the dimensionality reduction step (Fig. 10) should only be applied when it is required to preserve sparse representations as much as possible [6]. In practice, 1×1 convolutions are used before doing the expensive 3×3 and 5×5 convolutions.

Convolution operation

Convolution is a general purpose filter effect for images. Convolution filtering finds its use in altering the spatial frequency features for an image. It is a matrix applied to an image and a mathematical operation comprised integers. It works by calculating the value of a central pixel by totaling the weighted values of all its neighbors collectively. The resultant yield is a new altered filtered image. We apply convolution to achieve blurring, sharpening, edge detection, noise reduction, etc., in images.

A convolution is done by multiplying a pixel's and its adjacent pixels color value by a matrix termed kernel. A kernel is a (typically) small matrix of numbers that is used in image convolutions (Fig. 11). By fluctuating, the size of the kernels comprising diverse patterns of

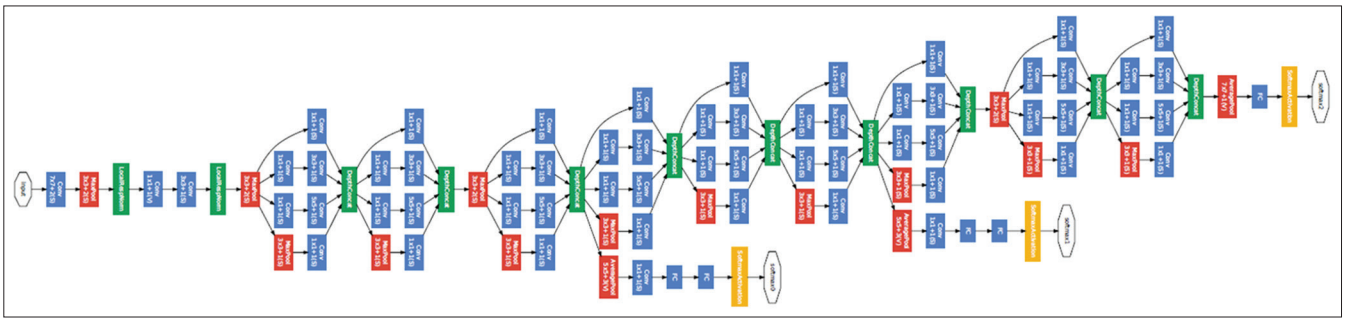

Fig. 13: Google net layout with inception architecture

numbers, different results are obtained under convolution operation. The size of a kernel is arbitrary but 3×3 or 5×5 is often used.

Pooling operation

Pooling is a form of non-linear down-sampling. Of the many non-linear functions to implement pooling, max pooling is most commonly used. The work of the pooling layer in CNNs is to gradually reduce the three-dimensional size of the representation to cut the amount of parameters and computations performed in the network, and as a result also regulate overfitting.

In max-pooling layer, the input image is divided into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum value (Fig. 12). Max-pooling is useful in computer vision for two reasons:

1. By disregarding non-maximal values, it cuts down computation for upper layers.
2. It provides a form of transformation invariance.

The classification module to be deployed will be Google net, an incarnation of the inception architecture (Fig. 13). The version of Google net submitted for ILSVRC 2014 competition achieved top-5 error rate of 6.67% on both validation and testing data and was placed first [6]. The dataset used for this submission contained 1.2 million training images, 50,000 validation images and 100,000 images for testing.

Advantages and disadvantages

CNNs are complicated to train for they contain additional hyper-parameters when compared to a standard MLP. The challenge lies in tuning the hyper-parameters of the CNN to obtain optimum results for the dataset in consideration. The numbers of filters, filter shape, stride, and max pooling-shape needs special attention. While training the network, ensemble learning methodology will be taken on. All the models trained with this procedure will have the same initialization and learning rate policies but vary in sampling methodologies and randomized input image order.

Another issue in CNN is the capacity of the network to over fit. Overfitting occurs when the neural network coadapts too well on the training data [5]. When the learner yields a classifier with 100% accuracy on the training dataset but yields only 50% accuracy on test dataset and reality it could yields a classifier that is 75% accuracy on both,then it is said to have over fit [4]. To combat, this issue of overfitting, regularization techniques are used [5].

Google net uses the regularization technique of drop out. Google net architecture adds auxiliary classifier to the intermediate layers, of which drop out layer forms an important part [6]. To apply drop out means to temporarily remove a unit from the network, along with all its incoming and outgoing connections (Fig. 14). Units are randomly omitted with a fixed probability p (usually set to 0.5) independent of other units. By applying drop out, the network is forced to learn several independent representation of the data (not relying on other hidden units to be present), thus preventing complex coadaptions. This makes each hidden unit robust (Fig. 15) [3].

A significant obstacle in obtaining satisfactory results using CNN or any deep neural architecture is that the network requires a large amount of training data to learn all possible features well enough. In many cases, such datasets with an ideal number of image instances may not be available. To handle this, image augmentation techniques are applied. To augment an image is to apply different filters on it to obtain an image varied in orientation, light conditions, etc. This technique also makes sure that the network learns to extract features from the image irrespective of external factors affecting it.

Expected results

At the SKU level, implementation of computer vision using machine learning capabilities can go about to reduce the effort of periodic human supervision to a great degree. Cameras trained well with neural networks can successfully identify objects of interest (the products at SKU level) and give count and class of present stock. Stock-keepers can keep track of the stock without manual supervision at any time. Stock-keepers will also have a better idea of fast-moving products. This will also work to improve customer retention because there will be very less chance of stocks running out completely.

At the neural network level, image classification of retail products needs sufficient training to produce satisfactory results (Fig. 16). The first job

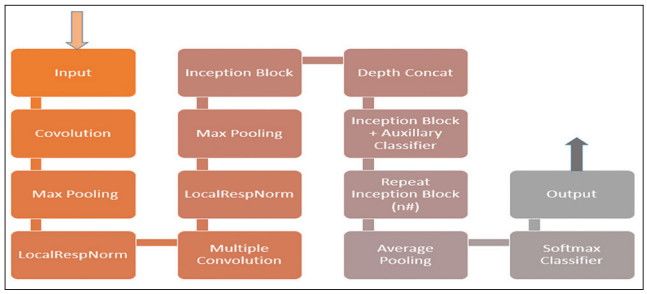

Fig. 14: Taxonomy of flow in a Google net architecture using inception module

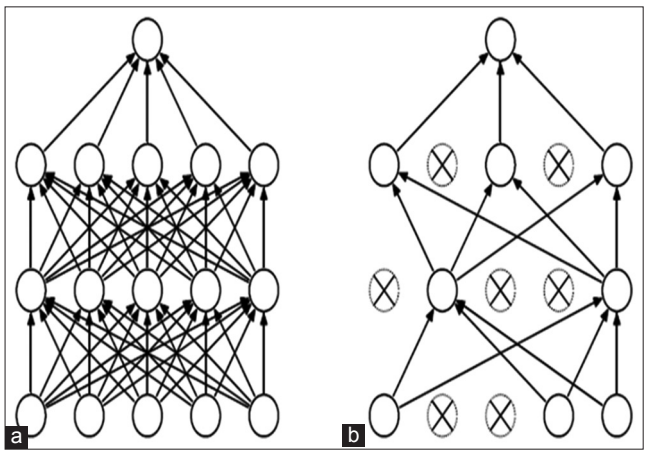

Fig. 15: (a and b) Standard neural network versus neural network with drop out

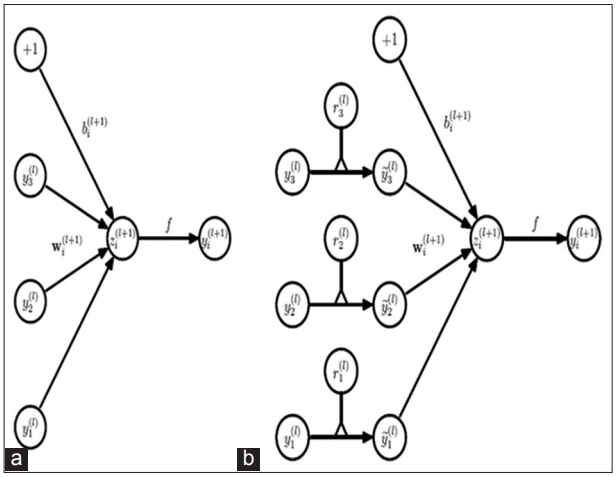

Fig. 16: (a and b) Learning in standard neural network versus learning in neural network with drop out

of a neural network will be to identify object(s) of interest and discard any unwanted noise in the background. Once, the region of interest in the image captured is established, a neural network should now be trained for edge detection of each item of the stock. This enables to establish the count of each and every individual item in stock. The next step will be to classify every counted item to generate a stock summary. The camera should be able to detect every time item(s) move out and then, the stock count is refreshed accordingly.

CONCLUSION

Intelligent retail stock keeping has been implemented before using weight sensors, RFID tags, etc. However, these implementations had various shortcomings and produced inaccurate results. Inaccurate stock keeping can hamper the business in sales and revenue. Inefficient stock keeping will, in turn, result in an increase in customer churn rates if customer needs are not aptly taken care of.

An intelligent camera can go a long way to solve everyday business problems reducing human labor and skill and produce results similar to what a human eye can. Although initially, investment of time and resources in training a neural network is high, but once trained successfully it makes tasks quite smooth and reliable.

The goal is to improve the architecture of deep convolutional networks to achieve performance gain of computer vision tasks reliant on high quality learned visual features [7].

REFERENCES

1. Computer Vision. Available from: https://www.en.wikipedia.org/wiki/Computer_vision.
2. Convolutional Neural Networks for Visual Recognition. Available from: http://www.cs231n.stanford.edu/.
3. Convolutional Neural Networks (LeNet). Available online: http://deeplearning.net/tutorial/lenet.html.
4. Domingos P. A Few Useful Things to Know about Machine Learning. Vol. 55. University of Washington: Communications of the ACM; 2012. p. 78-87.
5. Hinton G, Krizhevsky A, Srivastava N. Improving neural networks by preventing co-adaption of feature detectors, arXiv:1207.0580v1 [cs. NE] 3 July; 2012.
6. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S. Going deeper with convolutions. Computer Vision and Pattern Recognition, arXiv:1409.4842v1 [cs.CV] 17 September; 2014.
7. Szegedy C, Vanhouchke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision, arXiv:1512.00567v3 [cs. CV] 11 December; 2015.
8. Lin M, Chen Q, Yan S. Network in Network. arXiv:1312.4400v3 [cs. NE] 4 March; 2014.
9. Srivastava N, Hinton G, Krizhevsky A. Dropout: A simple way to prevent neural networks from over fitting. J Mach Learn Res 2014;15(1):1929-58.
10. Nielsen M. Neural Networks and Deep Learning. Available from: http://www.neuralnetworksanddeeplearning.com/index.html.
11. Convolutional Neural Networks for Visual Recognition. Available form: http://www.cs231n.stanford.edu/.
12. Building Powerful Image Classification Models using Very Little Data. Available from: https://www.blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html.
13. Deep Learning Framework: Caffe. Available from: http://www.caffe.berkeleyvision.org/.
14. Inceptionism: Going Deeper into Neural Networks. Available from: https://www.research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html.
15. TensorFlow Playground. Available from: http://www.playground.tensorflow.org.
16. Krizhevsky A, Sutskever I, Hinton GE. Image Net Classification with Deep Convolutional Neural Networks. Advances in Neural Information Proceeding Systems (NIPS; 2012).