Grenze International Journal of Engineering and Technology, Special Issue

An Exploration of Deep Learning in Recognizing Household Objects

Smita Gour [1] and Pushpa B. Patil[2]
[1]Basaveshwar Engineering College, Bagalkot, India
Email: smita.gour@gmail.com
[2]BLDEA's V P Dr PG Halakatti College of Engineering & Technology, Vijayapur, India
Email: pushpa_ms@rediffmail.com

***Abstract*—A realistic deep learning based mechanism for household object recognition is presented in this article. The deep learning encourages automatic feature extraction and works toward recognition under a stipulated workspace. Convolution Neural Network is one of the deep learning approaches which excelled in recognizing objects in images. It is implemented using 3 or more layers where each layer extracts one or more feature of the image. The Convolution Neural Network makes the user work easy by itself preprocessing the images fed by the user. The only thing here is to provide a large set of images to train or else augment the data. The model is going to compare the given image's features with the features of the training set images in order to recognize image and classify that into the category it belongs among various category of household objects. Tensorflow framework has been used to develop this deep learning application. Parameter tuning and data augmentation has been employed to improve the performance of the system. This model is able to classify new instances of objects with an accuracy of 87.96% using dataset of 56 object categories.**

***Index Terms*— Deep Learning, Convolution Neural Network, Tensorflow, Object Recognition.**

I. INTRODUCTION

Human recognize a multitude of objects in images with little effort, despite the fact that the image of the objects may vary in different viewpoints, such as scaled, translated and rotated. This task of object recognition is still a challenge for computer vision systems because objects in images appear in different scales, sizes and they may be geometrically rendered differently depending on the distance between the camera and the object. Also different sources of illumination can drastically change the appearance of an object in images. Computer vision is a field that includes methods for acquiring, processing, analyzing and understanding images in order to produce numerical or symbolic information to form decisions. One of the most challenging fields of computer vision is robotics where the object recognition is a major task. In recent years, research in robotic vision has gained attention and is increasing significantly. Science fiction has filled our heads with the dreams of human-like robots. In real-world applications, service robots are used for general purpose to locate and recognize the objects. For robots to be accepted in the home and offices, they need to identify specific or a general category of objects requested by the user. Household robots should take

Grenze ID: 01.GIJET.6.2.4

the advantage of their mobility and their relative static environments to make object recognition easier, by imaging objects from multiple perspectives before making judgments about object identity. Matching up the objects depicted in different images poses its own computational challenges. This motivated to propose research work for development of machine learning based techniques for various tasks of household object recognition system useful in robotic vision. The household object recognition can be accomplished in either off-line (by recognizing objects in images/photographs), or in real-time (the object is seen in the field of view of a camera). The algorithms for solving the problem of offline object recognition are more challenging than online object recognition because images of objects are not clear and affected with noise, occlusion, clutter and different illuminations. Rigorous survey has been undertaken to develop methods for these tasks. The details are discussed in ensuing section.

A. Literature Survey

Few of contemporary mechanisms for recognition of household objects found in related literature is presented in this section. All those mechanisms can be broadly categorised as follows.

1) CAD like primitive based mechanisms: These uses basic structures such as edges, primal sketch, etc.
2) Constitutional part based approach: Typically, methods that attempt to recognize the object based on recognition of constituent parts.
3) Methods based on how objects look like [1]: Attempts to recognize objects based on comparision to template objects.
4) Methods that rely on features that are invariant: These methods involve mechanisms to identify and extract the invariant features of objects precisely and making decision based on extracted features to recognize objects.

In literature many methods that attempt to recognize the objects from the image are reported. The method for object recognition based on Support Vector Machine (SVM) and RST invariants is presented in [3]. The method employs principal component analysis (PCA) for description reduction and SVM classifier for recognition. Invariant feature based methods [4] [5] [6-8] which are invariant to scale, translation and rotation are also present in literature. These all methods use KNN and SVM classifier for object recognition. The method presented in [9] relies on local and global features of the objects for 3D object recognition using Point Cloud Library (PCL). The method follows hybrid approach employing Viewpoint Feature Histogram (VFH) and Fast Point Feature Histogram (FPFH) methods for object recognition. For the object recognition based on global description VFH is used. Whereas, FPFH method is used as a local descriptor to estimate the position of the object in a scene. A strategy learnt system to select 3-D object in point cloud descriptor is presented in [10]. The mechanism is based on a 3-D classifier pipeline which involves reinforcement learning.

Basically image based object recognition has various sub tasks like preprocessing (removal of noise and other degradations in image [11], [12] and [13], object segmentation in image [14], [15], [16] and [17] and recognition using learnt mechanisms [23], [24], and [25] based on features that are extracted from image [18], [19] [20] and [21] . Efficient methods are yet required to perform these subtasks for object recognition. A deep learning based algorithm for household object recognition is presented in this research work.

The current computer vision systems are now relying on machine learning [2] for recognition of real world objects acquired images. Through machine learning a system can acquire the knowledge to recognize object from scene when accurately designed. In designing such intelligent system focus should be on building computer programs that are capable of learning the context, evolve and adapt when exposed to newer contexts. One of such new techniques called deep learning [25] which is the part of the machine learning is getting more attention of researchers. The results of applying deep learning in many applications [26], [27] are becoming more promising. In deep learning, a machine models learns to perform independently, the classification tasks from images, text, or sound with almost accurate results as compared to past. The models get trained by huge labeled data templates and neural network structure with many layers.

One such methodology is proposed and presented in this paper and its details are depicted in section II. In section III the detailed experimentation conducted has been discussed. And section IV depicts the conclusion and some future work.

II. PROPOSED METHODOLOGY

Our proposed module is a deep learning based model which makes use of convolution neural network. There are two phases one is training phase and another is testing phase. Model will be trained and knowledge is

created which will be made use in object recognition. The steps in proposed methodology are explained below.

A. Object Database

The proposed methodology is built for both standard database as well as real time database of household objects. The standard database is build by taking images containing objects from many databases like coil 300, home objects06, Caltech 100, willow 2D-3D. About 46 different categories of house hold object images each containing about 120 samples have been collected. Some of these categories have been shown in the Figure 1. Some of real time object images are also captured using webcam, like cellphone, watch, water bottle, spectacle, bag, bucket, mug, exam pad, toothbrush and pen. And these are combined with standard dataset. In this way at last we had a dataset with 56 object classes and a total collection of 6000 images. The Figure 2 represents the images of real world.

Figure 1. Standard dataset of household objects

Figure 2. Real time dataset of household objects

B. Convolutional Neural Network

One of the most familiar deep neural networks is Convolution Neural Network (CNN). A CNN interwines features learned with input data using 2D convolution layers, making its structure well accepted to process 2D data, such as images. CNN does the job of feature extraction by itself. So there is no need of identifying and extracting features manually. This automated feature extraction makes deep learning models highly recommended for computer vision tasks like object classification.

The architecture of the CNN is shown in the Fig 3. The convolution neural network has four layers. They are convolution, pooling, flattening and fully connection.

We assume a gray scale or binary image 'I' represented by size $n_1 \times n_2$ defined by a function given in Eq. (1).

$$\mathbf{I: \{1,...,n_1\} \times \{1,...,n_2\} \rightarrow W \subseteq R,(i, j)\ 7\rightarrow I_{i,\,j}} \qquad \mathbf{(1)}$$

Given the filter $K \in R^{2h_1+1 x 2h_2+1}$, the discrete convolution of the image I with filter K is given by Eq. (2).

$$\boldsymbol{I * K = \sum_{u=-h1}^{h1} \sum_{v=-h2}^{h2} K_{u,v} I_{r+u,s+v}} \qquad \mathbf{(2)}$$

where the filter K is given by Eq. (3)

$$\boldsymbol{K = \begin{pmatrix} k_{-h1,-h2} & \cdots & k_{-h1,h2} \\ \vdots & k_{0,0} & \vdots \\ k_{h1,-h2} & \cdots & k_{h1,h2} \end{pmatrix}} \qquad \mathbf{(3)}$$

A filter which is most familiar for smoothing is the discrete Gaussian filter $K_{G(\sigma)}$ which is given by Eq. (4)

$$\mathbf{K}_{G(\sigma)} = \frac{1}{\sqrt{2\pi\sigma^2}} exp2^{\left(\frac{r^2+s^2}{2\sigma^2}\right)} \quad \textbf{(4)}$$

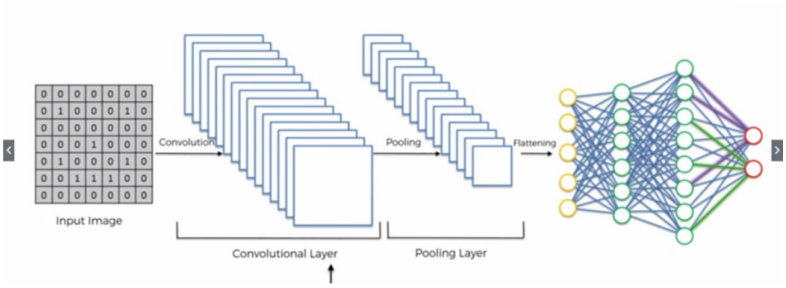

Figure 3: CNN architecture

a) Convolution Layer: The heart of the CNN architecture is this layer because almost all computational work is accomplished by it. The filters which are learnable are the important parameters to be considered. Each of them is small window with width and height spatially convolves through the input volume and dot products are computed between the entries of the filter during the forward pass.

Assume layer *l* be a convolution layer. The input to this layer constitutes $\boldsymbol{m_1^{(l-1)}}$ feature maps from the preceding layer with $\boldsymbol{m_2^{(l-1)}} \times \boldsymbol{m_3^{(l-1)}}$ size. When *l=1*, image I is a input consisting of one or more mediums. This is how a raw images are fed as input to the convolution neural network. The output from layer *l* constitutes $\boldsymbol{m_1^{(l)}}$ feature maps $\boldsymbol{m_2^{(l)}} \times \boldsymbol{m_3^{(l)}}$ sized. Eq. (5) is used to compute the feature map (i^{th}) of layer *l*.

$$\mathbf{Y_i^{(l)}} = \mathbf{B_i^{(l)}} + \sum_{j=1}^{m_1^{(l-1)}} \mathbf{K_{i,j}^{(l)}} * \mathbf{Y_j^{(l-1)}} \quad \textbf{(5)}$$

Where, $\mathbf{B_i^{(l)}}$ is a bias matrix and $\mathbf{K_{i,j}^{(l)}}$ is the filter of size $2h_1+1 \times 2h_2+1$ coupling the j^{th} activation map in layer $(l-1)$ with the i^{th} activation map in layer *l*. $m_2^{(l)}$ and $m_3^{(l)}$ are determined by border influences.

If layer *l* is a not linear, $m_1^{(l)}$ activation maps are the inputs to it and $m_1^{(l)} = m_1^{(l-1)}$ activation maps are the outputs, each of size $m_2^{(l)} \times m_3^{(l)}$ along with $m_2^{(l)} = m_2^{(l-1)}$ and $m_3^{(l)} = m_3^{(l-1)}$, given by Eq. (6).

$$\mathbf{Y_i^{(l)}} = \boldsymbol{f}\left(\mathbf{Y_i^{(l-1)}}\right) \quad \textbf{(6)}$$

where $\boldsymbol{f}$ is the activation function like Relu, hyperbolic tangent used in layer *l* and operates point wise.

b) Pooling Layer: Pooling is one more notion which is down-sampling with non-linearity. There are several such functions to adapt pooling. Max pooling is one among them which is the most common. It sections the input image into number of rectangles which are not overlapping with each other. The output of the pooling layer is maximum value of such sub-region. The pooling layer reduce the spatial size of the representation so that number of parameters, memory and amount of computation in the network is reduced, and hence controls the over fitting.

Assume pooling layer as *l* and $m_1^{(l)} = m_1^{(l-1)}$ activation maps of reduced size is the output of it. Generally pooling is accomplished by windows placed at positions which are not overlapped in each feature map and preserves one value per window. There are two ways of pooling. They are as follows.

1) Max pooling: As shown in the Fig 4 max pooling is to obtain the maximum value from each window. This is given by P_M. As discussed in [27], max pooling helps to reach convergence faster during training.

2) Average pooling: For the average pooling, the average of all the values in each window is taken as shoen in the Fig 4. The layer is denoted by P_A.

c) Flattening layer: After finishing the previous two steps, it is going to flatten our pooled feature map into a column. The reason it does this is that it is going to insert this data into an artificial neural network.

d) *Fully connected layer:* All the high level rationality obtained in the deep learning models is through this layer. It takes the output from several convolution and max pooling layers and performs affine transformation with matrix multiplication and bias offset to compute its activations.

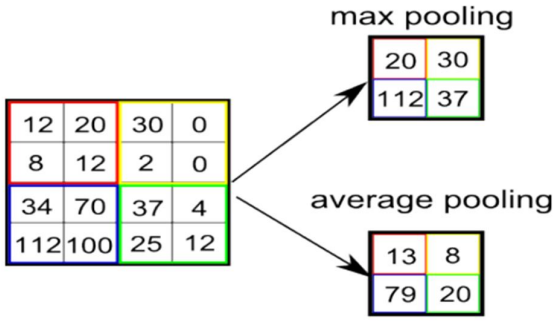

Figure 4: Pooling Process

Neurons in a fully connected layer have connections to all activations in the previous layer. This is why the convolution network are more efficient in classifying images.

Assume fully connected layer as *l*. The input to this layer is $m_1^{(l-1)}$ activation maps with $m_2^{(l-1)} \times m_3^{(l-1)}$ sized and any unit i in layer *l* calculated its output using Eq. (7).

$$\mathbf{y}_\mathbf{i}^{(l)} = \boldsymbol{f}\left(\mathbf{z}_\mathbf{i}^{(l)}\right) \qquad \textbf{(7)}$$

Where, $\mathbf{z}_\mathbf{i}^{(l)} = \sum_{j=1}^{m_1^{(l-1)}} \sum_{r=1}^{m_2^{(l-1)}} \sum_{s=1}^{m_3^{(l-1)}} \boldsymbol{w}_{i,j,r,s}^{(t)} \left(\mathbf{Y}_\mathbf{j}^{(l-1)}\right)_{r,s}$.

$\boldsymbol{w}_{i,j,r,s}^{(t)}$ is the weight with which the unit at position (r, s) in the j[th] feature map of layer $(\boldsymbol{l-1})$ and the i[th] unit in layer l are connected.

All these layers are implemented using Karas API which is written in Python and capable of running on top of TensorFlow, CNTK, or Theano. Keras integrates with lower-level deep learning languages (in particular TensorFlow), it enables you to implement anything you could have built in the base language.

III. EXPERIMENTATION

Initially it was tried to achieve latest substantial accuracy progressively by encompassing only 10 objects then went on to increase it to 56. We had to make certain changes to dataset as we increased the number of objects.

B. Experiment 1

In this experiment model for 10 objects and objects that were only in center of an image was built. Here we kept the quantity of images for each object class limited to 50 in training set and 12 images in test or validation set for validating the accuracy of model. We even created one more dataset named as ''Random Images'' which constituted of images which were not present in neither in training nor in test set. In this model we took a total of 150 epochs and at the very first attempt we got accuracy of 97.92%. in order to improve the accuracy epochs were increased to 200 and it resulted in improving accuracy i.e. 97.99%.

C. Experiment 2

Here, the model created in experiment 1 is extended for 35 objects and tried to encompassing objects that are not in centre of an image. Now during the creation of the dataset for this model we had to increase the number of images of each object class to 80 images/class for training and 25 images /class for test set. Here for same class of objects we needed to provide more images in same orientation and position as the features to extract from every image increased exponentially. We cropped the images as to try to keep objects as central in image as possible and some of the images had objects that were not in centre so that to expand horizons of our model. Now, it can recognize objects which were not in centre too. When we trained our model with 35 objects classes we increased the epoch numbers to 400. Here we got discouraging result. i.e. a accuracy of only 76.21%. This is because we have enlarged the dataset. We then decided to slightly change our model by changing argument values of some parameters. Here we have increased the number of convolution filters from 32 to 128 and again, new model was kept for training. This time accuracy reached 78.01%. Again the changes to the data augmentation function i.e. ImageDataGenerator () has been made by adding horizontal flip parameter to it. It resulted in accuracy of 84.92%.

C. Experiment 3

Here the model for 56 objects which also includes real time objects is built. During this phase we wanted to take our model a notch up to make it capable of recognizing the objects that were captured through webcam of laptop. To make it happen we needed to make the network denser so that it can work with various lighting and background scenarios the real-time image will incorporate. We then went on to create image dataset consisting real-time samples of various possible object classes. Planning to capture the image with minimal obstruction creating parameters was one of the most exciting phases of working in this experiment. By training this 56-object class model accuracy decreased to 84.24%. At this time epochs were increased to 600. Again, changes were made to the convolution layer. Here we have increased the number of convolution filters from 128 to 256 and with this change accuracy attained was 84.31%. To improve accuracy still more parameters were added to ImageDataGenerator function and shear range value is changed from0.2 to 0.4. Some of the added parameters are like vertical_flip, fill_mode, width_shift_range, height_shift_range. Thus, it was able to reach an accuracy of 87.96%.

The overall experimentation summary and accuracy obtained in each experimentation are given in Table I.

TABLE I. PARAMETRIC TABLE

Experiment	No. of training samples/object	No. of Objects	No. of Epochs	No. of filters	Data Augmentation	Accuracy (%)
I	50	10	200	32	No	98
II	80	35	400	128	No	78
					Yes/With shear range 0.2 and only horizontal flip	85
III	100	55	600	256	Yes/With shear range 0.4 and with both horizontal and virticle flip	87.96

IV. CONCLUSION

Object recognition through computer vision has many state-of-art applications in many domains. In this a computer based model is expected to be trained for identification, separation and recognition of objects from images and scene. One such system is presented in this work. Object recognition is done employing a deep learning approach called Convolution Neural Network which is inspired by biological system. The feed forward back propagation technique is used as a classifier. One drawback of CNN is that it requires lot of input for the training and as the size of database increased, it takes more time to recognize the object. Till now the model recognizes only single object in mage. The accuracy rate ranging from 87.34% - 97.99% for 10 to 56 object classes is obtained. This will work if it is embedded in as a vision system of very basic robots which work in a very restricted workspace. The work can be extended by building a model that recognizes multiple objects in images. Also training time may be reduced by applying segmentation techniques on images. Also movement invariant features can be combined with the proposed features to increase the accuracy. The use of webcam constituted for implementation of real-time recognition now in the model developed. Provisioning this model to work with independent cameras is to be worked on it in future. Then this vision system will be more reliable than its now.

REFERENCES

[1] F. W. Mattern and J. Denzler , "Comparison of appearance based methods for generic object recognition," *Pattern Recognition and Image Analysis*, Vol. 14, No. 2, pp. 255–261.2004

[2] V. N. Pawar and S. N. Talbar, "Machine learning approach for object recognition," *International Journal of Modeling and Optimization*, Vol. 2, No. 5, pp. 622-628, October 2012.

[3] R.Muralidharan and Dr.C.Chandrasekar, "Object recognition using support vector machine augmented by RST invariants," *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 5, No 3, pp. 280-286, September 2011.

[4] R.Muralidharan and Dr.C.Chandrasekar, "object recognition using svm-knn based on geometric moment invariant," *International Journal of Computer Trends and Technology*, pp. 215-219, July 2011.
[5] R.Muralidharan, "Object recognition from an image through features extracted from segmented image," *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 4, Issue 12, pp. 205-209, December 2014.
[6] R.Muralidharan, **"**Object recognition using k-nearest neighbor supported by eigen value generated from the features of an image," *International Journal of Innovative Research in Computer and Communication Engineering* , Vol. 2, Issue 8,pp. 5521-5528, August 201.
[7] Muralidharan, R. and C. Chandrasekar, "3D object recognition using multiclass support vector machine-k-nearest neighbor supported by local and global feature," *Journal of Computer Science*, pp. 1380-1388, 2012.
[8] R.Muralidharan, 2Dr.C.Chandrasekar, "Scale invariant feature extraction for identifying an object in the image using moment invariants," *Journal of Engineering Research and Studies*, Vol.2, Issue 1, pp. 99-10, March 2011.
[9] Khaled Alhamzi, Mohammed Elmogy, Sherif Barakat, "3D Object Recognition Based on Local and Global Features Using Point Cloud Library," *International Journal of Advancements in Computing Technology (IJACT)*, Vol. 7, Number 3, pp.43-54, May 2015.
[10] Jens Garstka and Gabriele Peters, "Adaptive 3-D Object Classification with Reinforcement Learning", Accepted for *International Conference on informatics in Control, Automation and Robotics* that will be held during July 21-23 2015.
[11] Mohammad Reza Faraji and Xiaojun Qi , "Face recognition under varying illumination with logarithmic fractal analysis," *IEEE Signal Processing Letters*, Vol. 21, No. 12, pp. 1457-1461, December 2014.
[12] Shan Du , Rabab K. Ward , "Adaptive region-based image enhancement method for robust face recognition under variable illumination conditions," *IEEE Transactions On Circuits And Systems For Video Technology*, Vol. 20, No. 9, pp. 1165-1175, September 2010.
[13] Mohd Awais Farooque, Jayant S.Rohankar , "Survey on various noises and techniques for denoising the color image," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, Vol. 2, Issue 11, pp. 217-221, November 2013.
[14] G. N. srinivasan, Shobha G, "Segmentation techniques for target recognition", *International Journal Of Computers And Communications,"* Vol. 1, Issue 3,pp. 75-81, 2007.
[15] Reza Oji, "An automatic algorithm for object Recognition and detection based on ASIFT Keypoints," *Signal & Image Processing: An International Journal (SIPIJ)*, Vol.3, No.5, pp. 29-39, October 2012.
[16] Khushboo Khurana, Reetu Awasthi, "Techniques for object recognition in images and multi-object detection", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Vol. 2, Issue 4,pp. 1383-1388, April 2013.
[17] Nikita Sharma, Mahendra Mishra, Manish Shrivastava, "Colour image segmentation techniques and issues," *International Journal of Scientific & Technology Research* Vol. 1, Issue 4, pp. 9-12, May 2012.
[18] Aristeidis Diplaros , "Color-Shape Context for Object Recognition," *IEEE Workshop on Color and Photometric Methods in Computer Vision*,pp.1-8,2003.
[19] Edward Hsiao, Alvaro Collet and Martial Hebert, "Making specific features less discriminative to improve point-based 3d object recognition," IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, pp.1-8, June 2010.
[20] Cedric Lemaitre, Fethi Smach, Johel Miteran, Jean Paul Gauthier and Mohamed Atri, "A comparative study of motion descriptors and zernike moments in color object recognition," *Conference: IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference*, pp.1-6, 2006.
[21] V.K Ananthashayana and Asha.V, "Appearance based 3d object recognition using IPCA-ICA," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol 37, Part B1, pp.1083-1090, 2008.
[22] Morgan & Claypool Publisher, "Algorithms for Reinforcement Learning", Draft of the lecture published in the Synthesis Lectures on Artificial Intelligence and Machine Learning, June-2009.
[23] Rich Caruana, Alexandru Niculescu-Mizil, "An empirical comparison of supervised learning algorithms", *Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, 2006.*
[24] Federico Prandi, Raffaella Brumana, Francesco Fassi, "Semi automatic objects recognition in urban areas based on fuzzy logic", *Journal of Geographic Information System*, 2010, Vol. 2, pp. 55-62.
[25] James Wang, "DEEP LEARNING: An Artificial Intelligence Revolution", *A white paper ARK Invest, 2011, pp. 1-41*
[26] Y. LeCun, K. Kavukvuoglu, and C. Farabet. *Convolutional networks and applications in vision. In Circuits and Systems*, International Symposium on, pages 253–256, 2010.
[27] D. Scherer, A. Müller, and S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In Artificial Neural Networks, International Conference on, pages 92–101, 2010.